TECHNISCHE
UNIVERSITÄT
DARMSTADT

COLLABORATIVE GAME-BASED LEARNING – AUTOMATIZED ADAPTATION MECHANICS *for* GAME-BASED COLLABORATIVE LEARNING *using* GAME MASTERING CONCEPTS

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

DIPL.-INFORM. VIKTOR WENDEL

Geboren am 22. Juli 1983 in Bad Kissingen

Vorsitz: Prof. Dr.-Ing. Franko Küppers
Referent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Dr.-Ing. Wolfgang Effelsberg

Tag der Einreichung: 10. Februar 2015
Tag der Disputation: 17. April 2015

Hochschulkennziffer D17
Darmstadt 2015

To my parents Hannelore and Gerhard and my brother Christian.
Thank you for your loving and continuous support.

ABSTRACT

Learning and Playing represent two core aspects of the information and communication society nowadays. Both issues are subsumed in Digital Education Games, one major field of Serious Games. Serious Games combine concepts of gaming with a broad range of application fields: among others, educational sectors and training or health and sports, but also marketing, advertisement, political education, and other societally relevant areas such as climate, energy, and safety. This work focuses on collaborative learning games, which are Digital Educational Games that combine concepts from collaborative learning with game concepts and technology.

Although Digital Educational Games represent a promising addition to existing learning and teaching methods, there are different challenges opposing their application. The tension between a game that is supposed to be fun and the facilitation of serious content constitutes a central challenge to game design. The often high technical complexity and especially the instructors' lack of control over the game represent further challenges. Beyond that, the distinct heterogeneity of learners who often have different play styles, states of knowledge, learning speed, and soft skills, such as teamwork or communication skills, forms a pivotal problem. Apart from that, the vital role of the instructor needs to be taken into account.

Within the scope of this dissertation, the problems mentioned above are analyzed, concepts to solve them introduced, and methods developed to address them. The first major contribution contains the conceptualization of a framework for adaptation of collaborative multiplayer games as well as for the control of those games at run-time through an instructor using the Game Master principle. The core concept hereby addresses the design of a model to represent heterogeneous groups and to represent collaborative Serious Games.

Based on that, a novel concept for adaptation of collaborative multiplayer games is developed, implemented, and evaluated. Automatic recognition and interpretation of game situations, as well as determination of the most well suited adaptation based on the recognized situations, is a major challenge here. Further, a concept is developed to integrate an instructor in a meaningful way into the course of the game, giving him/her the necessary resources to recognize problems and to intervene and adapt the game at run-time. Therefore, it will be taken into account that the elaborated concepts are applicable in a generic way independent of the underlying game.

The second major contribution of this work is the conceptualization and design of a simulation of players and learners in a collaborative multiplayer game that behave realistically based on a player, learner, and interaction model. This is supposed to enable an evaluation of the adaptation and Game Mastering concepts using freely configurable player and learner types.

The concepts introduced and developed within this thesis have been thoroughly evaluated using a twofold approach. As a test environment, a collaborative multiplayer Serious Game was designed and implemented. Within that simulation environment, the developed Game Mastering and adaptation concepts were assessed and tested with large sets of virtual learners. Additionally, the concepts were eval-

uated with real users. Therefore, two different evaluation studies with a total of 60 participants were conducted.

The results of the conducted evaluations help to broaden the areas of application of Serious Games as well as to improve their applicability, hence raising acceptance among instructors. The models, architectures, and software solutions developed within this thesis thus build a foundation for further research of multiplayer Serious Games.

KURZFASSUNG

Lernen und Spielen stellen zwei zentrale Aspekte der heutigen Informations- und Kommunikationsgesellschaft dar. Diese beiden Themen werden im Kontext von Serious Games unter dem Begriff 'digitale Lernspiele' zusammengefasst. Serious Games kombinieren spielerische Konzepte mit einem breiten Anwendungsbereich, darunter unter anderem der Bildungssektor, Training, Gesundheit und Sport, aber auch Marketing, Werbung, politische Bildung und andere gesellschaftlich relevante Themen wie Klima, Energie, oder Sicherheit. Im Fokus dieser Arbeit stehen dabei kollaborative Lernspiele, also digitale Lernspiele, die eine Kombination aus den Konzepten des kollaborativen Lernens und Spielekonzepten darstellen.

Obwohl digitale Lernspiele eine vielversprechende Ergänzung der existierenden Lehrmethoden darstellen, gibt es verschiedene Herausforderungen, die ihrem Einsatz im Wege stehen. Die Spannung zwischen einem Spiel, das Spass macht und der Vermittlung von Lernen durch ein Spiel stellt eine zentrale Herausforderung an das Game Design dar. Die oft hohe technische Komplexität oder insbesondere die mangelnde Kontrolle der Lehrenden über das Spielgeschehen sind weitere zentrale Herausforderungen. Darüber hinaus stellt die Heterogenität der Lernenden, die unterschiedliche Spielpräferenzen, Lernstände, Lerngeschwindigkeiten, sowie Soft Skills wie Teamwork und Kommunikationsfähigkeiten vorweisen ein wichtiges Problem dar. Außerdem muss der zentralen Rolle des Lehrenden in kollaborativen Lernszenarien Rechnung getragen werden.

Im Rahmen dieser Dissertation werden die oben genannten Probleme analysiert und Methoden und Konzepte zu deren Lösung erarbeitet und vorgestellt. Der erste Beitrag umfasst hierbei die Konzeption eines Frameworks zur Adaption von kollaborativen multiplayer Spielen sowie deren Leitung durch einen Lehrenden zur Laufzeit nach dem Game Master Prinzip. Dies beinhaltet die Modellbildung zur Darstellung heterogener Gruppen und kollaborativer Lernspiele.

Darauf aufbauend wird ein neuartiges Konzept zur Adaption von kollaborativen multiplayer Spielen entwickelt, umgesetzt und evaluiert. Die Herausforderung hierbei ist das automatische Erkennen und Interpretieren der im Spiel vorliegenden Situationen und basierend auf den vorliegegenden Situationen die am besten geeignete Adaption zu ermitteln, die den Charakteristiken der Spieler Rechnung trägt. Weiterhin wird ein Konzept erarbeitet, um einen Lehrenden sinnvoll in den Spielablauf zu integrieren, diesem die notwendigen Mittel zur Erkennung von Problemen und zum Eingreifen und Adaptieren des Spielablaufs zur Verfügung zu stellen. Dabei soll darauf geachtet werden, dass die erarbeiteten Konzepte möglichst generisch einsetzbar sind unabhängig vom zugrunde liegenden Spiel.

Der zweite wesentliche Beitrag dieser Dissertation ist die Konzeption und das Design einer Simulation von Spielern und Lernern in kollaborativen multiplayer Spielen, die sich basierend auf einem Spieler-, Lerner, und Interaktionsmodell realistisch verhalten. Dieses soll es ermöglichen, die Adaptions- und Game Mastering-Konzepte mit frei konfigurierbaren Spieler- und Lernertypen zu evaluieren.

Die im Rahmen dieser Arbeit vorgestellten und entwickelten Konzepte wurden auf zweifache Art und Weise evaluiert. Dazu wurde als Testumgebung ein kollaboratives multiplayer Spiel entworfen und implementiert und die entwickelten Game Mastering und Adaptionskonzepte in einer ersten Studie anhand dieses Spieles unter Verwendung einer großen Menge von virtuellen Lernern evaluiert. Darüber hinaus wurden zwei Evaluationsstudien mit insgesamt 60 Teilnehmern durchgeführt um die Konzepte mit realen Anwendern zu testen und bewerten.

Die Ergebnisse der durchgeführten Evaluationen tragen dazu bei, die Anwendungsgebiete von Serious Games zu erweitern und deren Anwendbarkeit zu verbessern und damit die Akzeptanz unter Lehrenden und Lernenden zu erhöhen. Die entwickelten Modelle, Architekturen, und Softwarelösungen stellen somit eine Grundlage für die weitergehende Forschung von multiplayer Serious Games dar.

ACKNOWLEDGMENTS

First of all, I would like to thank Prof. Ralf Steinmetz for his supervision and for giving me the opportunity to work on this exciting topic at Multimedia Communications Lab (KOM).

I would also like to thank my friends, colleagues, and partners at KOM. Particularly, I thank Stefan Göbel for his support and my colleagues of the Serious Gaming group - Michael Gutjahr, Johannes Konert, Florian Mehm, and Christian Reuter - for valuable discussions and advice.

Further, many thanks go to Torsten Zimmermann from AVM Rüsselsheim and Kai Erenli from BFI Wien for the great collaboration during the Serious Game evaluation.

Finally, I would like to thank my mother Hannelore, my father Gerhard, and my brother Christian for their ongoing support during the last years. Without you, this thesis would not have been possible.

Darmstadt, 2015 Viktor Wendel

CONTENTS

1 INTRODUCTION

»It is paradoxical that many educators and parents still differentiate between a time for learning and a time for play without seeing the vital connection between them.«

— Leo F. Buscaglia

Learning and gaming are two facets of life that at a first glance do not have much in common. Yet the vital connection between them is visible in many aspects. For children, playing is often a projection of a part of reality onto a frame of rules determining how to reach a goal. Thus, when children play, they inevitably learn something about the world and its rules and consequences. However, games can not only teach young children about life. Rather, games can teach in various ways. They can teach strategic thinking (e.g., chess), teamwork (e.g., football), memory (e.g., memory card game), vocal skills (e.g., Taboo), reaction (e.g., Uno), and more. Also, when looking at digital games, many computer and video games incorporate effective learning principles and can teach the player a lot while being played [50], even if they are played for the sole purpose of fun. There are reports of surgeons who improved their motor skills by playing games [135, 136]. Playing action video games is known to improve reaction time and decision making [40, 58], attention control [57, 41], and memory [20]. Moreover, (action) games have been reported to have extraordinarily effective reinforcement and rewarding mechanisms, which are considered critical for efficient learning [133]. This phenomenon can best be summarized with the words of the American author and poet Diane Ackerman: "Play is our brain's favorite way of learning."

In the following, the deeper connection between learning and gaming will be outlined in the context of digital educational games as a major field of so-called Serious Games, games with a purpose in addition to pure entertainment. The concept of multiplayer games offers additional advantages and opportunities as it can add a social component to the process of gaming, emphasizing the role of interaction between players and affecting social skills like teamwork or communication. Hence, multiplayer games will be the focus, and the role of the instructor in collaborative multiplayer Serious Games (Section 1.1) will be elaborated. Moreover, challenges of orchestrating and adaptating collaborative multiplayer Serious Games are described (Section 1.2). Based on those challenges, the research goal (Section 1.2.2) and the contributions (Section 1.3) of this thesis will be motivated. Finally, an outline of the reminder of this thesis' organization is provided (Section 1.4).

1.1 MOTIVATION

Education is a key topic in today's information and knowledge society. In a general view on education, the importance of learning on a school level is a much-discussed topic. PISA has underlined the strong need for improved teaching in mathematics, reading ability, and natural sciences [1].

In this context, the importance of the so-called STEM disciplines should be pointed out. The discussion about the importance of the STEM fields - science, technology, engineering, and mathematics - is on the spot. Countries with few natural resources, like Germany, especially are reliant on well-trained, highly qualified employees. Thus, the Institut der Deutschen Wirtschaft emphasizes the importance of availability of employees with a MINT (German equivalent of STEM) focus for companies in the technology sector [10]. However, it also emphasizes the current shortage of employees in MINT fields. In the German OECD study for vocational training, the weak PISA results of many German students in secondary school are identified as one of the reasons why many young people fail to transition from school to vocational training [70]. It has become clear that investments in education and training, especially in MINT fields, are necessary. As early as 2009, former CEO of Porsche AG, Wendelin Wiedeking, stated that "education is the strongest profit-yielding form of investment" [180].

Alternative learning methods have been used in schools, vocational schools, and institutes of higher education for many years, including group work and collaborative learning. For these methods of learning, where learners typically interact with each other, social skills like communication or teamwork are vital. Moreover, the importance of social skills for professional life has been recognized. Thus, today even in technical university courses, soft skills are considered vital and subsequently are part of the curriculum.

Obviously, motivation is one of the key factors of successful learning. Games usually provide high intrinsic motivation. Many players spend hours playing a game for no other reason that playing itself. Moreover, it is not uncommon for motivated players to spend time outside of a game to improve their gameplay, by, e.g., reading guides or discussing game-related topics in forums. Hence, it stands to reason that the motivation that arises from playing a game can be used for the process of learning. The concept of digital educational games follows that assumption. While the idea of Serious Games has broader areas of application than learning - like sports and health, opinion forming, or advertisement - the idea of using games or game technology for playful learning is one of the oldest and best-known application areas of Serious Games. The intention with Serious Games is to combine learning principles and game mechanisms to create games with a purpose other than mere entertainment but with fun still as one of their core components. Learning in Serious Games is intentional and not simply a positive side effect.

Playing games is today a common leisure-time activity, not only among young people. The cliché of young teen male gamers does not apply anymore. In fact, 47% of U.S. gamers are female [49]. According to statista [4], in 2014, 29% of gamers in the U.S. were under 18 years old, 32% were age 18 to 35, and 39% were 36 or older. According to bigfishgames [49], 29% of gamers are even older than 50 years. Altogether, 58% of Americans play video games. For Germany, the BIU - Bundesverband Interaktive Unterhaltungssoftware e.V. states that almost 29.3 million Germans play digital games regularly, more than one third of Germany's population. Of those, 53.9% are male and 46.1% are female [2]. In Germany, 29% of gamers are 19 or younger, 34% are age 20 to 39, and 37% of gamers are 40 or older, more than half of them 50 or older [3]. Thus, it can be stated that gaming has arrived in the middle of society. This broad acceptance is one argument in favor of using Serious Games for learning.

Serious Games have been shown to be one new addition to traditional learning methods that can help to address some problems in learning. While Serious Games cannot replace teachers or traditional teaching - which they never were meant to do - they can fill in gaps where traditional learning methods are weak, or provide an alternative method of learning for students. They provide an inherent motivation that can help otherwise weak students become motivated to learn.

However, Serious Games are still not being used widely. The most important reason for this might be that there is still little evidence of the effect of game-based learning which is a vital prerequisite for educational authorities or school boards to consider using Serious Games in the curriculum. From a market perspective, many game developers and publishers find it rather unattractive to develop Serious Games, owing to weak customer numbers and low development budgets. Another problem in this context is the fact that buyers of Serious Games are usually not users (i.e., the ones who play the games), in contrast to traditional games. Instead, potential buyers are those deciding about the use of Serious Games, like ministries of education and cultural affairs, school administrators, trainers, or parents.

Looking at the games market during the last years, a trend toward Serious Games can be identified. In 2013, the German games market was €2.66 billion strong, according to GAME Bundesverband e.V.[1] According to Gartner, in 2013, $93 billion was spent on games worldwide, with an estimated $101.6 billion in 2014 and $111 billion in 2015.[2]

Likewise, Serious Games have grown out of their niche existence. In 2012, the worldwide Serious Games market was estimated to range from $2 billion to $10 billion.[3] Apart from their field of application in schools, vocational schools, and universities, Serious Games are a promising idea in training and in-house education. Gartner predicts that "by 2015 more than 50% of organizations that manage innovation processes will gamify those processes."

A multitude of concepts and implementations for using Serious Games for learning have been proposed during the last two decades. So far there have been many successful examples of Serious Games combining learning and playing. Yet the majority of those are single-player games. One reason for this might be that designing games for learning in a group adds significant difficulty to the design process. There is no existing model yet for multiplayer Serious Games. This, however, would be necessary for research into the design and applicability of collaborative multiplayer Serious Games. Moreover, it is harder to oversee a group of learners interacting with a common learning environment (i.e., the game) than a group of players playing (and learning) separately, especially in a more complex learning scenario like collaborative learning.

This problem leads to the role of the instructor. The instructor in a multiuser learning scenario (usually the teacher, lecturer, or trainer) generally plays a vital role with various responsibilities and duties. School classes on average consist of about 21-23 (see [120]) students with heterogeneous knowledge, skills, and prerequisites. Thus, teachers are required to teach different students on individual levels, a challenge con-

1 http://game-Bundesverband.de/de/mit-266-milliarden-euro-ist-deutschland-groster-gamesmarkt-in-europa-newzoo-und-g-a-m-e-bundesverband-legen-marktzahlen-fur-2013-vor/ (retrieved: 2015-01-17)
2 http://www.gartner.com/newsroom/id/2614915 (retrieved: 2015-01-17)
3 http://www.hypergridbusiness.com/2012/08/serious-games-now-a-multi-billion-dollar-industry/ (retrieved: 2015-01-17)

sidering the usually tight schedule. In collaborative learning scenarios, the instructor needs to oversee the process of learning, including the actions and interactions of the group of learners. This means a very high cognitive load for the instructor.

1.2 RESEARCH CHALLENGES AND GOAL

Based on the introduction on multiplayer Serious Games and the instructor's role in game-based multiplayer learning scenarios, in this section, three challenges will be pointed out. These challenges represent the identified gap that forms the focus of this thesis. The causes of these challenges are identified and implications pointed out. Based on the identified challenges, the research goal is formulated.

1.2.1 *Research Challenges*

The following three research challenges were identified:

Challenge 1: *Heterogeneity of players and learners.*

A group of players/learners is usually heterogeneous, in terms of both gaming and learning. Players can strongly differ in the way they play games (behavior, style). Different players might prefer different genres of games. Studies show that gaming preferences differ based on gender [98, 28], gaming habits, and age [59]. All of these factors can influence player preferences toward preferred genres, motives, or amount of competition. Likewise, in terms of learning, people can differ widely in various dimensions. Learners differ in learning speed as well as in learning methods. It has been shown [122] that gender affects the learning performance of players, too. Also, learners' state of knowledge might differ (experience). All of these aspects contribute to the possible heterogeneity of a group of learners in a game-based learning scenario.

Challenge 2: *High cognitive load on the instructor.*

A typical collaborative learning scenario is performed in small groups of learners and an instructor. Instructor tasks are manifold. Instructors usually observe learners and learning behavior, analyze, coach, moderate the learning process, and guide the learners. In a game environment, an instructor additionally needs to fulfill these tasks in the context of the course of the game. In addition to analyzing, coaching, moderating, and guiding learners regarding the learning content, the instructor needs to take into account player behavior related to the game. This burdens the instructor with a very high cognitive load as the instructor needs to access and process information about the game in general, the current state of the game, and all individual players and their actions. Moreover, in a multiplayer game, due to the amount of players playing simultaneously, oftentimes a series of events happens within a short amount of time or concurrently.

Challenge 3: *Reluctance toward the use of Serious Games.*

Both among many teachers and among parents, the reluctance to use Serious Games is notable. Some of the distrust in Serious Games might result from a common reluctance towards traditional games which again is caused by various social problems which are believed to be caused by games. Examples for this are the discussion on violent video games or excessive use of video games. The lack of scientific proof of the benefit of the use of Serious Games is another major reason for the reluctance among many parents to use Serious Games in class. Moreover, many teachers refrain from using Serious Games for various reasons. One of the most obvious reasons is the lack of familiarity with the medium caused by technical hurdles and a lack of qualification. Some teachers also state that they fear a loss of control when using games in class [81].

1.2.2 *Research Goal*

Based on the motivation and first research analysis, the research goal of this thesis is to develop an approach to address the challenges described above. The reluctance toward the use of Serious Games among many teachers indicates that from a teacher's point of view, the use of Serious Games in class is far from an optimal state. To address this problem, the usage and application of Serious Games need to be improved for this target group. Hence, not only do the technical hurdles need to be overcome, but the fear of loss of control also needs to be addressed.

Regarding the cognitive load on the instructor during the gaming sessions, methods and concepts for assisting instructors during run-time need to be developed in order to be able to provide required information overseeably.

The heterogeneity of players requires adapting the game to various play styles and gaming preferences, different grades of experience in both game and learning content, and different learning styles. These adaptations can, in principle, be performed either by a human (Game Mastering) or automatically. Both alternatives will be addressed.

To formalize adaptations and adaptation access - both for a human Game Master (GM) and for an algorithmic use - a formal model for the underlying collaborative multiplayer Serious Game needs to be developed.

Hence, the goal of this thesis is to develop a model for an automatic adaptation of collaborative multiplayer Serious Games to the needs and preferences of a group of players/learners through integrating and supporting the instructor. Therefore, methods and concepts for the meaningful integration and support of the instructor need to be designed. Further, to reduce the cognitive load on an instructor mastering a game session, algorithms need to be designed to automatically adapt multiplayer Serious Games to the needs and preferences of a learner/player group at run-time based on various parameters such as difficulty, game pace, and player attributes. A test environment needs to be designed and deployed to be able to evaluate the developed concepts. For this purpose, a concept for simulating player and learner behavior - i.e., simulating different play styles and gaming preferences, states of knowledge, and learning styles - needs to be designed and developed.

1.2.3 *Hypotheses*

Based on the identified goal, the following two hypotheses are formulated.

1.2.3.1 *Hypothesis I - Adaptation of Collaborative Multiplayer Serious Games*

> Automatic adaptation of a collaborative multiplayer game to the needs of a heterogeneous group considering gaming, learning, and interaction improves learning success, game experience, and players' performance with regard to the goal of the game.

This hypothesis refers to the impact of meaningfully adapting collaborative learning sessions. It is expected that having a mechanism of meaningful adaptation of a multiplayer game to the preferences and needs of a heterogeneous group will have a positive influence on learners'/players' learning success, learning behavior, gaming success, and game experience.

1.2.3.2 *Hypothesis II - Game Mastering*

> Providing an instructor in a collaborative multiplayer Serious Game with technology to assess the game process and player information and to adapt the game according to the instructor's professional opinion improves learning success, game experience, and players' performance with regard to the goal of the game.

This hypothesis refers to the impact of the instructor on a game-based learning session when exercising his/her responsibilities within the learning scenario. It is expected that the instructor's ability to exert those responsibilities depends directly on how well the game allows him/her to influence the learning scenario. Thus, it is expected that having a suitable mechanism providing him/her with the necessary means to exercise his/her responsibilities in the desired way has a positive influence on players' learning success and learning behavior, but also on their gaming success and perceived game experience.

1.2.4 *Research Questions*

The resulting research questions are derived from the above hypotheses.
Regarding Hypothesis I, it needs to be clarified how a multiplayer game can be adapted automatically in a way to optimize player and learner performance of a group of heterogeneous players.

Research Question 1: *How can the most well suited adaptation of a multiplayer (Serious) game be determined depending on a given game situation with regard to players' traits, levels of knowledge, learning styles, and interaction?*

Research Question 2: *Can a positive impact on players' learning, gaming, and interaction performance be measured when automatic adaptation is used compared to a session without automatic adaptation?*

Regarding Hypothesis II, it needs to be clarified how a GM can obtain relevant information about a game session and how the game needs to support the GM in his/her intention to adapt the game.

Research Question 3: *How can a Game Master get the required information from a collaborative multiplayer (Serious) game and adaptation mechanisms to manipulate the game, considering the players and the current state of the game?*

Further, it should be considered how the information and adaptation possibilities are presented in a meaningful way to the GM. This, however, is a problem of presentation of information, which is rather an HCI research topic and not the focus of this thesis. But for prototypes, it will have to be considered how information is presented to the GM. The effect of an instructor using Game Mastering technology to oversee a gaming session needs to be measurable to be able to make assumptions about the impact of the Game Mastering concept.

Research Question 4: *Can a positive impact on players' learning, gaming, and interaction performance be measured when a Game Master using appropriate technological support is orchestrating a game compared to a session where an instructor oversees the learning/gaming process without Game Mastering support?*

1.3 CONTRIBUTIONS

This thesis describes new concepts and mechanisms for orchestrating and automatically adapting multiplayer games for collaborative learning considering a heterogeneous group of learners. The concepts are implemented and evaluated as Serious Game prototypes. Figure 1 shows the contributions and major outcomes as well as the interdependencies between the single contributions. The contributions and major outcomes are structured in five layers (analysis, models, concepts, application, and evaluation). The major outcomes are the developed group model and the collaborative multiplayer Serious Game model on the model level, the adaptation framework GameAdapt.KOM and the concept of player simulation on the concept level, and the Serious Game *Escape From Wilson Island*, as well as the implementations of GameAdapt.KOM and the player simulation on application level.

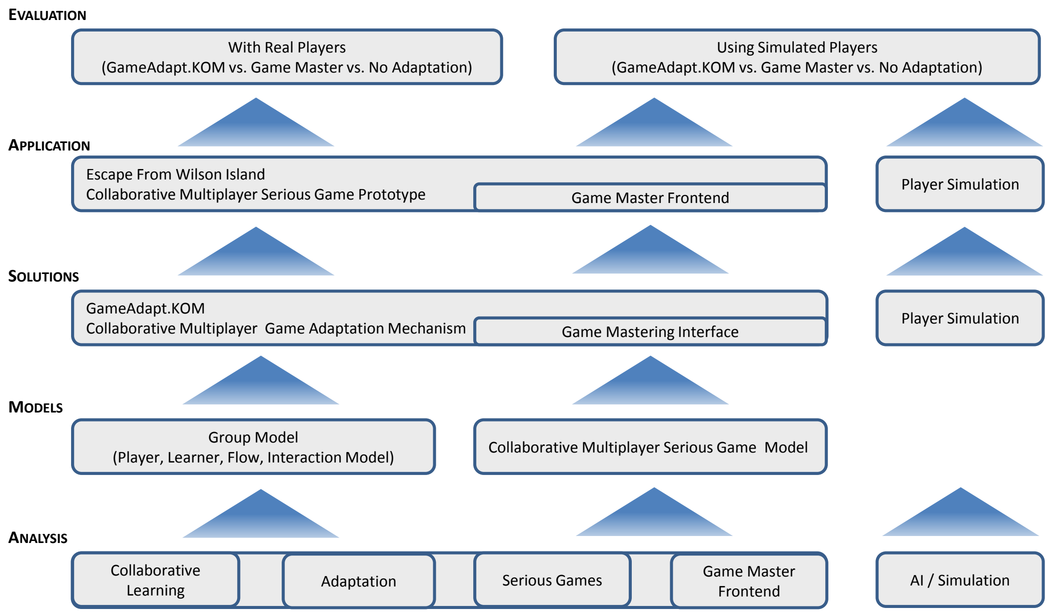

Figure 1: Composition and interdependencies of the contributions of this thesis grouped into layers: analysis, model, solution, application, and evaluation.

1.3.1 *Analysis*

The research fields and dedicated research and technological development aspects which are considered fundamental for the research in this thesis were investigated and the state of the art was determined, including the latest collaborative learning concepts and methods, especially those considering the role of the instructor in collaborative learning scenarios. The field of Serious Games was examined with a focus on multiplayer Serious Games and Serious Games design. In the intersection of these two fields is the field of collaborative gaming. Collaborative gaming concepts were examined and compared. Further, the state of the art in the fields of adaptation in games and in the context of learning was examined, as well as existing concepts for Artificial Intelligence (AI) in games and for simulating players and player behavior in games.

1.3.2 *Models*

This layer contains the design and development of a formal descriptive model of collaborative multiplayer Serious Games, including a specification of the underlying genre and of relevant game elements, their roles and functions in a game, and the relations and interdependencies between them.

A unified model for representing a group of heterogeneous players/learners in a collaborative multiplayer Serious Game was designed and developed. The model represents both features and qualities of a group of players/learners in the dimensions of learning, gaming, challenge, and interaction.

1.3.3 *Concepts*

A framework for adaptation - both automatic and by a GM - was designed. The framework, GameAdapt.KOM, includes a formalization of adaptation in collaborative multiplayer Serious Games, metrics for calculating player performance in the context of gaming and learning, methods and metrics for recognizing and classifying game situations, and an algorithm for selecting adaptations based on specific situations and the group model. Further, the GM interface was developed as a component of GameAdapt.KOM. A concept for simulating players with configurable player, learner, and interaction models was developed.

1.3.4 *Application*

A collaborative multiplayer Serious Game prototype, *Escape From Wilson Island (EFWI)* was designed and developed. The prototype was used as a foundation to implement the proof-of-concepts of the automatic adaptation mechanism and the Game Master front-end. The automatic adaptation mechanism and the GM interface were implemented as an extension of EFWI.

1.3.5 *Evaluation*

Two evaluations were carried out to determine the impact of the automatic adaptation on player performance and game experience. Three groups were compared:

1. Automatic adaptation
2. Adaptation by a Game Master
3. Reference group without adaptation

The first evaluation was carried out using a set of simulated players, which allowed the configuration of desired player, learner, and interaction models that would not have been possible with real players. For the second evaluation, real players/learners were selected to play the game, which allowed for evaluation with realistic player, learner, and interaction models and confirmation that the results are valid under real-world circumstances.

1.4 Thesis Organization

The remainder of this thesis is structured as follows: In Chapter 2, a description of the thesis' scenario is provided. This includes an explanation of small collaborative learning groups, the role of the instructor in such learning settings, and game-based learning groups. Moreover, basic foundations considered necessary for understanding this thesis are explained. Further, an introduction to the topic of Serious Games is provided, including Serious Games application areas, the use of Serious Games, using Serious Games in the classroom, and the research context of this thesis, namely the Serious Games group at the Multimedia Communications Lab (KOM) at at Technische Universität Darmstadt.

Chapter 3 gives an overview of related state-of-the-art topics that form the technical background for the work presented in this thesis. The concept of collaborative learning is covered in relation to computer-supported collaborative learning, communication, the role of the instructor in collaborative learning scenarios, and the importance of teamwork in collaborative learning. For a general understanding of Serious Games and especially the design of a Serious Game prototype, Serious Game design challenges are elucidated, as well as collaborative gaming and evaluation of Serious Games. An introduction to agent-based simulation of players is given, as developing the model requires that configurable players be simulated. Moreover, existing Game Mastering concepts will be presented along with work that has been done in the field of adaptation. Derived from the topics presented here, the identified research gap will be motivated at the end of this chapter. The overall concept for Game Mastering and adaptation of collaborative multiplayer Serious Games is presented in Chapter 4. The development of the collaborative multiplayer Serious Game model is described, including limitations of validity based on game genre. The core game components are defined and their purpose and use in the model described. Further, the collaborative group model is conceptualized based on a model for each of the four dimensions of learning, gaming, flow, and interaction.

In Chapter 5, the GameAdapt.KOM adaptation mechanism is explained in detail. Its overall functionality is presented, as well as its development based on an analysis of requirements, including the definition of the adaptation goal, the concept of player performance calculation, the recognition of game situations, and an explanation of the game interface. The adaptation selection algorithm is further explained, followed by the system architecture.

The Game Mastering interface conceptualization is described in Chapter 6. As a first step, requirements for a GM interface are derived from the literature. Based on that, the concept of giving the GM the required real-time information about the game, and providing the GM with run-time adaptation possibilities is explained. In addition, the integration of the GM interface into GameAdapt.KOM is shown.

To satisfy the need for an unlimited and configurable (considering learner, player, flow, and interaction models) set of players to test the concepts, in Chapter 7 the concept of simulating realistic player behavior based on a player, learner, and interaction model is described. The agent-based player simulation approach is motivated and the simulation execution procedure illustrated. Further, player goals and plans are defined in the context of *Escape From Wilson Island*.

In Chapter 8, the implementation of the Serious Game prototype is explained in detail. This includes details about concrete information and adaptations for the implemented collaborative multiplayer Serious Game. Moreover, the chapter describes how the Game Mastering concept and player simulation were implemented as a further development of the Serious Game prototype.

To prove the validity of the concepts presented in the previous chapters, they are evaluated in Chapter 9. First, the validity of the player simulation concept is evaluated to confirm that player agent behavior is consistent with the configured player, learner, and interaction models and that the resulting behavior is reasonable based on those configurations. Second, the adaptation mechanism and the GM interface are evaluated using a set of simulated player agents. Here, the goal is to assess the respective impacts of the automatic adaptation mechanism and of the GM using the GM interface to adapt the game on player performance, learning success, and interaction of the simulated players, compared to a reference group without any form of adaptation. In a third evaluation, this process is repeated with real players to show that the results are valid under real-world circumstances.

The final Chapter 10 summarizes this thesis, highlights its contributions and major findings and provides an overview of future work in the field of Game Mastering and automatic adaptation of collaborative multiplayer Serious Games.

2

FOUNDATIONS

»Play is the beginning of knowledge.«

— George Dorsey

The following chapter provides an overview of foundations which are relevant for the work presented in this thesis. This includes a description of the scenario for which this thesis is relevant, focusing on small collaborative learning groups, the role of the instructor, and game-based learning groups (Section 2.1). After that, an introduction into the topic of Serious Games is provided covering its application areas and the use of Serious Games (Section 2.2).

2.1 SCENARIO DESCRIPTION

This thesis focuses on learning scenarios in small, possibly heterogeneous groups of players/learners.[1] Heterogeneity refers to the state of knowledge, learning, and gaming preferences of the group. It is assumed that all learners of the learning group are in the same room and thus can communicate with each other verbally. Otherwise, it is assumed that the players have access to comparable communication methods, such as video conferencing or voice communication tools. The presence of an instructor, also referred to as teacher, trainer, or GM, is mandatory. The learning scenario described is a *collaborative learning* scenario. Thus, communication and interaction are important. This scenario can typically be found in learning groups in classes at school, in a university setting, in corporate in-house training, or in vocational schools.

2.1.1 *Small Collaborative Learning Groups*

The scenario described in this thesis is based on small collaborative learner groups. We denote group sizes as *small* if they consist of three to eight learners. *Groups* are characterized as follows: (see [39], p. 492).

- There is an ongoing (possibility of) communication.
- There is an inner structure of the group and a demarcation against the environment.
- There is a feeling of solidarity in the group.
- There is collaboration and mutual support.

To optimize learning conditions, the composition of the group of learners needs to be taken into account. In peer education scenarios it is most relevant to consider each individual's prerequisites in personality and proficiency for matching the peers

1 The terms *learner* and *player* are used interchangeably in this thesis as the users playing a Serious Game are players and learners simultaneously

for learning groups. Konert et al. developed an algorithm for optimizing groups of learners with heterogeneous abilities and skills [88]. However, in this thesis, the composition of the group of learners is considered to be given.

The scenario is further focused on *collaborative learning*. This means that members of the group (should) work together on a joint problem or learning goal. The techniques involved are based heavily on communication and interaction with other players. Group members are mutually dependent on each other to the extend that the success of the group work (i.e., outcome) depends on all members. Usually, group work is used when knowledge is to be developed by the learners themselves rather than taught by the teacher/trainer.

The learning content focused on in this thesis is any form of subject material. This includes typical school learning content (such as history, vocabulary, or mathematics), content taught in vocational schools (metal working, cooking, etc.), or content specific to corporate in-house training.

2.1.2 *Role of the Instructor*

Although not part of the core learning group, instructors hold a vital position in collaborative learning scenarios. Their tasks include moderation, monitoring, coaching, analysis, and intervention ([61], p. 30). Usually, it is the instructor (the teacher or trainer) who decides on learning goals and prepares group composition and task separation ([61], p. 51). Instructor tasks can be grouped in two categories: observing tasks and intervening tasks. The former assess the group and its performance, trying to extract insight into the learning process; the latter aim to optimize this process by coaching, guiding, helping, etc. (compare Figure 2). Hence, the instructor has a vital role in the learning process. The role of the instructor is elaborated in Section 3.2.3.

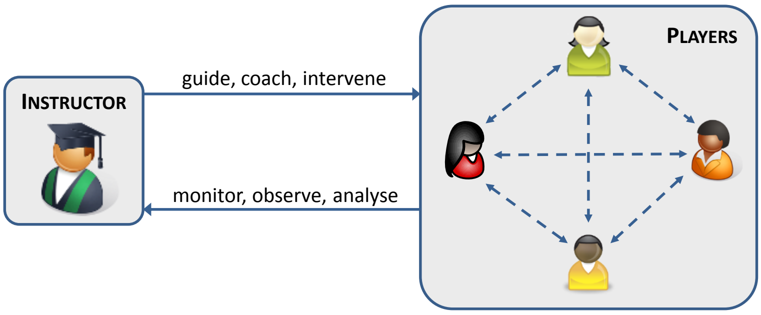

Figure 2: Communication graph for typical collaborative learning scenarios

2.1.3 *Game-based Learning Groups*

The scenario described in this thesis focuses on *game-based* learning groups. This means that the collaborative learning groups use game-based approaches - more specifically, digital educational games - for learning. We focus on team-based multiplayer games as they, by design, are made for a group of players playing together (as a team). This type of game already incorporates many of the social skills and features desired by a well-working collaborative learning setting: communication, teamwork, coordination, or a common goal. The group work focuses on combining the concepts of collaborative learning with the features of multiplayer games.

2.2 AN INTRODUCTION TO SERIOUS GAMES

The term *Serious Games* is defined in the following section. Moreover, an overview of Serious Games application areas is provided and the focus of this work put into context.

2.2.1 *Serious Games Definition*

> "All Serious Games are games; i. e. analogue to any other (pure entertainment) games Serious Games contain game play, goals and rules and use game technology. These elements are combined with further domain-relevant methods, concepts and technologies, e. g. pedagogic and didactic concepts for educational games or sensor technology for exergames and are applied within a broad range of Serious Game application fields."
>
> (Göbel et al. [54, p. 1])

In literature, different definitions of the term *Serious Games* can be found. One of the most prominent definitions is that: Serious Games are games that "do not have entertainment, enjoyment, or fun as their primary purpose" [112]. This implies that all Serious Games are games [54] as they rely on the same basics and mechanics as other games in terms of gameplay, rules, and technology (see [139]).

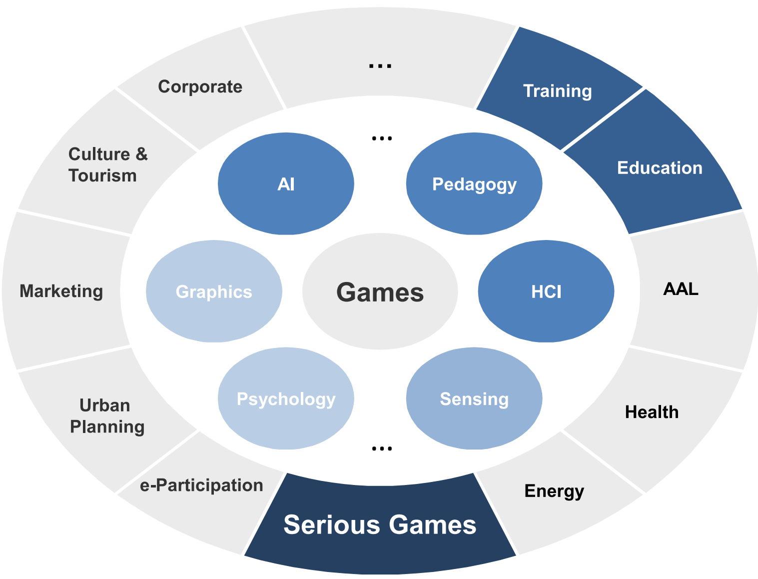

Figure 3: Serious Games application areas focused on in this thesis (image based on Göbel et al. [53])

Figure 3 shows how Serious Games combine game technology (in the center) with domain knowledge (inner circle) for various application areas (outer circle). This thesis focuses on Serious Games for education and training (highlighted blue on the outer circle). The (collaborative) learning concepts relate to the area of pedagogy.

The GM frontend tackles the field of Human-Computer-Interaction (HCI), although this is not the focus of this thesis. The concepts of automatic adaptation and player simulation are settled in the field of AI, among others.

Serious Games technologies are used for various purposes, such as as 3D training and simulation environments for emergency response teams (firefighters, medical staff, police) but also for pilots or bus drivers, or to train guards or service staff. They are also used as visualization and construction tools, e.g., for architecture or urban planning. Serious Games concepts are used in an educational context to support learners and teachers in educational settings at school or university, or in a health context to motivate people for sports, healthy nutrition, or other health aspects of life. Other examples of Serious Games focus on culture and cultural heritage, public awareness of societal issues (e.g., religion, politics, security, energy, climate), or assessment of human behavior and experience in complex and dynamically changing environments.

	Games for Health	Advergames	Games for Training	Games for Education	Games for Science and Research	Production	Games as Work
Government & NGO	Public Health Education & Mass Casualty Response	Political Games	Employee Training	Inform Public	Data Collection / Planning	Strategic & Policy Planning	Public Diplomacy, Opinion Research
Defense	Rehabilitation & Wellness	Recruitment & Propaganda	Soldier / Support Training	School House Education	Wargames / Planning	War Planning & Weapons Research	Command & Control
Healthcare	Cybertherapy / Exergaming	Public Health Policy & Social Awereness Campaigns	Training Games for Health Professionals	Games for Patient Education and Disease Management	Visualization & Epidemiology	Biotech Manufacturing & Design	Public Health Response Planning & Logistics
Marketing & Communication	Advertising Treatment	Advertising, Marketing with Games, Product Placement	Product Use	Product Information	Opinion Research	Machinima	Opinion Research
Education	Inform about Desease / Risks	Social Issue Games	Train Teachers / Train Workforce Skills	Learning	Computer Science & Recruitment	P2P Learning Constructivism Documentary?	Teaching Distance Learning
Corporate	Employee Health Information & Wellness	Customer Education & Awareness	Employee Training	Continuing Education & Certification	Advertising / visualization	Strategic Planning	Command & Control
Industry	Occupational Safety	Sales & Recruitment	Employee Training	Workforce Education	Process Optimization Simulation	Nano / Bio-tech Design	Command & Control

Figure 4: Serious Games taxonomy recreated after Sawyer and Smith [142]. Cells highlighted in dark blue are in the focus of this thesis. Cells highlighted in light blue are not in the focus but still relevant for this thesis.

In 2008, Sawyer and Smith published a preliminary Serious Games taxonomy [142]. They arranged types of games over areas of application to show the various types of Serious Games and the areas in which they are used (see Figure 4). From the taxonomy, one can see the following:

1. There are heterogeneous types of Serious Games, such as games for health, advergames, or games for education.
2. There are heterogeneous fields of application, such as defense, industry, health care, or education.

Each of them has different requirements for the related Serious Games depending on type and field of application. Obviously, exergaming (Serious Game for health in healthcare) is profoundly different from a Serious Game for learning.

The areas that are most relevant for this thesis are highlighted in blue; areas that are borderline relevant are highlighted in light blue. These are mainly games for training and games for education used in the areas of education, corporate, and industry, with some minor adjacent areas.

2.2.2 *Serious Games Challenges and Research Areas*

Serious Games are a highly complex scientific area considering the multifaceted characteristics of pure digital games plus the dimension of the serious part: The key challenge of Serious Games is to reconcile and balance true gaming experience on the one hand and the fulfillment of the additional purpose beyond pure entertainment, on the other. Thus, research in Serious Games is necessarily multi-disciplinary, and most of the currently available systems are specifically designed for a particular target application area. Such solutions for specific application areas have to be subjected to formative and summative evaluations considering the complex interplay of numerous factors.

Research objectives include an in-depth analysis of Serious Games, and the elaboration of new methodologies for (1) efficient, single-user or collaborative authoring of Serious Games, (2) personalized, adaptive, and context-sensitive control, and (3) empirical vs. objective, technology-enhanced evaluation of serious games.

The Serious Games research group at Technische Universität Darmstadt aspires to synthesize these objectives in a reference model for the description and evaluation of serious games, with the option to serve as a quality label in the long-term perspective [54, 110, 90, 55].

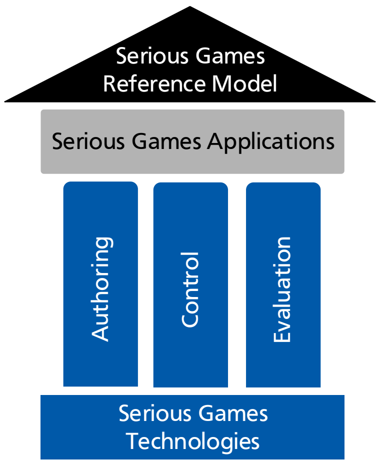

Figure 5: Serious games research areas.

Figure 5 shows the overall structure of research on Serious Games. The columns represent the above-mentioned three key research areas, authoring, control, and evaluation, which are essentially driven by an interdisciplinary collaboration of researchers in computer science, engineering disciplines and humanities as well as subject matter experts in the application domains (defining and further elaborating the foundational Serious Games technologies). Both authoring and control are adressed in the context of single- and multiplayer games. Evaluation refers to both measurable effects in Serious Games and affects of users playing them. A more detailed explanation of the three columns is provided in [53, 54].

In the key area *authoring*, the focus is on game modeling, metadata format design and author assistance, corroborated by results of conducted evaluations on usability aspects [108, 109]. Further, multiplayer tasks in Serious Games are adressed, respectively the authoring support to create such tasks (to be classified as part of the *authoring* research field in Figure 5). This includes the Serious Game authoring aspects of interdependency definition and visualization, solvability of multiplayer tasks, testing framework design and multiplayer tasks design patterns [129, 128].

In the key area *control*, the design and evaluation of adaptive and personalized mobile Serious Games for education and cognition are investigated making use of Smartphone capabilities and sensors for improving user experience and maximizing benefit [145, 146].

Further, the use of social media for peer education in single-player educational games is investigated, with one major focus on optimizing knowledge exchange among peers in social media applications and Serious Games [90, 88, 91].

Considering control in the multiplayer field, one focus is on instructor support in multiplayer collaborative Serious Games using game mastering concepts combined with methods from the fields of collaborative learning and collaborative gaming. This includes the identification of relevant parameters and visualization aspects for Game Masters (instructors), partly automated adaptation of the gameplay and interfaces as well as the modeling of meaningful metrics for Game Masters to assist in the human-based adaptation of game experience [175, 176, 177, 178].

Another major research focus in the application fields sports and health are exergames. Here, the focus is on the collection and interpretation of sensor data for game adaptation in the field of Serious Games for sports, health and rehabilitation. Besides the research on Serious Games *control*, this includes measurement of training effects as impact of the used games by *evaluation* [66].

In this context, one major focus is the research on how sensor technology can act as an enabler for novel concepts in indoor training games such as virtual cycling games. For this purpose, it is investigated how the degrees of freedom in such games can be increased in order to enhance training effects and user experience. In addition the development of metrics for the appropriateness of sensors and sensor data for indoor training games is of relevance.

In the key area *evaluation*, the assessment of Serious Games effects is focussed. This includes the development of methods and concepts to measure user experience (e.g., the questionnaire for serious games [55]) and to investigate emotional states and game element effects on vital parameters. In this context, the challenges of mobile, pervasive Serious Games are further adressed by investigating the applicability of persuasive Serious Gaming for several fields to identify the most relevant field. Strongly related research topics here are game design, sensor data integration, and context awareness as well as interaction and awareness design and end-user studies.

Similar to the knowledge media research group at KOM, the scientific approach of the Serious Games research group is characterized by this interdisciplinary work in order to elaborate, test and validate the research methods and concepts in the context of realistic scenarios (real settings with clear goals, human beings as users and realistic data, referred to as Serious Games applications in Figure 5). Subsequently, the interdisciplinary research with evaluations in different application areas constitutes and further shapes the targeted reference model.

The Serious Games research group at the Multimedia Communications Lab (KOM) defines the understanding and approach to Serious Games as the use of game technologies as a basis, supplemented and enriched by the knowledge and models from related interdisciplinary fields around, to be applied in manifold application areas.

Reference examples for Serious Games cover a broad range of application domains including educational settings (from kindergarten to collaborative workplace training), sports and health applications (prevention and rehabilitation) or other societal relevant topics.

2.2.3 *Use of Serious Games*

The use of games for serious purposes has been suggested many times and for many years. Prensky [125] argues that "a sine qua non of successful learning is motivation." He further argues that most learning content is not motivating at all, but rather boring. In contrast to this, games inherently provide strong motivation through fun, challenge, or competition. Therefore, he suggests we should "merge the content of learning and the motivation of games."

As Mitchell et al. [115] summarize in their literature review, games "motivate via fun ...[] via challenge and via instant visual feedback ...[]". They also provide a list of elements that make computer games engaging and how those characteristics contribute to players' engagement. For example, computer games are usually *fun*, making players feel *enjoyment* and *pleasure*. Computer games provide *goals*, which results in *motivation*. Computer games often require *problem solving*, which *sparks creativity* among players. Sancho et al. [140] examine the influence of gaming on motivation for learning. Their findings indicate that gaming is indeed a powerful motivator for learning.

Apart from spending a lot of time playing the actual games, many dedicated players engage in activities around their games, like reading and contributing to forums and wikis about their games, showing that they are willing to invest a lot of time in a motivating activity. Consequently, Gee [51] states that "the designers of many good games have hit on profoundly good methods of getting people to learn and to enjoy learning," because many players willingly spend a lot of time 'learning' how to get better at a game. Apart from motivation, Gee [50] states "games incorporate a whole set of fundamentally sound learning principles," like giving information on demand, just in time, and not out of context.

In order to master a game, a player needs to learn its game mechanics. Thus, whenever a player improves, he/she 'learns' something, be it an understanding of an important game mechanic, the development of a superior strategy, or motor skills. Good games provide players with sound ways to 'learn' how to improve in order to master the game using "well-established principles and models of learning" [169]. An important factor about learning in games is that "learning takes place within a meaningful (to the game) context" [169]. Games can create an environment in which the content to be learned can be placed in a meaningful context where it can be practiced. According to Mansour and El-Said [103], Kirriemuir and McFarlane [85], and

Squire [150], Serious Games can add several educational advantages such as training of soft skills (communication, negotiation skills, teamwork, and collaboration) or strategic thinking or planning: “Video-gaming is an activity that naturally engenders learning processes that contribute to the development of the player in many areas” [45].

Independent from what is trained with it, a game always provides a safe environment for practice without having to fear consequences. Moreover, failing in a game means only a short setback. In a well-designed game, the player is shown the reason(s) for his/her failure, enabling him/her to analyze and learn from mistakes. Also, the player will be able to retry within a short time span.

Another aspect is the fact that games enable players to “try out alternative courses in action in specific contexts and then experience consequences” [141]. With respect to the previous argument, this refers not only to the possibility to make mistakes, but also to the opportunity to try ’what-if’-scenarios. This is pointed out by Herz [69] especially in the context of historical scenarios. Simulation games like *Sid Meier's Civilization* or *Sid Meier's Gettysburg* have been used to replay historical scenarios and encounter alternative ways in history to learn historical implications [152, 69, 56]. Generally, simulation games are a very promising game genre for this kind of ’what-if’ learning experiences as they allow players to vary parameters and observe their effect: “Simulations are an established method of demonstrating and modeling within a range of educational and working environments” [85]. Owing to their ability to enable players to examine complex systems in a simplified way and manipulate and observe them, simulation games are one of the most prominent genres of Serious Games [172]. It has also been pointed out that players can learn different skills like information interpretation, reasoning, and strategic thinking by playing strategy games [85].

There is also a multitude of examples of traditional Massively Multiplayer Online (Role-play) Games (MMO(RP)Gs) being used as Serious Game environments. Delwiche performed learning units of an undergraduate communication course using the MMORPG *Everquest* and the sandbox game *Second Life* [33]. Childress and Braswell investigate the use of MMORPGs to foster communication and interaction and to facilitate cooperative learning [27]. Steinkühler addresses social aspects of learning of learning with and within MMO(RP)Gs [155].

An interesting characteristic of MMORPGs is the motivation they give players to spend time with a game without even playing it. Many dedicated players are willing to ’learn’ facts about the game helping them to become better at playing it, or train skills relevant for the game outside of the game itself, using third-party tools, forums, wikis, or other sources of information. In this context Bergsträßer [18] proposed a framework for context-aware interaction between players and virtual worlds offering a structured approach to providing information from a game set in a virtual world (e.g. an MMORPG) to players.

Despite the many advantages of Serious Games, drawbacks have been outlined, too. Different barriers related to Serious Games should be pointed out, based on Groff and Mouza’s [60] (cited in [86]) barriers to innovation in the classroom.

Those are

- research & policy factors,
- district/school factors,
- teacher factors,
- technology-enhanced project-related factors,
- student factors,
- technology factors.

The challenge in terms of policy is that "many educators do not feel they have the ability to develop rigorous, integrated, technology-based projects while still working towards the goals of annual state testing." Regarding research, there still is no exhaustive evidence of the improved outcome of using Serious Games. On district and school level, the implementation into the school curriculum is not trivial as many different parties are involved (amongst others parents) who might oppose the concept of game-based learning. Teachers are often reluctant to the use of novel technology which might challenge their role in the classroom. Other problems might occur, if the use of Serious Games results in distance from the school context and depends on resources which are outside of the teacher's control. Regarding students, it is required that they are comfortable with the use of the technology which might imply a larger workload with different, often open-ended tasks. The use of Serious Games is often closely related to other technologies (like the availability of an Internet connection), or hardware factors which might impact the implementation of Serious Games at schools.

One of the main problems of Serious Games is the fact that unlike traditional commercial video games, they are not usually bought by the users. While the game might be attractive for the ones playing it, the decision makers might not consider it educationally relevant. On the contrary, a Serious Game which is attractive to parents, or teachers, due to its well-integrated educational content, might be unattractive to the ones who are intended to play it. For those decision makers it is important to have proof of effect before buying Serious Games. Either parents buy games for their children, or in a school context, the decision to buy Serious Games is made centrally by the principal or the ministry of education, rarely by a teacher. However, significant results showing the positive effect of Serious Games (for learning) are still rare. On top of this, schools budgets are often a problem [86]. Moreover, as pointed out by van Eck [169], the technical challenges of digital game-based learning technologies are significant:

"Teachers are the gatekeepers of instructional technology" ([32], p. 37). Therefore, it seems necessary to incorporate teachers in Serious Games design and support them during Serious Games execution. This means reducing the technological barriers associated with utilizing, maintaining, and controlling a Serious Game in class.

2.2.4 *Serious Games in the Classroom*

Digital educational games have been used in schools for more than 20 years. Early examples were mostly drill-and-practice games for, e.g., vocabulary training. Other traditionally used Serious Games for learning can be described as learning environments with playful elements, where those playful elements were used mainly as a reward for successfully solving a learning task. Often the playful part was implemented in form of a mini-game (e.g. *Oregon Trail* or *Math Blaster*). Other examples are *2weistein*[2], and *Winterfest*[3]. A more advanced example of a Serious Game to be used in classroom, is the Serious Game *Ludwig*[4] which was developed for physics teaching based on the curriculum of 5th to 8th grade. *Eduventure II* is an approach to subtly incorporating knowledge into a game by embedding the knowledge into a historical first-person adventure [174]. Mildner et al. designed a Serious Game for architectural knowledge in the classroom using a story-based approach to teach the learning content [114].

A different approach is the use of existing games and game environments for teaching. This includes modding of existing games, (e.g. *Minecraft*, *Neverwinter Nights*, *Skyrim*, *Civilization IV*, *Civilization V*) but also using game environments as tools for learning beyond their original purpose as games (all of the examples of MMO(RP)Gs above apply [33, 27]).

From a report of the Bundesministerium für Bildung und Forschung (BMBF) [113] about the usage of digital media in German schools, it can be concluded that teachers generally use digital media in class only rarely. The BMBF argues that there might be a lack of motivation among teachers to use such tools. Concerning the use of games especially in class, several obstacles have been identified by Kirriemuir and McFarlane [85]. They argue that the limited time available requires games that can be used without losing time for setup, play testing, or dealing with technical problems. Moreover, the game must be verified as suitable to be used for the content to be taught. Furthermore, they point out that teachers rarely have the required time to familiarize themselves with the educational game components. This, among many other obstacles, might be an explanation for the reluctance among teachers to use digital educational games in class.

McFarlane et al. [106] summarize several requirements for Serious Games to be used in class: They state that "it is important for teachers to have some kind of record of what each group has done during a session of gaming." Moreover, games should be able to be adapted to different ability levels. Games should have an option to save the game progress at random points for more flexibility when using them in a tight schedule. Regarding learning outcomes, they define three types:

1. Learning resulting from tasks stimulated by the content of the game
2. Developing knowledge through the game content
3. Developing skills by playing the game (directly and indirectly)

Regarding the development of skills, it is important to point out that not only skills that are directly related to the context of the game can be trained, but also generic soft skills like cooperation, collaboration, teamwork, and peer tutoring. Squire points out that it is hardly possible to motivate all students of a class equally with one game. He describes an experiment of playing the simulation game *Civilization III* in class [151]. The findings show that the game was motivating and fun for some of the students, but too complicated and uninteresting for others. Also, in terms of failure, huge differences could be observed. For some students, failure caused frustration, whereas others regarded it as a learning opportunity.

2 http://www.2weistein-training.de/
3 http://www.lernspiel-winterfest.de/
4 http://www.playludwig.com

Summarizing the use of Serious Games in school, we can see that although there are many examples of games being used in a school context, there are still many obstacles and practical barriers -namely, the need for adaptation in terms of difficulty and content, the support of teachers while orchestrating the game and in assessment after the gaming session, and acceptance among teachers and parents.

3 RELATED WORK

»I don't love studying. I hate studying. I like learning. Learning is beautiful.«

— Natalie Portman

The core focus of this thesis is on adaptation and Game Mastering in collaborative multiplayer Serious Games. Several research fields contribute to the results of this thesis. The field of Game Mastering in collaborative Serious Games is an interdisciplinary work settled in between the fields of computer science, game theory, and pedagogy. Although the main contributions of this thesis are clearly settled in the field of computer science, especially simulation and adaptation of games as well as AI, it is profoundly based on findings from the fields of pedagogy, and game theory. In this context, game design (Section 3.1) was identified as a major related field, as well as collaborative learning from a pedagogical perspective (Section 3.2). The research field of adaptation in games is further investigated in Section 3.3, followed by a closer look into AI (Section 3.4), and finally Game Mastering concepts and their application in digital games in Section 3.5.

3.1 SERIOUS GAMES DESIGN

Game Design is a term describing the art or craft of creating a game through the use of design and concepts which define gameplay, game rules, goals, and challenges. Until today a multitude of books on game design have been published, covering various aspects like game technology, target group, genre, or narration [139, 30, 6, 143, 137].

However, the techniques and design approaches described there do not cover all aspects of Serious Games design. Serious Games are games with a purpose other than just fun. This means, Serious Games not only need to fulfill the requirements for good games, but on top of that need to be designed in a way such that they incorporate the learning content into the game design in a meaningful way. Harteveld [67] approaches the design of Serious Games in form of a 'Triadic game design', combining the three dimensions *reality, meaning*, and *play*. Reality here refers to the Serious Game's subject, the purpose other than fun. Meaning refers to the creation of a value, like motivation, relevance, and transfer. Play refers to those elements that make the game fun to play.

Bergeron [17] describes the different types of Serious Games (military, advertisement, learning, etc.) and how they differ. He further covers Serious Games business aspects. He elaborates the fact that unlike in other video games, in Serious Games players are usually not the ones buying the game.

3.1.1 *Design Challenges*

A major challenge when designing Serious Games is the integration of the learning content into the game(play). For this purpose, it is important to consider the learning content and its impact on the game design from the very beginning of the design process and to make sure that the different parties involved (subject matter experts, game designers, programmers, artists) get into discussions about the integration of the learning content. Kelly et al. [80], when designing the Serious Game *Immune Attack* - a Serious Game for biology learning - formed an advisory board consisting of subject matter experts (immunologists), instructional designers, experts in educational technology, and game designers. They developed a work-flow integrating all those experts from different fields into the design process.

In terms of integrating learning content into a Serious Game, another major challenge is how to integrate it in a seamless way. Many Serious Games suffer from poor integration, leading to an unnatural division between the learning and the gaming part of Serious Games. Examples of this problem are early Serious Games which can be described as learning environments asking questions and rewarding correct answers with short intersections of playing. Those interruptions are often perceived as not fun. Wechselberger [174] describes an approach of creating a Serious Game for history learning. In order to avoid the problem described above, a gameplay was created where the player is set into kind of a role-play game settled in the relevant era of history. The player is able to find out about relevant facts by being part of the story. His findings indicate that this is perceived much better from players as the learning happens seamlessly and no unintended interruptions are felt. However, it was harder to ensure that relevant knowledge is gained.

Another important design question is the target audience. Will the game be designed for young adults (e.g., students), pupils, or younger children? Are there special conditions to be regarded, like learning disabilities or ADHS? Sánchez et al. [140] designed a Serious game for children with serious communication problems like autism, dysphasia, ictus, or cerebral palsy. Their game design is based on the factors motivation, attention, concentration, and emotion. Therefore, they center their game design around the idea that playing itself needs to be the principal activity. It was pointed out that games for learning need to be good games in order to be successful. In this context, Amory et al. [8] investigated which game elements are most important to players. Their findings show that players value game elements like graphics, sound, and narration highest. These findings suggest that adventure games are a good fit for Serious Games, since they feature those elements.

Kiili [84] proposed a gaming model based on the *flow*[1] theory [31] as a link between educational theory and game design. The model defines three relevant components which should be taken into account when designing educational games: *person*, *task*, and *artifact*[2].

A more complicated model, the *game object model II*, is proposed by Amory [7] as a theoretical framework for educational game development. It defines abstract and concrete interfaces for game components, whereas concrete interfaces refer to design objects and abstract interfaces refer to pedagogical and theoretical constructs.

1 The concept of flow is explained in detail in Section 3.3
2 artifact here refers to the elemental activity which is required to solve a task

It is yet to be answered how those game models can help designing good Serious Games, as there are hardly any implementations using them to the best of our knowledge today. Another open question in game-based learning frameworks is the fact that most models do not address the learning behavior in game design [159].

3.1.2 *Collaborative Gaming*

Zagal et al. [183] describes a collaborative game in the following way: "In a collaborative game, all the participants work together as a team, sharing the payoffs and outcomes; if the team wins or loses, everyone wins or loses." In contrast to the collaborative learning concepts focused in the above, collaborative gaming aims at designing Serious Games where the process of collaboration itself is promoted. In a Serious Game context, this includes particularly soft skills like communication, teamwork, negotiation skills, and leadership skills. The underlying question is what design principles can be used to create a general gameplay, or puzzles, which foster the development of those skills resulting in an improved collaboration.

However, it is rather difficult to implement a collaborative gameplay, especially group collaboration [102]. With the exception of some cooperative games designed for two players and so-called co-op game modes for single-player campaigns, there are hardly any computer games designed exclusively for collaborative gameplay today. Various approaches exist, however: It has been proposed to make use of the five components essential for collaborative learning defined by Johnson & Johnson [77]. They can be transferred to gaming rules, as shown by Zea et al. [186].

Zagal et al. [183] analyzed one of the first collaborative board games, the Lord Of The Rings[3] board game. The difference to traditional board games is that in collaborative board games, the players do not play against each other but instead play together against the game. Therefore, they either all win or lose. During the last years, this type of board game has become more and more popular. Zagal et al. tried to analyze this type of game in order to work out its core design features. They came to the following lessons and pitfalls:

- Lesson 1: "To highlight problems of competitiveness, a collaborative game should introduce a tension between perceived individual utility and team utility."
- Lesson 2: "To further highlight problems of competitiveness, individual players should be allowed to make decisions and take actions without the consent of the team."
- Lesson 3: "Players must be able to trace payoffs back to their decisions."
- Lesson 4: "To encourage team members to make selfless decisions, a collaborative game should bestow different abilities or responsibilities upon the players."
- Pitfall 1: "To avoid the game degenerating into one player making the decisions for the team, collaborative games have to provide a sufficient rationale for collaboration."
- Pitfall 2: "For a game to be engaging, players need to care about the outcome and that outcome should have a satisfying result."

3 http://www.fantasyflightgames.com/edge_minisite.asp?eidm=58; accessed at 2015-02-05

- Pitfall 3: “For a collaborative game to be enjoyable multiple times, the experience needs to be different each time and the presented challenge needs to evolve.” [183]

Manninen and Korva [102] state that “...collaboration should involve goals in the form of perceived game-like challenges” and that “...easy achievement of this goal has to be prevented by a series of obstacles”. Moreover, they call passive obstacles of that kind *puzzles*. A main attribute of collaborative games according to their definition, is "players to cooperate to achieve a common goal against an obstructing force or natural situation..."

Those lessons and pitfalls as well as Manninen and Korva’s statements about collaboration can be affiliated to *positive interdependence*. Positive interdependence is the key design element for collaboration in games [19, 29]. Positive interdependence is implemented in some examples of first collaborative Serious Games [19, 65]. Core features are making players depend on each other in a crucial way, creating a need to share knowledge, require coordination in order to solve (time- or space-related) puzzles. Different types of positive interdependence have been defined in literature, among those are according to Collazos et al. [29]:

- Positive goal interdependence
- Positive reward interdependence
- Positive resource interdependence
- Positive role interdependence
- Positive task interdependence

Rocha et al. [132] present a set of design patterns for cooperative games based on an analysis of successful commercial games. Those are:

- Complementariety
- Synergies between abilities
- Abilities that can only be used on another player
- Shared goals
- Synergies between goals
- Special rules for players on the same team

These patterns can be applied in various challenge types like race, exploration, conflict, or economics. The race challenge creates a form of time pressure. Exploration is based on obstacles, like hindering players form easily reaching destinations. Those could be locked doors, traps, or platforms (higher areas). Conflict-based challenges are e.g., protecting, escorting, or capturing a player or valued item. Economics refers to the existence of resources and resource management.

Finally, Nasir et al. [119] designed a cooperative multiplayer Serious Game as an ice-breaker to be used before a collaborative task. Their game design aims at creating a game which incorporates intensive need of collaboration. It is based on the following principles: *Balanced individual participation*, *uniqueness of roles*, *need for social interaction*, *use of cooperative patterns*, and *concurrent play*.

3.1.3 *Evaluation of Serious Games*

Apart from designing Serious Games, some research has recently been made to investigate how to evaluate Serious Games. To the best of our knowledge, there is no unified list of criteria for evaluating Serious Games. One criterion certainly is *user experience*. Other criteria are defined by Mayer et al. [105]. They present a first list of requirements for a framework for Serious Game evaluation along with a first conceptual framework for Serious Game evaluation. Ijsselsteijn et al. [73] propose flow[4] and immersion[5] as two pivotal elements for evaluating gameplay. Nacke [117] proposed a formal theoretical framework to conduct user experience in games using both physiological user responses and psychometric questionnaires for assessment of players' subjective emotion and cognition during gameplay. Nacke et al. [118] further propose a framework specifically for measuring *gameplay experience* in Serious Games. Gameplay experience here consists of game system experience, individual player experience, and player context experience.

3.2 COLLABORATIVE LEARNING

In this section, the concept of collaborative learning will be discussed, starting with a definition of collaboration, followed by a definition of collaborative learning and parameters which are agreed to be beneficial for its success in literature. After that, the concept of *computer-supported collaborative learning* will be elaborated, followed by the impact of communication in collaborative learning scenarios. Finally, the role of the instructor in collaborative learning scenarios is put into relation.

Roschelle and Teasley [134] define collaboration as "a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem". Compared to Dillenbourg's definition of cooperation [34], "In cooperation, partners split the work, solve sub-tasks individually and then assemble the partial results into the final output", this is much more than just cooperation. Usually, this manifests in form of a synergy effect where the result of collaboration is more than the sum of the individual actions. Dillenbourg defines collaboration as follows: "In collaboration, partners do the work 'together'." [34], whereas the 'together' refers to the synergy effect mentioned before. Thomson et al. state the following about collaboration:

> "Collaboration is a multidimensional, variable construct composed of five key dimensions, two of which are structural in nature (governance and administration), two of which are social capital dimensions (mutuality and norms), and one of which involves agency (organizational autonomy)." (Thomson et al. [161])

They define a structural model of collaboration based on those five dimensions and use it for an initial approach for finding parameters for measuring collaboration.

The various definitions of collaboration are used as a foundation for a formalization of the concept of *collaborative learning*, which is used widely in e-learning and game-based learning. Dillenbourg defined collaborative learning as "a situation in which two or more people learn or attempt to learn something together" [34]. He further states that it is necessary to trigger specific learning mechanisms to learn, including both individual activities, but more important the interaction between the learning partners to trigger interaction activities like explanation, or disagreement

---

4 The concept of flow is explained in detail in Section 3.3

5 in a gaming context, the term immersion describes the perception of being part of the game

which again trigger various cognitive mechanisms. Yet, there is no guarantee that those interactions occur. Therefore, he also specifies four categories of ways to increase the probability of those interactions to occur:

- To set up initial conditions (e.g., group size and composition)
- To over-specify the 'collaboration' contract with a scenario based on roles (e.g., reciprocal teaching)
- To scaffold productive interactions by encompassing interaction rules in the medium (e.g., semi-structured interfaces)
- To monitor and regulate the interactions (e.g., teacher as facilitator, providing hints, redirecting group work)

There are various parameters defining the success of collaborative learning. One core element is the group of learners. The group can be characterized by its size, and by its composition. Regarding size, we differentiate between small groups (2-8 learners), class-sized groups (9-40 learners), and large groups (40+). In this work, the focus is on small learner groups. When forming learning groups for knowledge exchange among learners a variety of criteria including personality traits and level of proficiency need to be taken into account. Moreover, these criteria need to be matched in some cases homogeneously, in others heterogeneously, depending on learning targets and scenario [89]. In addition to that, Johnson and Johnson [77] define five essential elements of cooperation which are a prerequisite for cooperation to take place in cooperative[6] learning scenarios. Those are:

- *Positive interdependence*: knowing to be linked with other group members in a way so that one cannot succeed alone
- *Individual accountability* and *Personal responsibility*: individual assessment of each group member's performance and giving back the results to both the group and the individual
- *Promotive interaction*: Promoting each other's success by e.g., helping, encouraging and praising
- *Appropriate use of social skill*: Interpersonal and small group skills are vital for the success of a cooperative effort
- *Group processing*: Group members discussing their progress and working relationships together

Positive interdependence results from mutual goals. In this context, interdependence includes resource, role, and task interdependence. There is various evidence about the effects of positive interdependence in collaborative learning scenarios as summarized in [78], e.g., when players depend on other players due to their role (i.e., a player needs another player's help because only that player has a certain resource). "*Individual accountability* exists when the performance of each individual member is assessed and the results are given back to all group members to compare against a standard of performance" [77]. *Promotive interaction* occurs when group members encourage each other, help, or facilitate each other's efforts towards the group goal. *Appropriate use of social skills* means that group members need to possess and be able to use various soft skills like communication, supporting each other, or being able to resolve conflicts. *Group processing* is the act of reflecting on the group members' actions as individuals and as a group in order to evaluate their effort [34].

6 'cooperative' is used as a synonym for 'collaborative' in this context

It appears important to mention that up to today there is no computational model for collaborative learning [36]. It is argued that this is mainly because the parameters defining successful collaborative learning are interacting with each other in yet uninvestigated ways.

3.2.1 *Computer-Supported Collaborative Learning*

> "Communication and cooperation with other learners within a team can provide the extend of interactivity, personalization, and feedback. In addition, the requirements for team capabilities and self-organized learning suggests a need to introduce cooperative learning methods. [...] Modern computers with their ability to support communication can overcome the isolation of individuals, connecting them with other learners, tutors, and teachers." (Steinmetz and Nahrstedt [156], p. 189)

Computer-supported collaborative learning (CSCL) is a learning concept which is based on collaborative learning using computer or online technologies [153]. Early forms of computer-supported collaborative learning included the use of Wikis [93], Forums, or just email. "The primary form of collaboration support is for the computer [...] to provide a medium of communication" [153].

More complicated tools have been designed specifically for Computer-supported Collaborative Learning (CSCL) applications. Their fields of application are communication, coordination, cooperation in groups, and cooperative learning rooms (especially virtual learning rooms) ([61], p. 358). Whereas first virtual learning rooms were CSCL applications specifically designed for a CSCL purpose, most often integrating a chat system and a shared screen, later versions used existing virtual worlds like Second Live[7] or Massively Multiplayer Online Role-Play Game (MMORPG) worlds [44]. In this context, Steinmetz and Nahrstedt provide a comprehensive insight into digital learning (Steinmetz and Nahrstedt [156], p. 173ff).

In recent years, first CSCL Serious Games have been designed and implemented. They incorporate the CSCL principles and combine them with Serious Games mechanics resulting in first multiplayer Serious Games for collaborative learning [185]. Hämäläinen [63] describes an approach of a collaborative game for vocational learning focusing on design elements essential for collaboration and Reuter et al. [129] describe an approach for designing and authoring multiplayer adventures for collaborative learning deriving concepts for puzzle design in multiplayer games.

3.2.2 *Inter-learner Communication*

As mentioned above, collaboration does not happen automatically just because learners are placed in a collaborative learning setting. Various prerequisites must be met and mechanisms triggered for collaboration to take place [134]. One of those mechanisms which is vital for collaboration and collaborative learning, is communication.

Triebel et al. point out the importance and the impact of voice communication in Massively Multiplayer Online Games (MMOGs) [165] and provide a P2P-based solution of large-scale voice communication in MMOGs. Manninen described the concept

7 http://secondlife.com/; accessed at 2015-02-05

of interaction forms in multiplayer games [101]: He states that "the communicative aspect of current multiplayer games is enabled by a relatively limited set of interaction forms" and provides a hierarchical model of interaction forms. He further states that it is possible for players to communicate effectively in various forms in multiplayer games, given the system supports them in a memorable, but invisible manner. He suggests that "a creative combination of various communication channels would, perhaps, make it possible to enhance the overall interaction and further increase the communicative, collaborative and constructive aspects of multiplayer games".

Baker and Lund [14] presented design principles to promote reflective activities in interaction in the context of a computer-mediated communication. Their concept compares a free communication with a structured interface to restrict communication. Their results indicate that interactions produced using the structured interface are more task-focused and reflective.

Rauterberg examined the effect of communication on cooperation in games. Players which were able to communicate continuously, had a significantly increased amount of coalitions[8]. Moreover, it was shown that the ability to provide group process feedback had a positive influence on the extend of coalitions among players [127]. Innocent and Haines designed methods for nonverbal communication in digital games and virtual worlds going completely without written words and using pictogramms instead [75]. Metoyer et al. [111] presented a coding scheme for a structured analysis of communication of players playing a strategy game when explaining the game. They aim at better annotation and demonstration tools for machine learning systems. Their findings might also be used in other dynamic environments in which users have to fell decisions under specific spatial and temporal constraints.

3.2.3 *Role of the Instructor*

Although today there are still no systematic research findings on real-time orchestration in CSCL scenarios [64], various approaches exist for defining the role of the instructor in a CSCL scenario. Olivares [121] summarizes instructor roles within CSCL environments as *regulator*, *monitor*, and *guide*. The instructor's job is to regulate the interaction between the participants of the CSCL session. Thus, the instructor regulates interaction between learners, between learner(s) and instructor, and between learners and technology/learning environment.

Moreover, the instructor is responsible for monitoring the group of learners. The instructor can do this by asking questions, listening to the learners' answers, or by responding to them. This includes monitoring the amount and content of conversation (both chat or voice-based), and other actions taken by the learners.

This leads to the final task of an instructor: guiding the learners both in the cognitive aspects of learning and in the social dynamics (interaction) between learners. A vital part of this task is giving feedback and keeping the learners on track, i.e., assuring that they are focused on learning.

Hämäläinen and Oksanen [64] conducted a study about the role and influence of an instructor in a scripted 3D game for vocational learning. Their findings indicate that the the presence of an instructor who is making use of his/her competencies positively influences

8 Rauterberg uses the term 'coalition', which in this context can be seen as collaboration

1. the overall time needed to complete the CSCL scenario,
2. the time used on task solving and shared knowledge construction processes,
3. the time wasted for off task discussions,
4. students' capability to explain their own situation to lead to productive knowledge construction.

The underlying 3D game they used was a scripted game consisting of three puzzles that required knowledge and skills from different professions. Due to the scripted nature of the game, the course of the game appears rather predictable and different gaming session are likely to run in a similar way. Therefore, the instructor is assumed to be able to prepare to expected problems well. This, however, might be more difficult in more open games, requiring a more profound support of the instructor at real-time orchestration.

3.2.4 *Teamwork*

A major aspect of collaborative learning settings is how members of the group work together. This includes the existence and use of social skills among the learners' group, especially teamwork. However, it is important to point out that putting learners/players in a group does not automatically make them a team. At this point, we want to make use of a definition of the term *team* in order to define the difference of a mere group and a team working together towards a defined goal. Paris et al. [123] summarize the following characteristics defining a team:

- multiple sources of information
- task interdependencies
- coordination among members
- common and valued goals
- specialized member roles and responsibilities
- task-relevant knowledge
- intense communication
- adaptive strategies

Morgan et al. [116] define the term *team* as: "a distinguishable set of two or more individuals who interact dynamically, interdependently and adaptively to achieve specified, shared and valued objectives" In this context, the definition of *teamwork* is: "behaviors associated with cooperation, communication, and coordination among team members" opposed to *task-work*: "behaviors associated with the specific task being performed".

Having a definition of a team along with characteristics describing teams and teamwork, it is of major importance how teamwork can be assessed, measured, and interpreted. In literature, different performance measures are proposed: Bowers et al. [21] use coordination as a measure for teamwork. They created a list of coordination behaviors based on seven behavioral dimensions used to assess the frequency and quality of coordination. Those dimensions are: Communication, situational awareness, leadership, assertiveness, decision making, mission analysis, and adaptability.

Paris et al. [123] created a taxonomy of variables which have an influence on team performance providing the relevant factors with examples and applicable interventions to train/improve those factors (see Table 1).

Factor(s)	Description	Examples	Applicable Intervention(s)
Contextual Factors	Variables that pertain to the environment in which the team activity is embedded	Culture Climate Training/education systems Reward systems Information systems	Team selection Task design Training
Structural Factors	Variables impinging primarily from sources external to the team, but may include some internal to the team (e.g. team organization)	Physical environment Organizational arrangements Technological systems	Team design Training
Team Design Factors	Variables inherent to the team itself	Work design Task interdependence Team size/composition Leadership	Team selection Task design Training
Process Factors	Variables inherent to the team itself and the way in which it functions	Boundary management Task cohesion Performance norms Communication Team interactions Potency/ Team self-efficacy Team spirit	Team selection Task design Training
Contingency Factors	Variables impinging from sources internal and external to the team	Team application/mission Resource availability Procedural requirements Rules of operation, managing, or decision-making	Task design Training

Table 1: Team performance taxonomy based on Paris et al. [123].

Apart from parameters and factors which can be used to measure teamwork, several metrics have been proposed: The *cooperative performance metric* concept is described by El-nasr et al. [42]. It contains the following six metrics to measure teamwork, which all rely on observation with game sessions being recorded, observed, analyzed, and annotated.

- Laughter or excitement together
- Worked out strategies
- Helping
- Global strategies
- Waited for each other
- Got in the way of each other

Shapiro et al. [144] provide an overview over metrics for team performance for simulation-based training in the domain of healthcare. They distinguish four types of metrics: Event-based measurements, behavioral observation scales, behaviorally anchored rating scales, and self-report measures. One of their main results is the fact that "there is no criterion standard team performance metric or set of metrics [...] across the healthcare disciplines".

3.3 ADAPTATION

Many of a Game Master's or instructor's tasks can be defined as adapting the game according to different factors. Those factors might be difficulty, narration, players'/learners' preferences, or the need to compensate a deficit. A common definition of adaptation is "ability to make appropriate responses to changed or changing circumstances" [79].

Various common explanations of the term adaptation explain it as an adjustment to a changed circumstance or environment. In the field of software engineering, often the term context adaptation is used which describes systems adapting their structure, functionality, or behavior at run-time according to existing environmental circumstances. Other definitions explain adaptation as an intelligent reaction of a system, or an automatic control mechanism based on feedback. Steinmetz and Nahrstedt define adaptive learning systems as "learning programs capable of adapting themselves to the individual abilities of the learner, e.g., previous knowledge, interests, weaknesses or preferences with regard to forms of representation" [156].

Adaptation is a major field of research in gaming, especially in Serious Games. It is desirable to adapt games in various dimensions, like those shown above. As stated by Charles et al [25], "learning and adaptation are viewed by some as a having a crucial part to play in next-generation games." Different adaptation principles, techniques, and methods are existing in Serious Games (see Kickmeier-Rust and Albert [82]):

- Procedural and adaptive level and content generation
- Adaptive behavior of agents
- Adaptive and interactive storytelling
- Guidance, hinting
- Motivational interventions
- Adaptive presentation
- Adaptive curriculum sequencing
- Navigation support
- Intelligent solution analysis

Whereas some of them apply directly to adapting a game's difficulty, others do indirectly, and others again rather adapt visual properties or gameplay relevant elements. Zimmermann provides a comprehensive overview over the different kinds of adaptation in the field of e-learning [188]. She differentiates between structured, rule-based adaptation processes and unstructured, experience-based adpatation processes. The former refers to e.g., language or presentation formats, whereas the latter refers to a changing learning goal, difficulty, or learning strategy. It is argued that providing an automatic support for unstructured adaptation process is considered difficult as it is based on experience.

3.3.1 *Flow*

In terms of difficulty, the concept of *flow* should be considered. Csikzentmihalyi [31] introduced the term flow for a state in which a person is totally immersed into an action, in a way such that he/she even forgets about time. He states that a person in flow experiences the following characteristics:

- Clear goals and immediate feedback
- Equilibrium between the level of challenge and personal skill
- Merging of action and awareness
- Focused concentration
- Sense of potential control
- Loss of self-consciousness
- Time distortion
- Autotelic or self-rewarding experience

Figure 6 shows the *flow channel*, in which a person's skill of a task is drawn over the challenge. If the challenge is too high for the person, the perceived feeling might result in anxiety. On the opposite, if the challenge is too low for the person's skill, this might be perceived as boredom. The *flow channel* is the small channel where challenge matches the given skill. Given that a person's skill improves during exercising a task over a longer time, or repeating that task, it is assumed that the respective skill will improve over time. In order for the person to stay in the flow channel, the difficulty of the taks needs to be adapted accordingly.

Sweetser and Wyeth [157] defined the term *game flow*, transferring the concept of *flow* to games with the goal of designing and evaluating enjoyment in games. Their model includes the eight dimensions concentration, challenge, skill, control, clear goal, feedback, immersion, and social interaction.

Chen [26] concludes three fundamental conditions for *flow* in games to happen:

- The game needs to be intrinsically rewarding
- The game offers the right amount of challenges to match with the players ability
- The game provides a sense of personal control over the game for the player

Abrantes and Gouveia [5] developed a survey to test for *flow experience* in game. Their survey (questionnaire) uses the five dimensions *control*, *attention focus*, *curiosity*, *intrinsic interest* (see [164]), and *sense of time* (see [107]), based in the characteristics described by Csikzentmihalyi. Challenge is a major component of every game. Players want to test and master skills relevant for the game [95, 48]. A major element of

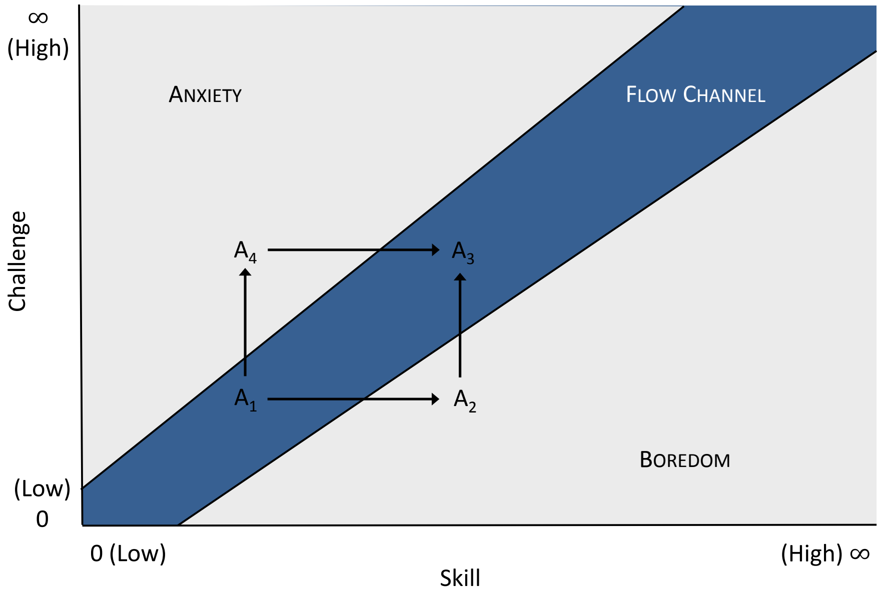

Figure 6: Flow channel after Csikszentmihalyi recreated from "The Art of Game Design" book by Jesse Schell [143].

many games is to overcome challenging opponents [171] in order to reach a desired goal [46]. Gee [51] states that it is necessary that all players, no matter what level their skill is on, perceive the game as challenging, but feasible.

3.3.2 *Player Modeling*

A core part of adaptation in Serious Games is player modeling. Only when having an accurate model of the player(s), it is possible to adapt the game to their needs.

> "In order to have an impact, Serious Games must be more concerned than traditional games with creating an accurate model of the player. This is in order to better tailor the game experience to the player's needs and preferences, including potential Non-Player Characters (NPCs) to accurately communicate with and hopefully persuade the player."
>
> (Encarnação [43])

Smith et al. [147, 148] developed a taxonomy of player modeling. They define a player model using the four dimensions *scope of application*, *purpose of use*, *domain of modeled details*, and *source of model's derivation or motivation*. Regarding scope, they differentiate based on *applicability* (one player (individual), a class of players, all players (universal), and hypothetical). In terms of purpose, they differ between *generative* and *descriptive*. The domain specifies whether the model defines *game actions* or *human reactions*. The source facet has four characteristics: *Induced* (learned by algorithmic means), *interpreted* (concluded via reasoning from records), *analytic* (derived purely from the game's rules and related models), and *synthetic* (justified by reference to an internal belief or external theory). They classified 31 existing player model concepts within their taxonomy. The majority of those were classified as universal or individual (13 each). Only five player models were classified as class-based, and only two as hypothetical.The *Passage* player model was classified as both class-based and

Instance	Scope	Source	Purpose	Domain
"Speedrunner" and "Completionist"	Class	Interp.	Descr.	Act.
Bartle's player types	Class	Interp.	Descr.	Both
WoW guild archetypes (Thurau)	Class	Induced	Descr.	Act.
PaSSAGE (Thue)	Class	Synth.	Gen.	React.
Storyboards (Fullerton)	Hypo.	Synth.	Descr.	Act.
Ludocore (Smith)	Hypo.	Analytic	Gen.	Act.
Houlette	Indiv.	Induced	Descr.	Act.
Playtracer (Andersen)	Indic.	Induced	Descr.	Act.
PaSSAGE (Thue)	Indiv.	Induced	Descr.	Act.
Race track generation (Togelius)	Indiv.	Induced	Gen.	Act.
NonyBots	Indiv.	Interp.	Gen.	Act.
Drivatars	Indiv.	Induced	Gen.	React.
Polymorph (Jenning-Teats)	Indiv.	Induced	Gen.	React.
Interactive fiction walkthroughs (Reed)	Indiv.	Synth.	Both	Act.
QuakeBot (Laird)	Indiv.	Synth.	Gen.	Act.
IBM's Deep Blue and Watson	Indiv.	Synth.	Gen.	Act.
Mario bots (Togelius)	Indiv.	Analytic	Gen.	React.
PaSSAGE (Thue)	Indiv.	Synth.	Gen.	React.
Heatmaps for Halo 3	Uni.	Induced	Descr.	Act.
Preference modeling (Yannakakis)	Uni.	Induced	Descr.	React.
Polymorph (Jenning-Teats)	Uni.	Induced	Gen.	React.
Engames tablebases (Bellman)	Uni.	Analytic	Gen.	Act.
EMPath (Sullivan)	Uni.	Analytic	Gen.	Act.
IMPLANT (Tan)	Uni.	Analytic	Gen.	Act
Ludocore (Smith)	Uni.	Analytic	gen.	Act.
Market bots	Uni.	Synth.	Gen.	Act.
Launchpad (Smith)	Uni.	Synth.	Gen.	Act.
EMPath (Sullivan)	Uni.	Synth.	Gen.	React.
Race track generation (Togelius)	Uni.	Synth.	Gen.	React.
Flow inspired (Czikszentmihalyi)	Uni.	Synth.	Gen.	React.
Mario bots (Togelius)	Uni.	Analytic.	Gen.	React.

Table 2: Overview over player models according to [148].

individual (theoretical and empirical). Thus, it appears three times in the taxonomy. Table 2 summarizes the taxonomy by Smith et al.

A different approach is described by Laws [94], who classifies role-players using the following classes:

- The *power gamer* uses the rule system to maximize his/her power; usually tries to use weaknesses in the system for his/her advantage
- The *butt-kicker* favors simple gameplay and action
- The *tactician* wants rules to be realistic, consistent, and logic; favors tactical decisions over role-play
- The *specialist* focuses on a special character type and challenges according to the related class
- The *method actor* values role-playing his/her character above everything
- The *casual gamer* rather plays to be part of a social group than for the game itself

This can be considered a rather descriptive player model.

Another very well known player model is Bartle's player model [15]. Bartle categorizes players along two axes: *acting* ↔ *interacting* and *players* ↔ *world* (see Figure 7). Players who favor to act on other players, are called *killers* as this type of players likes to affect other players which in most role-play games is fight-based. Players who like acting on the game world are called *achievers*. They are interested in acting in the world. Players preferring to interact with other players are called *socializers*. Interaction is most often focused but not limited to communication. Players who like interacting with the world are called *explorers*. Their main objective is to explore and experience the game world. Bartle's player model is assigning a value in the

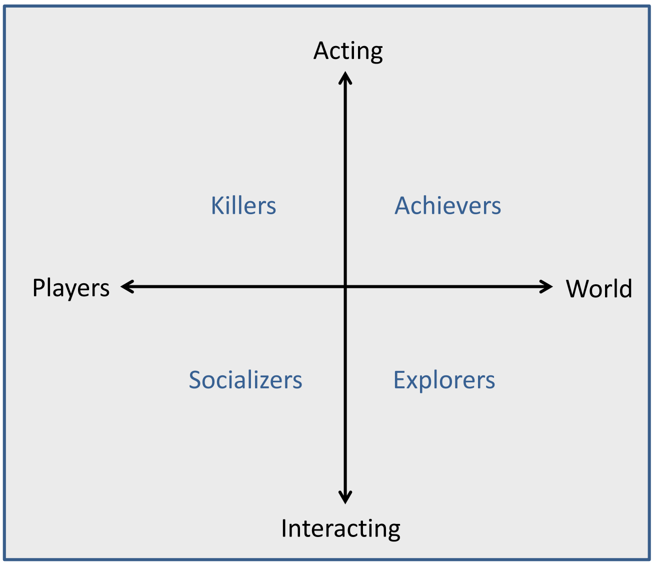

Figure 7: Player model after Bartle [15] showing the two axes Players ⟷ World and Acting ⟷ Interacting. The four player types are located between the axes.

range [0,1] for each of these types to a player indicating to which extend the player is an achiever, explorer, etc. This player model is also descriptive. According to Smith et al. [147] it's scope is class-based, the source is empirically interpreted, and the domain defines both game actions and human reactions.

A more generic model is proposed bei Houlette [72]. The model is based on a set of player traits which can be freely defined according to the game domain. Each trait is assigned a value in the range [0,1]. Again, this is a descriptive player model. This player model's scope is on individual players, the source is empirically induced, and it defines game actions.

The process of player modeling in adaptive games consists of several steps: First, an initial model of the player is established using player preferences. This model is used throughout the game to adapt it. However, the model should be updated during the game according to the player's actions and behavior. This cycle of re-evaluation of the player model and adaptation of the game is shown in Figure 8.

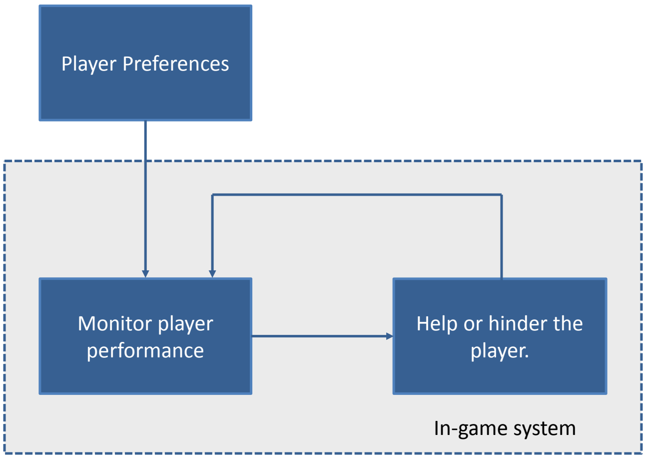

Figure 8: Basic adaptive game system after Charles and Black [24].

Charles and Black [24, 25], propose a player model based on neural networks. Their system also uses a feedback loop for measuring if an adaptation had a positive influence (according to its intention). They infer that if an adaptation was bad, either the wrong adaptation was chosen or the player was modeled wrong. They use this data to re-model player tapes and re-evaluate the adaptation algorithm.

It is important to show that player types (within a player model) are not necessarily independent or contradictory from each other. Yee [182] proposed an empirical model of player motivations based on Bartle's player model. Results of an empirical study with 3000 participants showed that play motivations in MMORPGs do not suppress each others as suggested by Bartle. A player who scores high on the achievement score does not automatically score low on the social component.

3.3.3 *Interaction*

In multiplayer games, it is not only important to model players' behavior, but also their interaction. It is hard to predict and subsequently to model interaction between players because this depends on a multitude of variables. Those are like before variables of the respective player and the game world, but also the state and actions of other players. To the best of our knowledge, there is only little research covering this topic. Manninen [100] proposed a hierarchical interaction model for multiplayer games. The levels differentiate between a cognitive level, sub-goals, purpose, combined signals, and motor skills. Interaction forms in multiplayer games are mapped to those.

An approach of player modeling and adaptation in the field of learning games is proposed by Padilla-Zea et al. [187]. They use an agent-based approach to monitor player behavior and interactions. A facilitator agent then decides about whether an adaptation should be applied or an instructor should be informed.

Konert proposed interaction patterns to support peer interaction between players, to allow peers to influence gameplay [87]. His findings showed that players experiencing game adaptation by their peers show significantly stronger acceptance values compared to players playing a game without adaptation. However, in the described scenario, the peers were not part of the game itself. Instead they adapted their peers' game from outside (via social media).

3.3.4 *Learner Modeling*

Learner modeling refers to capturing a learner's state of knowledge, learning style, and learning path, i.e., the order in which a learner acquires new knowledge. The last aspect is especially interesting in digital learning environments or games where learning content might be presented to the learner in a predefined order and interdependencies between the learner's state of knowledge and the game progress might exist. Learner modeling thus can be used to assess a learner's state of knowledge, learning preferences and learning style.

A well-established basic model for modeling knowledge of a specific problem is the *knowledge space theory* by Doignon and Falmagne [38]. Their model focuses on observable solution behavior and does not consider learning objectives, skills, or competencies. They state that "'knowledge' of an individual in a particular domain of knowledge can be operationalized as the solving behavior of that individual on a domain-specific set X of problems." A learner's knowledge state is defined as the subset of problems he/she is able to solve.

This model has been extended by Korossy [92] resulting in the *Competency-based Knowledge Space Theory (CbKST)*. The goal of this extension is to be able to link observable behavioral aspects with the non-observable construct of skills or knowledge related to the behavior. The define *performance* as "... the observable solution behavior of a person on a set of domain-specific problems." Further, they define *competence* as "... a theoretical construct accounting for the performance." Korossy defines the term *knowledge structure* as the pair (X, K) where X is a set of problems and K is a family of subsets of X, the empirically expectable solution patterns. The elements of (X, K) are called *knowledge states*. Korossy further defines a knowledge space as a knowledge structure (X,K) with $\emptyset, X \in K$ and 'K is stable under union'.

Heller et al. [68] propose an extension of the *knowledge space theory* to link learning objects and assessment problems with relevant skills. The *Extended Knowledge Space Theory* includes a set of assessment problems, a set of learning objectives, and a set of skills relevant for solving problems, and taught by the learning objects. They define the knowledge structure K over a domain Q as the collection of possible knowledge states of Q, with $\emptyset, Q \in K$. They model the knowledge domain using Hasse diagrams. Figure 9 shows a Hasse diagram of a knowledge domain with five elements and dependencies between them. A dependency between element x and y is considered to be an "x-requires-y"-relation.

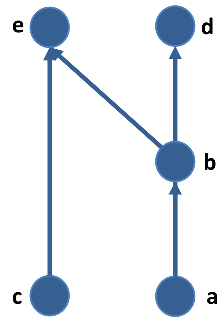

Figure 9: Example of a knowledge domain Q = a,b,c,d,e and dependencies in form of a Hasse diagram (from [68]).

From the relationship between the elements, the knowledge structure can be deduced which shows possible learning paths. The knowledge structure contains the set of all possible knowledge states, considering which knowledge needs to be acquired as a prerequisite for another piece of knowledge. Figure 10 shows the knowledge structure related to the knowledge domain relationship of Figure 9.

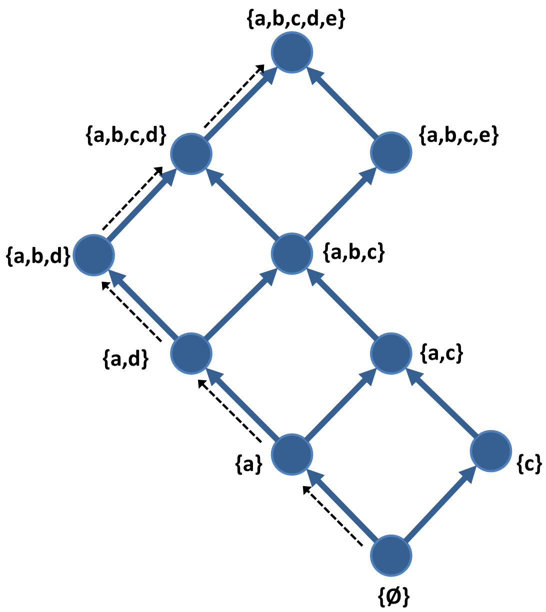

Figure 10: Example of the knowledge space for the skill graph shown in Figure 9 modeled as a Hasse diagram (from [68]).

The *outer fringe* of a knowledge state is the set of problems which can be tackled next from starting from a given knowledge state. It is defined via the successor states of that knowledge state. Using the example from Figure 10, the two successor states of $\{a, b\}$ are $\{a, b, c\}$ and $\{a, b, d\}$, enhancing the current knowledge state about either $\{c\}$ or $\{d\}$. Therefore, the outer fringe of the knowledge state $\{a, b\}$ is the set $\{c, d\}$.

The *inner fringe* indicates what a learner knows or which problems he/she can solve at the moment. It can also be interpreted as what a learner learned most recently. For the knowledge state $\{a, b\}$, this is the set $\{b\}$. For the knowledge state $\{a, b, c\}$, it is $\{b, c\}$.

The *Extended Knowledge Space Theory* (see [68]) extends the model about the concept of learning objects and skills. The set of skills S shows which skills are relevant for solving the problems of set Q. They are taught by the learning objects L. This is meant to be a more fine-grained description of a learner's capabilities. A mapping r associates a subset of skills, the required skills, to each learning objective. A mapping t associates a subset of skills, the taught skills, to each learning objective. Thus, it is possible to define which skills need to be acquired for a learning objective to be taught and what a learning objective does teach.

Similar to the knowledge structure, the competence structure can be modeled via a Hasse diagram [83] (see Figure 11). Thus, competencies are ordered in a semi-order using a directed graph which is reflexive, anti-symmetric, and transitive.

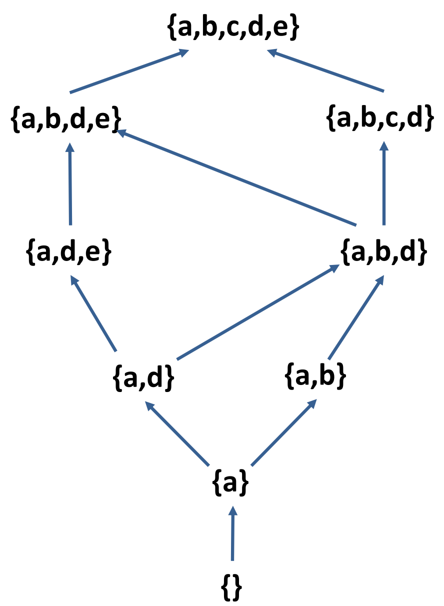

Figure 11: Example of a competency graph modeled as a Hasse diagram (from [83]).

Steiner et al. [154] use the CbKST in their learner (user) model which is composed of a skill-based plan, a competence goal, the competence state and the knowledge state. The model combines the concepts presented above and models relations between problems, activities and skills based on competence goals, competence states, and a skill-based plan. In their model, competence goal and competence state are defined as a set of skills of a specific domain map. The competence goal defines what a learner should learn skill-wise. The competence state defines which skills the learner has learned already (see above). The skill-based plan is defined as a list of activities a learner should perform. The knowledge state is defined as a set of problems which the learner is capable of solving (see definition of knowledge state above).

Göbel et al. [52] introduced the concept of the *Narrative Game-based Learning Objects (NGLOBs)* which makes use of the CbKST but also includes a gaming and a narrative dimension. An NGLOB is a meta-structure to describe a scene in a story-based learning game. The description includes a narrative context using the model of Campbell's [22] Heroes Journey. The gaming dimension describes the appropriateness of the scene regarding the used player model (e.g., after Bartle), i.e., if the scene is rather fit for an explorer or an achiever. It describes which competencies/skills of the competence state are a required for this scene to be chosen, and which competencies/skills are assumed to be taught by playing this scene.

All learner models try to represent a learner's state of knowledge, skills, or competencies. Therefore assessment of those skills is crucial. "Assessment is the one thing in successful adaptation" [82]. Kickmeier-Rust and Albert propose a list of relevant performance related indicators (in Serious Games):

- Score
- Task completion rates
- Task completion times
- Taks success rate
- Task success depth
- Progress in the game world
- Incongruent behavior

Score can be an in-game defined assessment of player performance (e.g., highscore). The task completion denotes which percentage of a task was completed, whereas the task completion time refers to how many times a task was completed. The success rate denotes the percentage of successful, i.e., correct completion of a task, whereas the success depth differentiates between different degrees of success. The progress in the game world refers to how far players were able to get within the game. This might be matched to player performance, considered progress steps relate to obstacles/tasks which need to be overcome and for which knowledge or special skills are required. Finally, it might be helpful to track incongruent behavior as an indicator for succeeding by chance.

Augustin et al. [11] provide a theoretical model for assessment of knowledge and learning progress in the context of digital learning games. Their approach is based on a mathematical framework which describes a learner's problem-solving behavior in an explorative and problem-oriented gaming situation.

It should be noted that all concepts based on the CbKST have in common that establishing the family of knowledge states (or competence states) requires domain experts.

3.3.5 *Group Modeling*

Modeling groups of players/learners is more than the sum of modeling the single players/learners. A very important component which comes into play here is interaction. Moreover, modeling cooperation and collaboration requires a model of interaction. While researching existing solutions for modeling groups of learners/players, only few scientifically founded models could be found.

Inaba et al. [74] developed interaction patterns in collaborative learning scenarios. They developed a collaborative learning ontology and used it to model the learning concept of peer tutoring.

A theoretical model of student interaction in the context of cooperative learning is proposed by Webb [173]. It models interaction in terms of expression for help, and giving help.

Zea et al. [184] propose the integration of various existing models. They differentiate between task and goal models, state of the gamer model, player model, and group model. They differ between formal groups and temporal groups, whereas the former are grouped over a longer period of time, and the latter disappears after the respective activity is finished. Further, they focus on communication in the group and define four elements to describe the group model. Interactions are studied using information from a set of networks describing the interactions between the players. Those networks are established using social media analysis.

Hoppe and Ploetzner [71] describe a model-based support for group learning. Their group model us constructed from individually assessed models of single learners. Their modeling primitives are: *knows*, *has difficulties*, and *can help*. They state that it is "indeed very hard to predict whether collaboration and the mutual complementing of knowledge and skills really takes place".

Konert et al. [91] examined the assessment of player and learner models of players based on personality traits. Their findings indicate that separate modeling for the adaptation *game flow* (playing) and *learn flow* (learning) is necessary.

3.3.6 *Adaptation Algorithms*

Next, an overview over existing adaptation concepts and algorithms in the contexts of learning and games is given, some of them in combination with player and learner models.

Carron and Marty [23] propose an adaptation mechanism for a learning game based on a user model. Their adaptation algorithm responds to recognized deficits or unwanted player behavior and updates the user model accordingly. A desired user model can be defined as e.g., an improvement of a skill, etc. The adaptation algorithm then can chose adaptations which might help to achieve the desired model.

Bellotti et al. [16] propose an adaptation engine which selects tasks based on a player model. According to the definitions of player and learner model in this work, their player model can also be considered a learner model as it models learning behavior, knowledge, and skill, too. Tasks are defined through parameters describing their entertainment value, skill relevance, covered learning styles, difficulty, difficulty adaptation range, and others. The adaptation algorithm then calculates costs for sequences of tasks and chooses the optimal one.

Mehm proposed an approach to enable an author of a Serious Game via an authoring tool to define adaptativity within a scene-based game [108]. Hence, the author would be able to define adaptive story paths on a macro-adaptive level or micro-adaptivity within scenes, i.e. adapting scenes towards the characteristics of a player. The developed authoring tool *Storytec* allows the author to specify which scenes or scene shapes are fit best for which type of player and how a model of the player can be derived from his/her decisions within the game.

Further, there are various approaches based on agents. Westra et al. [179] use learning agents coordinated by an organizational framework that specifies the limitations of the adaptation in each context. They make use of a user model, the agent's preferences, and the organizational model to optimize the learning curve.

The approach by Vassileva & Bontchev [170] uses a 3-dimensional model consisting of a learner model, a domain model, and an adaptation model. Predicate logic was used to define adaptation rules. Rules are composed of starting rules, pass-through graph rules, and rules updating the learner model. The adaptation engine calculates an optimal course for a learner through the learning environment based on the present model.

Spronck et al. [149] describe an approach to use dynamic scripting to adapt opponents' strategies in role-play games. The goal is to provide players with opponents matching their skill level which improves while playing a game over a longer time span. Thus, players should be held in the flow channel.

In the context of the <*e-Adventure*> Serious Games authoring tool, Torrente et al. [163] define an adaptive learning pattern which enables an adaptation within a game able to fit different learning styles by displaying different game behaviors. Their model operates on two layers: Choice of an individual game path to diversify the learning experience, and choice of game content for a more fine-grained adaptation.

A similar approach is used by Göbel et al. [52]. Their adaptation mechanism uses the information about the player in the three dimensions learning, gaming, and narration in order to adapt the game on two levels. The macro-adaptation chooses an individual path through the game, selecting scenes according to their usefulness in terms of the story and their associated learning objectives. The micro adaptation adapts the selected scene in terms of player model, knowledge state, and learning preferences.

Yannakakis and Maragoudakis [181] define an algorithm based on criteria which make a game interesting. They define a metric using difficulty, diversity in opponents' behavior, and a preference to aggressive behavior of opponents. They show that the use of their algorithm can generate more interesting game instances for players using the game *Pacman*.

3.4 AI/PLAYER SIMULATION

Typically, in a computer game it can be differentiated between characters controlled by the player(s), and characters which are not controlled by players, so-called NPCs. Those NPCs can either be simple adversaries, like the ghosts in *Pacman*, or realistic human (or similar) characters in role-playing games. Hence, as they are not controlled by real players, NPCs are controlled by AI. The topic of AI in computer games is probably one of the most profoundly researched topics during the last two decades with significant advantages towards both the realism in behavior of virtual characters and an 'intelligent' behavior of AI adversaries without the need of unfair advantages. However, today, AI computer enemies are still far behind human players in complex games under real-time conditions. Whereas today's (super) computers are able to beat the world's best chess players, real-time computer strategy games are far more complex due to the immense amount of variables and possible game states. On top of that, only a small part of the computation time per frame can be assigned to AI calculations. The aspect of intelligent opponents, however, is not focus of this work. Instead another aspect of AI in computer games shall be focused. It is the simulation of players with the goal of a believable/realistic behavior. When designing an AI, it needs to be differentiated between AI structure and generation of behavior, whereas the structure describes how the AI is built.

3.4.1 *Structure*

There are various methods of AI simulation of players. One method is using a Finite State Machine (FSM) [47]. An AI consists of a finite set of states and transitions between those states. Usually, an AI needs one transition and one successor state for each relevant event to be modeled for one state. For games this is too complex. Consequently, for AI in games, abstract states are used instead to encapsulate behavior.

A very prominent and intuitive concept for simulation of real people is the *agent*-based modeling. Russel and Norvig [138] define an agent as an entity which can perceive its environment via sensors and interact with the environment via actuators. An agent's behavior is defined via its agent function. Russel and Norvig differentiate between four types of agents:

1. Simple reflex agent: this agent type only reacts to the current sensation. Previous impressions or actions are not considered for choosing the next action. Based on the current perception of the environment and a set of rules mapping perceptions to actions, an action is chosen and executed.
2. Model-based reflex agent: this agent type contains an internal model of its world based on previous perceptions. Actions are chosen according to the current world state. It can be considered an extension of the reflex agent.
3. Goal-based agent: this agent type is an extension of the reflex agent. It has a predefined goal which it desires to achieve. This goal is a certain world state. With respect to this goal, it chooses its action such that the world stat is moved towards the desired state. Thus, the actions it chooses are not based on a set of rules, but on a goal to be reached.
4. Utility-based agent: Similar to the goal-based agent, the utility-based agent works towards a certain desirable world state. In addition to this, a utility function defines how well any reachable state is to be evaluated. Thus, it can be used to decide on an optimal way towards the desired goal state. Figure 12 depicts a model of a utility-based agent. As utility-based agents make use of the game state (which is the result all all previous events and actions), they can be considered to take the past into account.

3.4.2 *Behavior*

There are various approaches for generating behavior. The most simple approach is the use of a look-up-table with predefined behavior or plans [96]. Another, more complex, approach is making the AI learn an optimal behavior. Tan and Cheng [158] developed an AI for real-time strategy games using reinforcement learning to learn which strategies of a predefined set to use. Through the learning process (experience), the AI is able to decide which strategy is best fit to be used in which situation, improving its performance compared to a fixed strategy or a randomly chosen strategy.

Another concept, which is especially useful for movement, is the concept of potential fields [62], where attraction points within level/terrain are generated based on goals or targets. A goal (pickup, enemy, etc.) hence creates a potential field. The AI perceives the superimposed potential fields thus, being 'attracted' to a certain position potential from each position in the level.

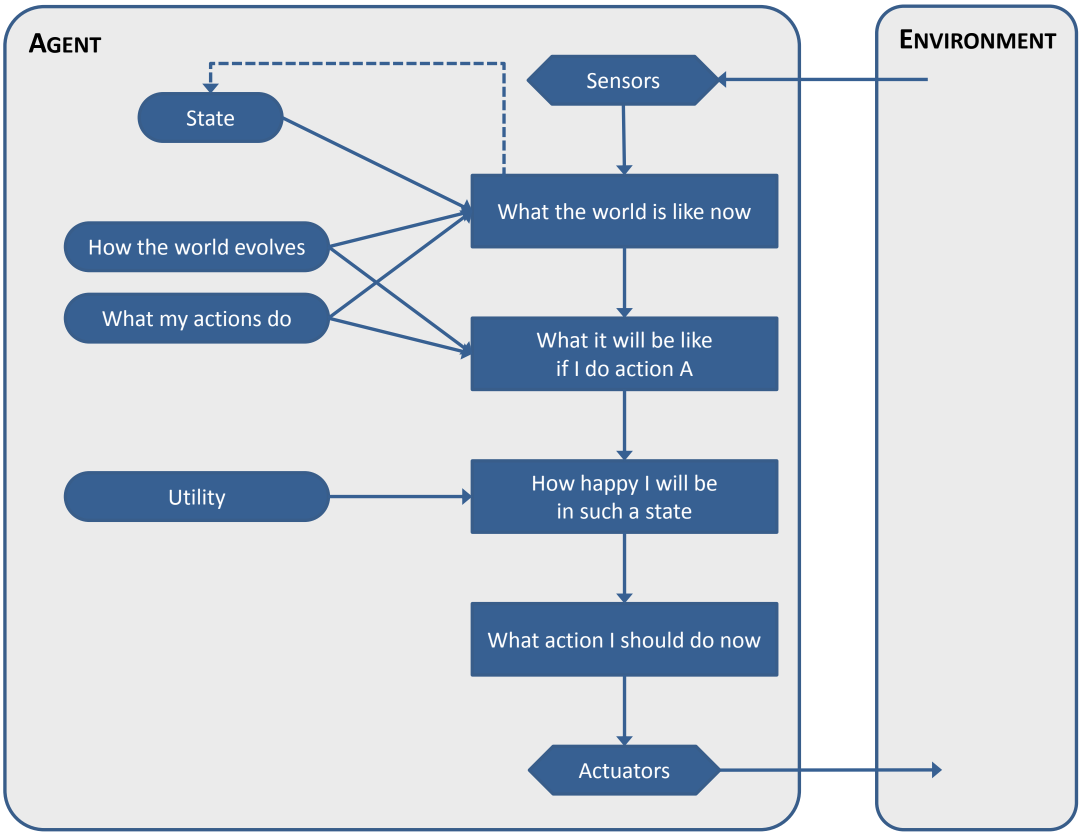

Figure 12: Utility-based agent schematic after Russel and Norvig [138].

In addition to this, planers have been proposed as a useful concept. They are able to create one or more plans for how an agent can reach a designated goal from a certain state. Russel and Norwig [138] describe different types of planers:

- A search in the state space: world states are modeled via a tree whereas the root of the tree is the current state and child nodes present possible successor states. A search algorithm then iterates through the tree until the desired state is found. The resulting plan is the path along nodes towards the goal node.
- Planning graphs: Planning graphs are a special graph-based structure. They consist of several levels, alternating between levels representing states or actions, whereas actions need to be executable from the previous state and states need to be reached by executing an action. Using the graph plan algorithm, a plan can be created from this graph in polynomial complexity.
- Situation calculus: Situation calculus generates plans using first-order logic. An appropriate representation of states and actions is mandatory. The calculus can create a plan which reaches the goal via inference[9].
- Partially ordered plans: They are generated using a search in the planar room[10].

All of those planers depend on feasible heuristics to work efficiently.

A lot of research in this field targets realistic behavior in terms of narration and storytelling. Riedl and Young propose an intend-driven planning algorithm for nar-

9 inference is the act or process of deriving logical conclusions from premises known or assumed to be true

10 i.e., the plans are represented as a planar graph

ration generation [131]. Theune et al. [160] developed the Virtual Storyteller, a framework for story creation. Their framework is based on autonomous characters. A similar approach is taken by Mateas and Stern [104], who defined a reactive planning language designed for authoring believable agents.

The approach of Ram et al. [126] focuses on an adaptive game experience. Virtual characters should behave differently according to the player's actions thus adapting the game(play). They use a case-based reasoning approach for the creation of reactive agents.

Bellotti et al. [16] simulate player behavior based on a player and learner model. They use synthesized virtual players to simulate how players might advance through a learning game when presented with various tasks. Their player simulation is used to test their adaptation engine implementation with synthetic players using different gaming preferences, learning styles, knowledge, and skills (see Section 3.3.6).

3.5 GAME MASTERING CONCEPTS

In this section, the role of the instructor in collaborative scenarios shall be picked up again. Similarities to the role of a Game Master in pen&paper role-play games will be elaborated. As a next step, first approaches to bring those concepts into digital games are proposed, focusing especially on the problem of the *Narrative Paradox*. The second part will cover approaches of using those concepts for instructor support in game-based collaborative learning scenarios.

Pen&paper role-play games are story-based games played purely in the players' imagination. A so-called Game Master (GM) tells a story of which the players, often taking the roles of heroes, are part. The most important difference to a pure story is the fact that the players influence and alter the story through their actions.

Lindley [97] introduced a classification plane (Ludic Space) as a triangle (see Figure 13) with the three dimensions *game*, *narrative*, and *simulation*. *Simulation* is often referred to as interaction in other similar classifications. As we can see in Figure 13, a pure game like *Tetris* can be classified in the *game* corner, as it does not incorporate any significant form of narration or simulation. Its main feature is the gameplay. Movies are a pure form of *narrative* as they do not have any game or simulation elements due to their lack of interaction. Simulations are the pure form of *simulation*. They do not need a narration or need to be playful. Instead they simulate a real world aspect in a simplifying but realistic manner. In the center of this triangle are role-play games. They incorporate game aspects, but also a major narration aspect. Moreover, they incorporate a lot of interaction, both player-to-player and player-to-game world.

One of the most important features of a good Game Master is the ability to be able to adapt the story according to the players' actions. While doing this, the GM has to pursue two main objectives:

1. Keeping track of the original story (goal)
2. Avoiding to make players feel to be only a marionette

These obviously opposing objectives are also referred to as the *narrative paradox* (see [12]). Approaches exist to address this problem using techniques from role-playing games (see [13]). Translating these objectives towards a collaborative learning scenario, the GM would keep track of his/her goal (the learning goal) while provid-

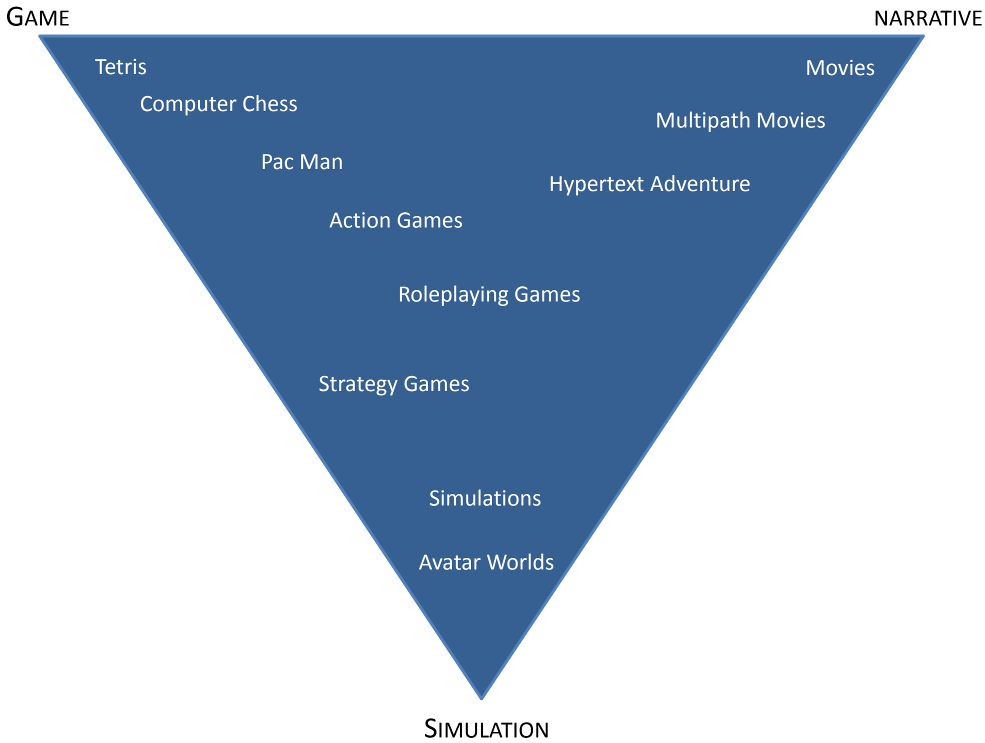

Figure 13: Ludic Space after Lindley [97]. Examples of narrative, simulation and game applications are arranged in the triangle.

ing the learners with as much freedom as possible so that they can learn in the constructivist learning style that is intended in collaborative learning.

There have been some approaches to adapt the concept of Game Mastering for digital games. Tychsen et al. [168], and Aylett et al. [13] analyzed the role of a GM in role-play games. Tychsen moreover provided an overview of Game Master functions in RPGs. Peinado & Gervas [124] suggest to use the principles of Game Mastering to solve the *narrative paradox* in digital games. Tychsen [167] points out the importance of a difference in actual authorial control and perceived authorial control among players. It is important for players to have the feeling that their actions do contribute to the game development opposing to believing to be a mere marionette. Tychsen [166] also points out one major weakness of state-of-the-art computer role-play games. Their narration is limited to predefined story paths. Moreover, NPCs often have a limited functionality and low credibility in terms of realism. He argues that a GM can help to overcome those issues.

In order to use Game Master concepts for orchestration for collaborative learning scenarios, it is necessary to investigate the role of teachers/trainers, or generally instructors. Dillenbourg and Jermann [35] work out teacher's tasks in learning scenarios, like leadership, flexibility, control, integration, linearity, continuity, drama, relevance, and awareness. Ketamo and Kiili [81] investigate the role of teachers in game-based learning scenarios. They point out one major problem which leads to a reluctance among teachers to use games in class: Being afraid of losing control of the learning process. We also would like to refer to Section 3.2.3 for the role of the instructor in this context.

3.6 identified gap

Despite the field of Serious Games existing for many years and research in different areas related to Serious Games being conducted, there are still many open research questions related to this field. Although there has been initial research about Game Mastering and instructor support in Serious Games, there is still no unified concept for orchestrating Serious Games or for supporting instructors when they do so. There is some insight into how to transfer Game Mastering concepts from pen-and-paper role-play games. But it is still unclear how to incorporate the concept of Game Mastering into Serious Games, especially in a generic way. The role of the instructor in collaborative learning scenarios has been explored and described pretty well. However, it is still unclear how to incorporate the instructor into a (collaborative multiplayer) Serious Game and enable him/her to perform his/her duties in the game environment. Hence, the identified gap in this field of research is the lack of concepts for the meaningful integration of an instructor into a (collaborative) Serious Game. Consequently, it appears necessary to develop concepts to provide instructors with the necessary means to assess and adapt game-based learning processes, possibly using Game Mastering concepts.

In terms of AI in games, there has been research for more than 30 years, but it has mainly focused on 'intelligent' adversaries. During the last two decades, focus has shifted more toward realistic characters and consequently on realistic behavior. Yet most work in this area is focused on virtual characters that serve a certain purpose within the game, possibly within the scope of narration. There are still few approaches to simulating player behavior in multiplayer games with respect to player behavior, learning, and interaction between players. The identified gap here refers to a need to simulate gaming behavior, learning behavior and progress, and interaction between players in the form of virtual players for testing purposes based on appropriate player, learner, and interaction models.

Regarding adaptation in games, several approaches exist. The most prominent and well-known adaptation mechanisms are used for difficulty adaptation in single-player games where skill, number, or other properties of AI enemies are adapted according to the player's performance. There are, however, few approaches for adaptation in multiplayer games. Also, for adaptation in terms of learning - i.e., the selection of suitable learning content - there are initial approaches, but only for single-player Serious Games or scene-based (e.g., adventure) games. To the best of our knowledge, there are no concepts for adaptating multiplayer games to the needs of a group of heterogeneous players. In order to be able to react to problems and deficiencies in multiplayer learning games, adaptation mechanisms that consider not only one player but a group of players/learners need to be developed.

The resulting gap is thus the adaptation of multiplayer Serious Games in collaborative multiplayer scenarios, where adaptation refers to either automatic adaptation or human adaptation by an instructor. In the second case, the identified gap concerns the question of how an instructor can be enabled to adapt a game in a meaningful way.

3.7 CHAPTER SUMMARY

In this chapter the related fields of research and relevant work in those fields in relation to this thesis were analyzed. In the following the focus is on technical aspects motivated by interdisciplinary research questions and challenges originated in the learning/teaching domain.

The concept of collaborated learning is discussed in Section 3.2. Different definitions for collaboration are compared and mechanisms for collaborative learning introduced as a first step. Prerequisites for collaboration and cooperation in learning scenarios are examined. CSCL environments and game-based CSCL environments are described. In the context of collaborative learning using computer technology, the special opportunities and challenges of computer-supported collaborative learning are elucidated, and the importance of communication in collaborative learning scenarios is further discussed with a focus on communication on virtual worlds and game-based approaches. As the instructor was identified as a major aspect of collaborative learning, the role of the instructor is examined, and typical instructor tasks and responsibilities and studies about instructor influence on game-based learning scenarios are explained. Additionally, the role of teamwork in collaborative learning and assessment of teamwork is discussed. This includes a characterization of teamwork and teamwork elements and factors for successful teamwork.

The second major field of related work, discussed in Section 3.1, is the field of Serious Games design. The various design challenges of designing collaborative multiplayer Serious Games are investigated including the integration of learning content into the design process of Serious Games, especially seamless integration. Moreover, the role of the target audience is considered, and important factors with a major impact on game design decisions are examined. Based on that, important design guidelines, lessons, and pitfalls for designing collaborative multiplayer games are discussed and compared. Lastly, criteria for evaluation of Serious Games are summarized from the literature with a focus on user experience and methodologies to measure user experience and gameplay experience.

Simulation of players and player behavior in games is discussed in detail in Section 3.4, which looks into different structures of AI-based simulation concepts as well as the generation of meaningful behavior. From a structural perspective, the concepts of simple reflex agents, model-based reflex agents, goal-based agents, and utility-based agents are examined. From a behavioral perspective, different approaches to generating realistic and believable behavior are analyzed.

The concept of Game Mastering is elaborated in Section 3.5, which introduces and explains the term *Game Master* and how it derives from role-play games. In this context, the importance of the *narrative paradox* is explained and how GMs try to solve it in role-play games. Preliminary approaches transferring the concept into digital games are explored.

In Section 3.3, the term *adaptation* is defined and important aspects of adaptation in games are elaborated. The influence of flow on games is explained using the model of Csikzentmihalyi and extensions of the model with a closer focus on games. Different player model concepts are discussed and compared, with the outcome that different player models are suited for different game genres and types with different scales of granularity. Some player models are more suitable for trying to model players individually, whereas others are better for modeling classes of players. Learner models are introduced, and different types of learner models are discussed and explained with a focus on the Competency-based Knowledge Space Theory, which was identified as the most relevant learner model in the context of this thesis. Subsequently, existing adaptation algorithms in gaming and learning context are discussed. These are either player model-based, agent-based, use dynamic scripting, or define metrics based on difficulty, opponent diversity, or opponent behavior.

Finally, the identified gap is derived from the related work presented above. The resulting gap is the adaptation of multiplayer Serious Games in collaborative multiplayer scenarios, where adaptation refers to either automatic adaptation or adaptation by a human instructor. Thus, the identified gap contains the question of how an instructor can be provided with meaningful information about a game session and how he/she can be enabled to adapt a game in a meaningful way. Moreover, it contains the question of how such a game session can be adapted automatically, including an underlying game model, adaptation algorithms and metrics (Section 3.6).

4 CONCEPT AND APPROACH TOWARDS A GAME MASTERING AND ADAPTATION MODEL

»If a child can't learn the way we teach, maybe we should teach the way they learn.«

— Ignacio Estrada

In this chapter, the core concept and general approach of this thesis will be described (Section 4.1) based on the findings in the previous chapter. This contains the approach towards adaptation in collaborative multiplayer Serious Games according to the needs of a group of learners and towards the support of the instructor during learning sessions. Therefore, the formalization of collaborative multiplayer Serious Games will be described in Section 4.2 in detail which constitutes a foundation for the developed concepts for the collaborative multiplayer game adaptation mechanism (Chapter 5) and the Game Mastering interface (Chapter 6). Moreover, the concept for the collaborative group model will be presented in Section 4.3. The group model is the approach to answer the question of how multiple, heterogeneous players can be modeled in terms of learning, gaming, interaction, and challenge. The presented model will be a formal representation of players in collaborative multiplayer Serious Games.

4.1 APPROACH

The overall approach followed in this thesis is to

1. Conceptualize a formal representation and model of Collaborative Multiplayer Serious Games (CMSGs) as a formal foundation for an algorithmic adaptation approach.
2. Develop a model for individual players in a CMSG including the dimensions learning, gaming, interaction, and challenge. Subsequently, develop a group model for a group of players.
3. Define a generic interface for interaction with the CMSG for observation and adaptation which uses the formal representation.
4. Define types of observation and adaptation to be used in CMSG by Game Masters and methods to assess them.
5. Develop an algorithm for an optimal selection of adaptations based on current game state and information provided by the group model.
6. Design and implement an agent-based simulation of players for evaluation of the adaptation mechanism using configurable player traits and knowledge.
7. Design and implement a CMSG prototype following collaborative multiplayer SG design serving as a testbed and foundation for implementation and evaluation of the concepts and algorithms developed in this thesis.
8. Evaluate the player simulation to assure that their behavior is determined by their configured traits and knowledge and can be considered as reasonable.

9. Evaluate the adaptation mechanism for effectiveness in terms of learning and challenge and for Game Mastering effectiveness and satisfaction. The evaluation is performed by both a large set of simulated players following a realistic player and learner model and by real players playing the game while instructed by a GM or GameAdapt.KOM.

4.2 COLLABORATIVE MULTIPLAYER SERIOUS GAME MODEL

This section contains the presentation and explanation of the developed model for a collaborative multiplayer Serious Game. The goal of the developed model is to have a formalized presentation of CMSGs as a foundation for a structured work with algorithms and models making use of CMSGs. The model can be interpreted as a characterization of the underlying game in terms of all relevant characteristics, i.e., in terms of gameplay, learning, game mechanics, interaction, and presentation.

4.2.1 *Game Type Limitation*

Considering the current state of the art in the field of gaming models, a generic game model including all kinds and genres of multiplayer (Serious) games, seems highly unrealistic. Therefore, with respect to the focus of this thesis, the model presented here focuses on multiplayer games for a small group (4-6) of players. It is assumed that players have an identity and a representation in the game which is controlled by the respective player. This could be an avatar in a 2D/3D (action adventure-like) game where the players control a character through a game world. In a game where players are bodiless, like in a simulation, this identity might be reduced to a distinguishable unique name or an icon. It is further assumed that players are able to be aware of each other's presence in the game and that they can interact with each other.

Additionally, the game is considered to be non-linear in its sequence and spatially restricted but offering free movement within the game world. This implicates that there cannot be a completely deterministic predefined course of the game. Whereas in a linear game an author has full authorial control of the sequence of scenes and subsequently the sequence of content and actions presented to players, in the scenario here this is not possible. The course of the game in genres regarded here, like action adventures or simulations is profoundly based on player actions. Their sequence, however, can hardly be predefined. This makes recognizing in which game situation a group of players is located at a certain point in the game a major challenge. The presentation of the game can be either 2D or 3D with various camera modes (first person, third person, top-down, etc.).

Finally, the focus is set on digital educational games, more closely on collaborative learning games. Those games are characterized through a dedicated learning goal which is pursued by playing the game and a focus on collaboration throughout the game. This is usually reflected in a gameplay which requires intensive interaction between players.

4.2.2 *Definition GM-relevant*

As a first step, the term *GM-relevant* will be defined:

Definition: GM-relevant. *A game element is considered to be GM-relevant if the element does influence the type and the flow/sequence of the game such that a manipulation of the element changes the type or the flow of the game in terms of gameplay, learning content, or interaction between players (including difficulty) in a way that is considered relevant to the GM.*

4.2.3 *Game Components*

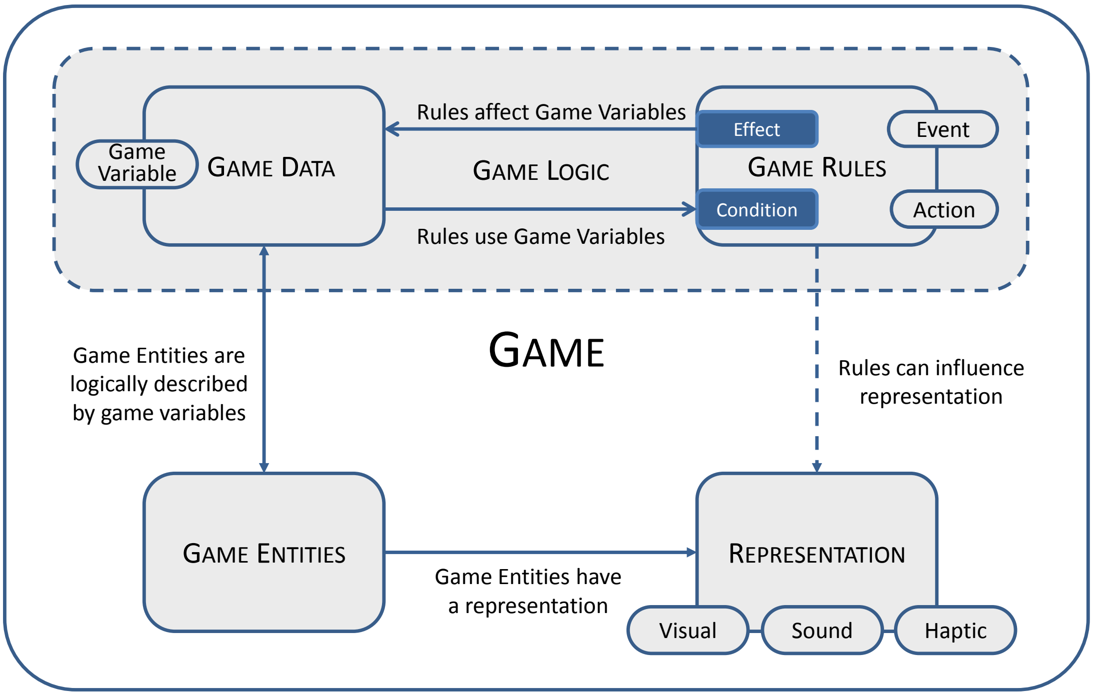

Figure 14: Generic game model showing the core elements of a game and their dependencies.

In the context of this thesis, a game is considered to consist of a *game logic*, *game entities*, and a *representation*, whereas the game logic consists of *game data* and *game rules*. Figure 14 shows the generic model of a game. Although this model does apply to the collaborative multiplayer Serious Game, it is not specific for this sub-class of games, but rather represents a generic model of a game as it does not yet consider the number of players and interaction (and hence collaboration).

4.2.3.1 *Game Entity*

Every object which exists in the game has a representation and a set of related game data. They are referred to as *game entities*. The related game data is essentially a set of elemental game variables. Thus, each game object is described completely by its associated game variables. A game entity's representation depends on the underlying technology. Usually, a game object is represented either visually, or via sound, or both. Some haptic devices might also represent a game object. In a 3D game, a visual game object consists of meshes, textures, and other structures which are created by artists to design the visual appearance of the game object. For the purpose of this thesis, visual appearance of a game entity is considered to be artistic and will not be considered in terms of the game model. Likewise, sound and haptic representation are not considered. For the game model, it is only relevant that a game entity does have a representation in the game. If, however, visual or sound properties of a game entity are of relevance, i.e., the property has a meaning beyond the artistic appearance, it is modeled using a game variable. An example for this is a glow or blink effect to highlight an entity. If the visual effect is used in the gameplay, e.g., to draw players' attention, the effect would be modeled using a 'glow' variable which is set via game rules. If, however, the effect is solely artistic and has neither an effect on gameplay nor can it be changed by the game or players, it is not considered in the context of this thesis.

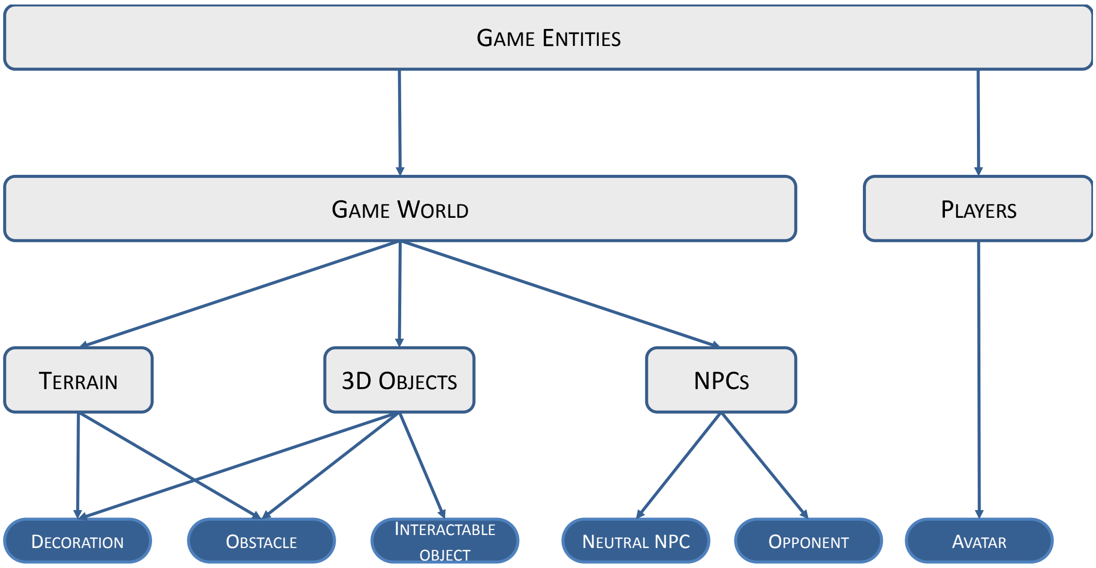

Figure 15: Avatar-based multiplayer game entities hierarchy.

The set of game entities contains all entities which appear in the game. Figure 15 contains a hierarchical overview over game entities of a generic multiplayer game. It is divided into *game world* and *players*. The game world contains all visual, corporeal, or sound related game elements, like terrain, 3D objects, or NPCs. Among those are obstacles, structures, ambiance related objects, and interact-able objects. This also contains autonomous objects which are characterized as NPCs or (AI-based) opponents or units. The set of player entities contains all human controlled player entities. They are visually represented by an avatar and in terms of gameplay described by a set of game variables, like e.g., their position in the game world or specific player parameters. The model specifically focuses on avatar-based games, i.e., games where players have a physical representation in the game world (an avatar) as opposed to simulation or real-time strategy games where players are not personally represented or control a set of units (e.g., soldiers, etc.) which are attributed to the player.

A game entity is characterized by the specifiers shown in Table 3. The *entity name* is a unique identifier for the specified entity. The *description* is a text describing the meaning and purpose of the entity in the context of the game. This information can be used by authors defining adaptations to determine an entities purpose and can be used as an explanation text for e.g., Game Masters. The *type* field specifies the entity type. The *game variables* list contains the set of game variables associated to this game entity.

Field Name	Explanation of Field	Value Type
Entity Name	Unique identifier	String
Description	Description of the entity's in-game function	String
Type	Type of the game entity	Decoration, obstacle, interactable object neutral NPC, opponent, avatar
Game Variables	List of game variables specifying the entity	List<game variable>

Table 3: Game entity specification.

4.2.3.2 *Game Data*

The set of *game data* contains all GM-relevant facts. According to the definition above, those are facts which characterize the game, like certain game parameters or player attributes. Those facts contain GM-relevant information. The entirety of game variables forms the Game State. A game variable is a unit of information about one relevant part of the game, like a parameter specifying a score, or difficulty value of a certain property. A game variable is either assigned to a game entity, i.e., describes (a part of) that game entity in terms of gameplay, or it is assigned directly to the game, i.e., is a global game variable (e.g., time, score, level, etc.). Hence, global variables can be attributed to a set of game entities (e.g., a global definition of the speed all enemies moth with). In that case, a game entity (e.g., an enemy) defines a variable which is a reference to a global variable.

Player attributes, although essentially just game parameters, are treated as a separate category for the sake of clarity. The set of player attributes contains all game parameters which directly describe the state of or directly affect that player.

Game variables are specified by the game developers during the game development. A game variable is characterized by the following specifiers:

The *variable name* is the unique identifier for the game variable. The *description* field describes the purpose of and the information contained by the game variable. The *type* field denotes the game variable type (i.e., game variable or player parameter). The *value type* field denotes the value type of the game variable and the *value range* denotes valued values for the game variable.

A *struct* is a composition if two or more simple types (string, float, integer, or boolean). An example for a struct is the position of a player avatar, which is described by three floats representing a coordinate in a 3-dimensional space.

Field Name	Explanation of Field	Value Type
Variable Name	Unique identifier	String
Description	Description of the fact	String
Type	Type of the game variable	Game variable, player parameter[ref PlayerID]
Value Type	Data type of the game variable	{String, float, integer, boolean, struct}
Value Range	Range or set of valid values	Described by [min;max] or discrete set of valid values

Table 4: Game variable specification.

4.2.3.3 *Game Rules*

The set of *game rules* contains the GM-relevant set of rules, i.e., those rules which impact the game and gameplay. GM-relevant rules are those rules which impact the core gameplay as opposed to other rules, like e.g., representation or most physics rules. Usually, physical behavior of objects in games is determined by a set of rules, like gravity. As long as gravity is not a core concept of the gameplay, it can be considered as a non-GM-relevant rule. However, if e.g., the game is about solving physical puzzles and the narrative allows for a meaningful manipulation of such a rule (like a space scenario), physics rules might be GM-relevant as the manipulation of those rules might then be a core element of gameplay.

A rule is characterized by a *condition* which needs to be fulfilled for the rule to fire and an *effect* which describes the consequences on the game(play) when the rule is fired. The *effect* is defined by an action which is executed when the rule fires.

Which rules are GM-relevant depends on game design and therefore it has to be specified by the game developers. Game rules are characterized by the specifiers shown in Table 5. A rule consists of a *rule name* which is a unique identifier for the game rule. The *description* explains the meaning of the rule in the context of the game and gameplay. The *condition* term denotes which conditions need to be fulfilled for the rule to fire. The condition is a boolean expression over game variables. Thus, the current game state determines which rules' conditions are true or false at a given point of time in the game. The specified *action* is a reference to the the action which is to be executed when the rule fires (see definitions below).

Definition: Game Condition. *A game condition is a boolean expression which evaluates to 'true' or 'false'. A boolean expression is either an elemental boolean expression or a composed boolean expression. An elemental boolean expression compares an input value with a target value using an arithmetic test operator. A composed boolean expression combines two boolean expressions with a boolean operator, whereas both boolean expressions can be elemental or composed. The arithmetic test operators used to compare input values and target values are:* $\{=, >, <, \geqslant, \leqslant, \neq\}$. *The boolean operators used to compose boolean expressions are: {and, or, not, xor}. The input value is a Game Variable and the target value is a concrete value of the input value's value range.*

Field Name	Explanation of Field	Value Type
Rule Name	Unique identifier	String
Description	Description of the rule;	String
	intended purpose	String
Condition	Condition to be fulfilled	Boolean expression
	for the rule to fire	
Action	Action to be executed	<ref action>

Table 5: Game rule specification.

Definition: Game Variable. *A game variable* $v \in V$ *is an elemental piece of information about the game. The set* V *contains all game variables. The function* ω *assigns a value of* v*'s codomain:* $\omega : v \to \{\mathbb{N}, \mathbb{R}, \mathbb{B}, [\mathbb{N}, \mathbb{R}, \mathbb{B}]^n\}$*.*

The codomain can be either natural, real, or binary numbers or a combination of n of them (for structs). Game variables can change either through game events or through *Actions*.

Definition: Action. *An action* $a \in A$ *is an elemental player activity which has a well-defined effect on the game state* GS*, whereas* A *is the set of all actions. The effect on* GS *is defined as a manipulation* α *of a subset of game variables* $V' \subseteq V$*:* $\alpha : GS \to GS$ *with* $GS \circ a = GS'$*.*

All available *actions* are defined via the game interface including relevant conditions and effects. Actions are triggered manually either by players or by the GM. Actions might have conditions which need to be fulfilled for the action to be triggerable.

Definition: Game Event. *An event is a triggered occurrence at a specified point of the game which has a well-defined effect on the game state* GS*. The occurrence is defined by the associated game condition.*

A game *event* thus can be anything which 'happens' in the game including the placement, movement, or removal of objects, sending notification, or triggering of complex events. An action thus can be used to alter the game in a desired way, either by changing a game parameter or by triggering an event which changes the game in a specified way. Table 6 shows a game rule which sends a message to a player if a certain condition is fulfilled, e.g., "you need to eat berries", if the player's satiety value is below 20.

Rule Name	$Low\_Satiety\_warning$
Description	warns players on low satiety
Condition	$satiety < 20$
Action	send message 'eatBerries'

Table 6: Example of a game rule.

4.2.3.4 *Collaborative Multiplayer Serious Game Model*

The game structure consisting of the game components as described above is shown in form of a UML diagram (see Figure 16). Game variables are shown red, game rules green, and game entities blue. This model shows the structural hierarchy of the game components and their interaction.

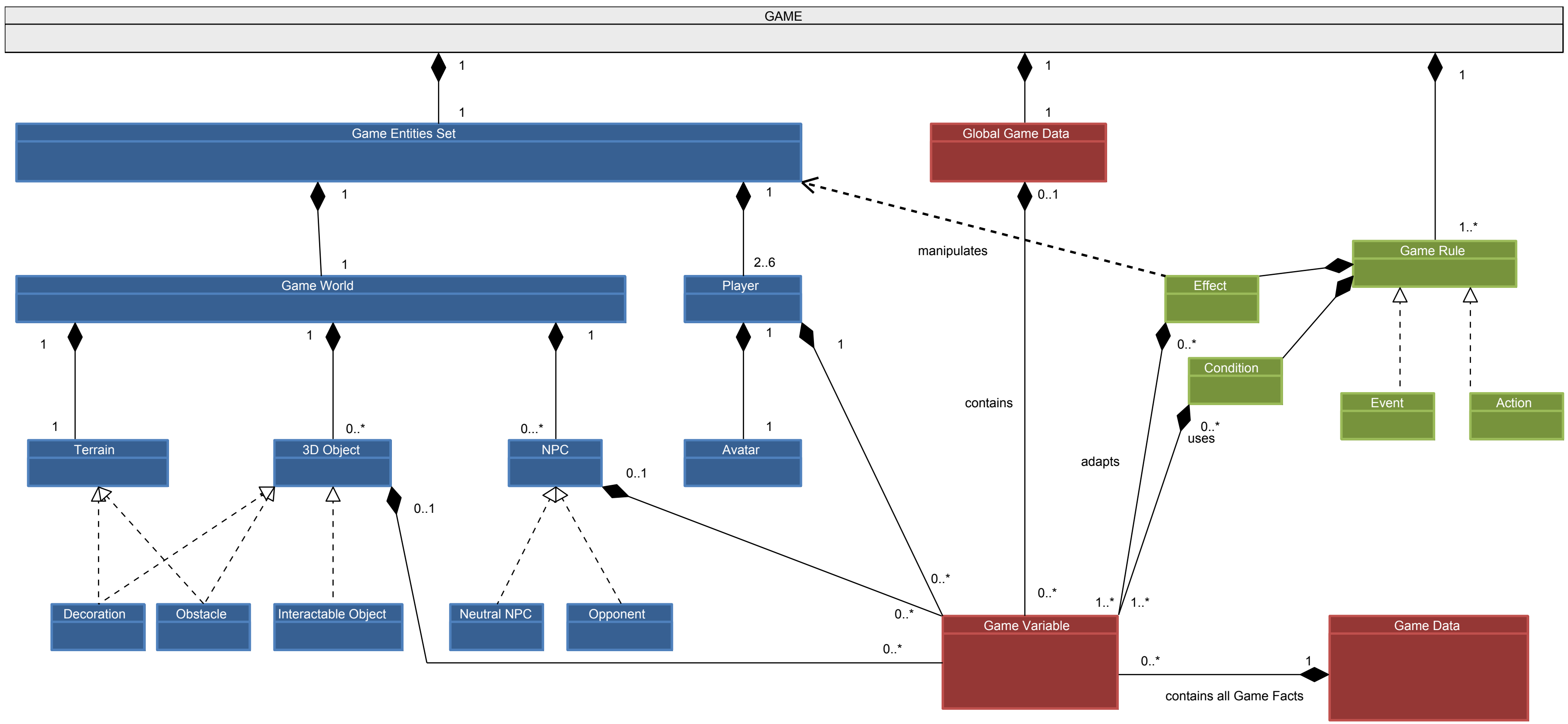

Figure 16: Collaborative Multiplayer Serious Game Model UML diagram. Game entities (left) are shown in blue, game variables (center) in red, and game rules (right) in green.

4.3 COLLABORATIVE GROUP MODEL

In order to be able to adapt the game towards players with various features like play-style, knowledge, learning preferences, communication and teamwork skills, a model of the player/learner is required. The model should comprise all relevant player features in terms of learning, gaming, and interaction (see Section 3.3). The group model is the compound of the player/learner/interaction model of all players. Relevant player features are categorized as play-related (gaming), learn-related (learning), or interaction-related (interaction).

Based on the findings of Konert et al. [91], which indicate that separate modeling for the adaptation *game flow* (playing) and *learn flow* (learning) is necessary, the approach in this thesis models all four dimensions (gaming, learning, interaction, and challenge) independently.

4.3.1 *Player Model*

The purpose of the *player model* is to define a model to describe player traits, preferences and play style which can be used to

- optimize the players' game experience and to
- predict player behavior.

4.3.1.1 *Player Model Requirements*

There are several requirements for the player model. It is desirable to be able to model individual player preferences in terms of play-style (e.g. aggressive, passive, interaction-focused, etc.). To be able to capture what players are doing and why, it is also necessary to be able to model concrete player actions and decisions in the game. Moreover, this should not be limited to one or more concrete aspects of the game(play) but rather to all game relevant aspects. The player model should be able to model player behavior without complete knowledge of the game's rule set. In other words, we do not address games with full information (like e.g., chess) from which the player model can derive a strategy. To summarize: The player model

- should take into account individual preferences
- should take into account player actions and decisions inside the game
- should be able to model as many aspects of the game(play) as possible

4.3.1.2 *Player Model Selection*

For deciding which player model is fit best for the requirements stated above, the taxonomy of player models by Smith et al. [148] (see Section 3.3) is used. They classify player models in the categories *scope*, *source*, *purpose*, and *domain* (see Table 8). The categories used by Smith et al. are explained in Table 7.

For selecting which type of player model fits best regarding the requirements above, the single categories of Smith et al.'s taxonomy are evaluated. In terms of scope, an individual player model is fitting best as each single player needs to be modeled individually.

Scope	
Individual	Applicable only to one player
Class	Applicable to a sub-population
Universal	Applicable to all players
Hypothetical	Unlikely to be applicable to any players
Purpose	
Generative	Literally produces details in place of a human player
Descriptive	Conveys a high-level description
Domain	
Game Actions	Details recorded inside of the game's rule system
Human Reactions	Details observable in the player as a result of play
Source	
Induced	Learned/fit/recorded by algorithmic means
Interpreted	Concluded via fuzzy/subjective reasoning from records
Analytic	Derived purely from the game's rules and related models
Synthetic	Justified by reference to an internal belief or external theory

Table 7: Player model categories according to [148].

In terms of purpose, a descriptive player model appears adequate as a high-level description is sufficient. Generative player models tend to model a player in one area of competency. However, a model is needed which is able to represent the player in all areas.

Regarding domain, the player model should consider player actions in the game rather than player reactions outside of the game as reactions outside of the game cannot be captured from within the game.

Regarding the source facet, an interpreted model seems hardly fit as it would require a human interpretation of the data, often using past experience and intuition. An analytic model strongly depends on game rules, often with the goal to optimize a player's behavior in terms of strategy. This would be good for a strategy game. However, for the games looked at in the scope of this thesis, this is rather impractical. Synthetic models usually reference to a concept from outside the game itself, often using "hunches, intuition, or beliefs which are not traceable to any particular piece of evidence". Therefore, induced player models appear to be most appropriate as they use recorded data and rely on machine learning or statistical analysis for interpretation.

Looking at the list of player models characterized by Smith et al. (Table 8), only three player models fulfill the requirements: The player model of Houlette [72], the Playtracer model of Andersen et al. [9], and the PaSSAGE model of Thue et al. [162]. However, the PaSSAGE player model is designed for narrative choices in a strongly story-based game. Playtracer needs a set of concrete scenes and scene transitions and

Instance	Scope	Source	Purpose	Domain
"Speedrunner" and "Completionist"	Class	Interp.	Descr.	Act.
Bartle's player types	Class	Interp.	Descr.	Both
WoW guild archetypes (Thurau)	Class	Induced	Descr.	Act.
PaSSAGE (Thue)	Class	Synth.	Gen.	React.
Storyboards (Fullerton)	Hypo.	Synth.	Descr.	Act.
Ludocore (Smith)	Hypo.	Analytic	Gen.	Act.
Houlette	Indiv.	Induced	Descr.	Act.
Playtracer (Andersen)	Indic.	Induced	Descr.	Act.
PaSSAGE (Thue)	Indiv.	Induced	Descr.	Act.
Race track generation (Togelius)	Indiv.	Induced	Gen.	Act.
NonyBots	Indiv.	Interp.	Gen.	Act.
Drivatars	Indiv.	Induced	Gen.	React.
Polymorph (Jenning-Teats)	Indiv.	Induced	Gen.	React.
Interactive fiction walkthroughs (Reed)	Indiv.	Synth.	Both	Act.
QuakeBot (Laird)	Indiv.	Synth.	Gen.	Act.
IBM's Deep Blue and Watson	Indiv.	Synth.	Gen.	Act.
Mario bots (Togelius)	Indiv.	Analytic	Gen.	React.
PaSSAGE (Thue)	Indiv.	Synth.	Gen.	React.
Heatmaps for Halo 3	Uni.	Induced	Descr.	Act.
Preference modeling (Yannakakis)	Uni.	Induced	Descr.	React.
Polymorph (Jenning-Teats)	Uni.	Induced	Gen.	React.
Engames tablebases (Bellman)	Uni.	Analytic	Gen.	Act.
EMPath (Sullivan)	Uni.	Analytic	Gen.	Act.
IMPLANT (Tan)	Uni.	Analytic	Gen.	Act
Ludocore (Smith)	Uni.	Analytic	gen.	Act.
Market bots	Uni.	Synth.	Gen.	Act.
Launchpad (Smith)	Uni.	Synth.	Gen.	Act.
EMPath (Sullivan)	Uni.	Synth.	Gen.	React.
Race track generation (Togelius)	Uni.	Synth.	Gen.	React.
Flow inspired (Czikszentmihalyi)	Uni.	Synth.	Gen.	React.
Mario bots (Togelius)	Uni.	Analytic.	Gen.	React.

Table 8: Player Model overview according to [148]. The highlighted rows mark player models which meet the postulated requirements.

also strongly depends on the metrics to evaluate distance between scenes. Therefore, the player model of Houlette appears to be the most fit player model.

Thus, the resulting player model PM is a vector of $|T|$ values $t_0...t_{|T|-1}$ representing the player values in the respective $|T|$ traits of the player model (as proposed by Houlette [72]). The set of traits T contains all traits. The player model traits are chosen in a way such that they describe the relevant player traits best regarding the game. They are defined by an expert.

$$PM = \begin{pmatrix} t_0 \\ t_1 \\ ... \\ t_{|T|-1} \end{pmatrix} with\ t_i \in [0,1]. \tag{1}$$

The following example shows a player model PM^p for a sample player p who is rather action-oriented, does not care much for dialogues, is a little curious, and very ambitious.

$$PM^p = \begin{pmatrix} 0.9 \\ 0.2 \\ 0.3 \\ 0.8 \end{pmatrix} \tag{2}$$

based on a player model consisting of the four traits *action-oriented*, *dialogue-oriented*, *curious*, and *ambitious*.

Modifications to the player model are calculated by a modified addition of a vector $m = (m_0, m_1, ..., m_{|T|})$ of length $|T|$ denoting the modification of each trait. The player model is updated according to a line by line addition of PM and m^T with $t_i^{new} = (t_i + m_i)\big|_{[0,1]}$. Results of the addition are bound to the codomain of PM: $[0, 1]$.

If, for example, the existing player model above would be changed caused by a player action with the modification vector $(0.4, 0.4, -0.4, -0.4)$, the resulting player model were:

$$PM^p = \begin{pmatrix} 0.9 \\ 0.2 \\ 0.3 \\ 0.8 \end{pmatrix} + \begin{pmatrix} 0.4 \\ 0.4 \\ -0.4 \\ -0.4 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.6 \\ 0.0 \\ 0.4 \end{pmatrix} \tag{3}$$

4.3.2 *Challenge Model*

The purpose of the *challenge model* is to model how players can cope with the different challenges which appear in a game. This can be used to optimize the challenge in terms of game difficulty (flow). The game should neither be boring nor overstrain players by being too difficult. The flow model presented here takes into account difficulty and an optimal relation between skill and challenge.

Usually, a game's difficulty at a certain point during the game session cannot be measured by one global challenge value as challenge is composed of various game aspects. Those are, among others, speed, number of tasks, number of enemies, skill or strength of enemies, type of tasks, difficulty of tasks, etc. A player might be dealing very well with some of those aspects while he/she has major problems with others. The important fact is that an over-challenge in one aspect does not compensate for an under-challenge in another one to make a good overall challenge. Rather the opposite is true. The player might be bored and frustrated at the same time, leading to an even more negative experience. Thus, it is necessary to model the different aspects of challenge in a game individually in order to be able to address them individually. The model needs further to be able to reflect changes of challenge throughout a game session (due to players getting more experienced or an adaptation of a game variable).

Subsequently, the resulting challenge model CM is a vector of $|C|$ values $c_0 ... c_{|C|-1}$ representing the player values in the respective $|C|$ categories of challenges. Categories of challenge are connected to player tasks in the game. Players can be challenged too much by a task, be under-challenged by a task, or the task may be balanced in relation to the players' skill level. The set of traits C contains all challenge categories.

$$CM = \begin{pmatrix} c_0 \\ c_1 \\ ... \\ c_{|C|-1} \end{pmatrix} with\ c_i \in [-1, 1]. \tag{4}$$

The following example shows a challenge model using the four categories of challenge in the respective game as specified below:

- Satiety
- Health
- Task_CaptureNPC
- Collect_Items

The categories are linked to player tasks which are meaningful for the game. The player needs to maintain a high value of satiety and health. The player needs to capture an NPC and should collect items.

Following, a concrete challenge model CM^p for a sample player p using the challenge categories above is shown:

$$CM^p = \begin{pmatrix} 0.8 \\ -0.5 \\ 0.0 \\ 0.8 \end{pmatrix} \tag{5}$$

This challenge model can be interpreted in the following way: The player has problems keeping his/her satiety high (as indicated by the high challenge value of 0.8). Keeping the health value high seems to be too easy for this player. The player seems to be challenged exactly right concerning the task of capturing the NPC. The player seems to have major problems collecting the items. This information will later be used by the adaptation mechanism to find suitable adaptations to make the game easier in some aspects, while making it more difficult in others.

Modifications to the challenge model are calculated by a modified addition of a vector $m = (m_0, m_1, ..., m_{|C|-1})$ denoting the modification of each challenge. The player model is updated according to a line by line addition of FM and m^T with $t_i^{new} = (t_i + m_i)\big|_{[-1,1]}$. Results of the addition are bound to the codomain of FM which is $[-1, 1]$.

If, for example the existing player model above would be changed caused by a player action with the modification vector $(0.6, 0.6, -0.6, -0.6)$, the resulting player model was:

$$CM^p = \begin{pmatrix} 0.8 \\ -0.5 \\ 0.0 \\ 0.8 \end{pmatrix} + \begin{pmatrix} 0.6 \\ 0.6 \\ -0.6 \\ -0.6 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.1 \\ -0.6 \\ 0.2 \end{pmatrix} \tag{6}$$

4.3.3 *Learner Model*

As the model should also be able to represent learning progress and knowledge, the collaborative group model contains a *learner model*. Usually, the skills to be taught within a (collaborative) learning game are reflected in (a subset of) the game's challenges. Hence, there is an interconnection between the challenges of a game and the associated skills. However, it is necessary to model both dimensions (challenge and learning) separately, as challenge represents how good players can deal with a task and learning represents the state of knowledge which is required to solve a task. The learner model refers to what knowledge to present next in order to be able to solve the next tasks, whereas the challenge model deals with the difficulty of those tasks.

4.3.3.1 *Learner Model Requirements*

The learner model should reflect all relevant skills with regard to the game. Relevant skills in this context might be knowledge skills, i.e., learning content, but also knowledge about the game, i.e., game mechanics. Whereas the former is usually defined by a Serious Game's learning content, the latter is usually defined by the game(play) itself. Game mechanic skills need to be taken into account, as they thoroughly impact player behavior and game experience and subsequently can influence the learning process. If, for example, players should learn about physics by experimenting in a 3D world, they need to be able to handle the game elements. If a player is not able to solve a learning task due to a game mechanical deficit (like not understanding controls), this firstly impacts the learning process and secondly it might frustrate the player. Moreover, the model should be able to present to which extend a player mastered such a skill. Finally, the model should be able to reflect dependencies between skills, like skills which have prerequisites or skills which should be learned in a given order.

4.3.3.2 *Learner Model Selection*

Therefore, a learner model similar to the Extended Knowledge Space Theory (see Section 3.3.6) seems appropriate as it fulfills those requirements. However, the learner model used in this thesis works on a more fine-grained level, using skills as elementary pieces of knowledge. Skills can be modeled in a hierarchical way with dependencies. Using the concepts of the *inner* and *outer fringe*, it is possible to find the set of 'interesting' skills, i.e., those skills which should be focused next on by the players.

The set Σ of skills is defined as all game relevant skills which a player can acquire. This includes skills related to learning goals as well as skills related to game mechanics. Essentially, an elemental piece of knowledge can be either acquired or not acquired. However, often it is hard to say with absolute certainty if a learner has acquired a piece of knowledge or not. Rather, a probability is used to depict to which extend a skill might be learned. Moreover, mechanical skills (game mechanic skills) should represent different stages of competence (being bad/mediocre-/good/very good in a task). To respect these facts, the co-domain of a skill $skill_x \in \Sigma$ is $[0, 1]$ instead of $\{0; 1\}$: Thus, $f : skill_x \to [0, 1]$. We say that $skill_x$ is 'learned' when $f(skill_x) \geqslant \tau$ whereas τ is a threshold with $\tau \in [0, 1]$. The current value $f(skill_x)$ of $skill_x$ is denoted as σ: $\sigma = f(skill_x)$

To reflect the fact that skills can depend on other skills, a dependency relation is introduced similar to [68] defining which skills need to be acquired before another skill can be learned. Therefore, for each skill $skill_x \in \Sigma$, a set of direct predecessors $P_{skill_x} \subseteq \Sigma$ is defined, with $\emptyset \in P_{skill_x}$. The set of predecessors $\tilde{P}_{skill_x}$ of $skill_x$ is the transitive closure of P_{skill_x}.

Thus, skills are ordered in a semi-order using a directed graph which is anti-symmetric and transitive:

$$\begin{aligned}
&\forall\, skill \in \Sigma : skill' \in P_{skill} \wedge skill \in P_{skill'} \to skill = skill'; (anti-symmetry)\\
&\forall\, skill \in \Sigma : skill \notin P_{skill}; (irreflexivity)\\
&\forall\, skill \in \Sigma : skill' \in P_{skill} \wedge skill'' \in P_{skill'} \to skill'' \in P_{skill}; (transitivity)
\end{aligned} \tag{7}$$

To avoid circles in the dependencies, the following restriction is added:

$$\begin{aligned}
&\forall\, skill \in \Sigma : skill' \in \tilde{P}_{skill} \wedge skill \in \tilde{P}_{skill'} \to skill = skill'\\
&(anti-symmetry\ regarding\ the\ transitive\ closure);
\end{aligned} \tag{8}$$

Thus, it is possible to visualize skill dependencies using Hasse diagrams as the relation R with $xRy : x \in P_y$ (see Figure 17).

Similar to the Extended Knowledge Space Theory, it is now possible to define the knowledge state of a learner as the set of skills, he/she has acquired. A knowledge state ks is defined as the set of skills $\Sigma_{ks} \in \Sigma$ which have been learned with reaching ks. A knowledge state ks' with $\Sigma_{ks'} = \Sigma_{ks} \cap skill_{new}$ is a successor to ks if $skill_{new}$ can be learned from ks. s_{new} can be learned from ks, if $\exists s \in ks\ with\ s \in P_{s_{new}}$. The set Σ_{out} contains all skills $s_{new} \| \exists_{\geqslant 1} s \in ks$ with $s \in P_{s_{new}}$. Σ_{out} is called *outer fringe*. Thus, the knowledge space can be generated from the skill relation as the set of all knowledge states and their successor relations. This, again, can be visualized as a Hasse diagram (see Figure 18).

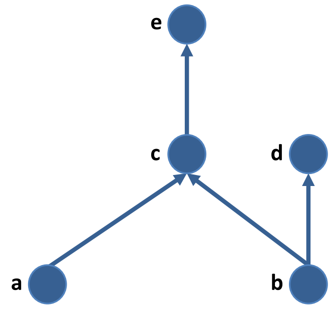

Figure 17: Example of a skill graph modeled as a Hasse diagram (lower nodes are prerequisites of higher nodes).

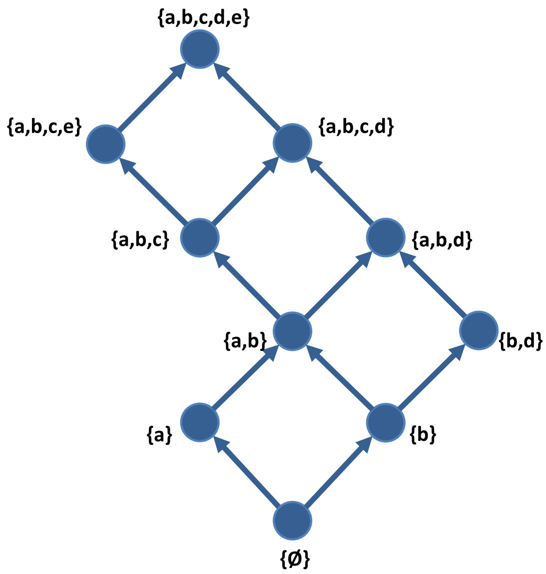

Figure 18: Example of the knowledge space for the skill graph shown in Figure 17 modeled as a Hasse diagram (lower nodes are prerequisites of higher nodes).

Therefore, the learner model structure is modeled in form of a graph depicting prerequisite relations between skills. It is defined via the set Σ, which contains all skills of the learner model and the function g: $skill_x \longmapsto \Sigma' \subset \Sigma$, which assigns a set of prerequisite skills to each skill:

$$LM = (\Sigma, g) \;\; with \; g : skill_x \longmapsto \Sigma' \subset \Sigma \tag{9}$$

The function $f : skill_x \longmapsto [0, 1]$ assigns a value σ_x to each $skill_x \in \Sigma$ depicting the player's skill value of the respective skill. Hence, the resulting learner model LM^p for player p is the vector of all $|\Sigma|$ skill values:

$$LM^p = \begin{pmatrix} \sigma_0 \\ \sigma_1 \\ \dots \\ \sigma_{|\Sigma|-1} \end{pmatrix} with \; \sigma_i \in [0; 1]. \tag{10}$$

Similar to the player model, modifications to the learner model are calculated by a modified addition of a vector $m = (m_0, m_1, ..., m_{|\Sigma|})$ of length $|\Sigma|$ denoting the modification of each skill. The learner model is updated according to a line by line addition of FM and m^T with $t_i^{new} = (t_i + m_i)\big|_{[0,1]}$. Results of the addition are bound to the codomain of LM which is $[0, 1]$.

4.3.4 *Interaction Model*

For modeling interaction, there is, to the best of our knowledge, not very much related work existing, although a lot of research covers the topic of assessing teamwork and interaction (see Chapter 3). The purpose of the *interaction model*, is to model how players interact with each other in the game. This should contain the quality of interaction, but also provide a quantitative measure.

Essentially, interaction can be divided into two basic forms: communication and game-based interaction. Communication contains all forms of verbal and written interaction, this is to say all forms of text-based chat, but also voice communication related to the game. This also includes more specific methods of in-game communication like pings[1] or character-based signs (e.g., waving). Game-based interaction contains all forms of interaction in a game between two or more players where their actions are related to each other in a semantic way. Obviously, this contains game actions where one player directly affects another player, like trading items, healing a player, or attacking/shooting at a player. However, merely following a player or walking next to each other can also be an interaction. Another commonly used example concerns puzzles. When a player pushes a button or pulls a lever somewhere in the game to open a door for another player, they are considered to interact with each other as the purpose of the first player's action affects the second player game-wise.

The definition for collaborative player interactions of Reuter et al. [130] summarizes this in the following way: "Collaborative player interactions are synchronous actions in which multiple players coordinate themselves to reach an outcome which is intended to benefit their shared goals. These interactions may consist of several smaller actions. Each action may be directed upon another player or the game world in general and their distribution may vary between the players."

The most simple and obvious metric to measuring communication and teamwork is to measure the number of communication and interaction acts. Therefore, however, it would be required that the game provides an information about which player actions are interactive. This might be easy for communication actions like pings or avatar-based communication, as well as for chat. If voice based communication is used, this is hard to be captured from inside the game. Voice communication might be outside of the game and furthermore needs to be filtered so that only game relevant communication would be used. Game-based interaction is also hard to be measured apart from some obvious interactions. The game, for example, would need to be able to decide if players are in close proximity of each other for a game-based reason or if it is just coincidental. Moreover, even if all this is possible, the mere number of interactions is not necessarily a measure of interaction and teamwork quality.

1 visually pointing towards a location of the game (usually on a mini-map) to draw the other players attention to that location

Therefore, the interaction model proposed here includes domain specific knowledge by having experts decide which player actions should be rated how in terms of communication and teamwork. The idea is to assign a value to each player action indicating if it is a sign for good communication/teamwork, or not.

Subsequently, in the interaction model IM^p for a player p, a value for 'teamwork' ι_0 and a value for 'communication' ι_1 is assigned representing the player's skill value in the respective skill, whereas $\mathcal{I} = (\iota_0, \iota_1)$ is the set of all interaction skills:

$$IM^p = \begin{pmatrix} \iota_0 \\ \iota_1 \end{pmatrix} with\ \iota_0,\ \iota_1 \in [0,1]. \tag{11}$$

Thus, the interaction model information is built directly from player actions, i.e., from how players behave in the game and from game events which are related to player performance. Therefore, function h assigns a tupel (η_0, η_1) to an action a or an event e indicating the impact of this action or event on the player's interaction model.

$$i : a \longmapsto \begin{pmatrix} \iota_0^a \\ \iota_1^a \end{pmatrix} with\ a \in A \wedge \iota_0^a,\ \iota_1^a \in [-1,1]. \tag{12}$$

An example for a player action is:

$$heal := \begin{pmatrix} 0.1 \\ 0.0 \end{pmatrix} \tag{13}$$

This indicates that the 'heal' action when performed by player would positively influence the player's teamwork skill and not affect the communication skill. An example for an event influencing the interaction model might be:

$$carry\_palm\_timeout = \begin{pmatrix} -0.2 \\ 0.0 \end{pmatrix} \tag{14}$$

This event could be triggered when the players fail to solve the task 'carry_palm' for a significant amount of time. Due to the nature of that task which requires players to collaborate to carry palms together, taking too much time to solve this task might be an indication for bad teamwork. Thus, an event like this might be useful to update the interaction models of the players if they fail to solve the collaborative task in time.

Again, analogue to the player model, modifications to the interaction model are calculated by a modified addition of a vector $m = (m_0, m_1)$ denoting the modification of each interaction skill. The player model is updated according to a line by line addition of IM and m^T with $t_i^{new} = (t_i + m_i)_{|[0,1]}$. Results of the addition are bound to the codomain of IM which is $[0, 1]$.

Although this interaction model is capable of capturing in-game communication like chat or avatar-based communication, it does not capture voice-based communication. To automatically capture and analyze voice communication, voice recognition would be required which also needed to be able to semantically analyze what players talk about. This, however, is beyond the scope of this thesis.

4.3.5 *Integrated Player Model*

The term *Integrated Player Model* refers to the combination of the player model, the challenge model, the learner model, and the interaction model for one player.

For a player p, the *Integrated Player Model* IPM^p is:

$$IPM^p = \left[\begin{pmatrix} t_0^p \\ t_1^p \\ \dots \\ t_{|T|-1}^p \end{pmatrix}, \begin{pmatrix} c_0^p \\ c_1^p \\ \dots \\ c_{|C|-1}^p \end{pmatrix}, \begin{pmatrix} \sigma_0^p \\ \sigma_1^p \\ \dots \\ \sigma_{|\Sigma|-1}^p \end{pmatrix}, \begin{pmatrix} \iota_0^p \\ \iota_1^p \end{pmatrix} \right] \tag{15}$$

4.3.6 *Group Model*

The group model is the combined model for all n players:

$$IPM = \begin{pmatrix} IPM^{p_0} \\ IPM^{p_1} \\ \dots \\ IPM^{p_n} \end{pmatrix} \tag{16}$$

An equivalent representation of the group model is given when the player, learner, flow, and interaction models of the n players are concatenated each:

$$IPM = \left[\underbrace{\begin{pmatrix} t_0^{p_0} \\ t_1^{p_0} \\ \dots \\ t_{|T|-1}^{p_0} \\ t_0^{p_1} \\ t_1^{p_1} \\ \dots \\ t_{|T|-1}^{p_1} \\ t_0^{p_{n-1}} \\ t_1^{p_{n-1}} \\ \dots \\ t_{|T|-1}^{p_{n-1}} \end{pmatrix}}_{PM_G}, \underbrace{\begin{pmatrix} c_0^{p_0} \\ c_1^{p_0} \\ \dots \\ c_{|C|-1}^{p_0} \\ c_0^{p_1} \\ c_1^{p_1} \\ \dots \\ c_{|C|-1}^{p_1} \\ c_0^{p_{n-1}} \\ c_1^{p_{n-1}} \\ \dots \\ c_{|C|-1}^{p_{n-1}} \end{pmatrix}}_{FM_G}, \underbrace{\begin{pmatrix} \sigma_0^{p_0} \\ \sigma_1^{p_0} \\ \dots \\ \sigma_{|\Sigma|-1}^{p_0} \\ \sigma_0^{p_1} \\ \sigma_1^{p_1} \\ \dots \\ \sigma_{|\Sigma|-1}^{p_1} \\ \sigma_0^{p_{n-1}} \\ \sigma_1^{p_{n-1}} \\ \dots \\ \sigma_{|\Sigma|-1}^{p_{n-1}} \end{pmatrix}}_{LM_G}, \underbrace{\begin{pmatrix} \iota_0^{p_0} \\ \iota_1^{p_0} \\ \iota_0^{p_1} \\ \iota_1^{p_1} \\ \dots \\ \iota_0^{p_{n-1}} \\ \iota_1^{p_{n-1}} \end{pmatrix}}_{IM_G} \right] \tag{17}$$

Here, p_0 to p_{n-1} denote the n players. To simplify this representation, the concatenated player, learner, challenge, and interaction models are denoted as PM_G, FM_G, LM_G, and IM_G. The index G stands for 'group'.

Subsequently, IMP can be represented as:

$$IPM = [PM_G, FM_G, LM_G, IM_G] \tag{18}$$

So, altogether the group model represents the state of all players in terms of gaming, learning, and interaction. This information can now be used to

1. inform the Game Master or
2. adapt the game,

as it shows tendencies of the players in the gaming dimension, the current state of knowledge, and the teamwork and communication skills or deficiencies. All of this information can be used by either the GM or the adaptation engine to adapt the game such that

- it fits better to the players' player models,
- it adapts challenge to prevent over-challenge or boredom,
- it presents optimal tasks to enhance the players' knowledge based on their current state of knowledge,
- it encourages teamwork and communication better.

4.4 CHAPTER SUMMARY

This chapter describes the overall approach to adressing automatic adaptation and Game Mastering in collaborative multiplayer Serious Games (Section 4.1).

As a first step, a novel model for collaborative multiplayer Serious Games is introduced, which allows game data, game rules, and game entities to be characterized independent from their representation (Section 4.2). Game entities are described more precisely to enable modeling their in-game purpose and to differentiate between gameplay relevant elements and decorative elements. A specification of a game variable and a game entity data model is provided, along with a formal specification of game rules. The term *GM-relevant* is defined as game elements that are relevant for the game; i.e., they influence the type and the flow/sequence of the game in terms of gameplay, learning content, or interaction. Core game components are identified and classified formally as *game entities*, *game variables*, and *game rules*. The described game type is modeled using those component types such that all relevant elements of a game can be represented using the model, including their dependencies and correlations. Finally, a representation of the game model using the standardized UML format is presented. This represents a solution for the question of what the relevant elements of a Serious Game are and how they can be modeled. Thus, it constitutes a foundation for the adaptation and Game Mastering concepts presented in Chapter 5 and Chapter 6.

The second major contribution of this chapter is the conceptualization of a novel unified model representing players and learners in the dimensions of gaming, learning, challenge, and interaction (Section 4.3), which enables a unified representation of player traits, state of knowledge, and interaction behavior. This model, together with the collaborative multiplayer Serious Games model, allows problems and deficiencies in the course of the game to be recognized, regarding the game itself, the learning content, and interaction between players. This information in turn can be interpreted to select appropriate adaptations (see Chapter 5).

Relevant requirements for the player model are derived from the literature. Based on these requirements, a suitable player model from the set of available player models as presented in Chapter 3 is selected. Further, it is explained why challenge should be modeled as a vector of challenge values considering the various tasks within a game rather than by a single challenge value. The learner model is based on the Competency-based Knowledge Space Theory such that player knowledge can be modeled as a set of skills and the necessity of previous knowledge can be modeled as interrelations within the set of skills. In this context, the term *outer fringe* of a skill graph is used as the set of skills that should be acquired next. A method to calculate the outer fringe from the current state of knowledge of a player is introduced. The interaction model is developed as a vector of the social skills *teamwork* and *communication*. The *integrated player model* is introduced as the combined model for one player/learner. It combines all four dimensions represented as one matrix. Finally, the group model is introduced as the aggregation of the integrated player model of all players.

5 GAMEADAPT.KOM - COLLABORATIVE MULTIPLAYER GAME ADAPTATION MECHANISM

»Play is a uniquely adaptive act, not subordinate to some other adaptive act, but with a special function of its own in human experience.«

— Johan Huizinga

In this chapter, the adaptation mechanism itself, GameAdapt.KOM is described in detail, starting with an explanantion of its core functionality in Section 5.1, followed by requirements to the game interface in Section 5.2. The adaptation goal is stated in Section 5.3 and the processing of information to calculate players' challenge is explained in Section 5.4. A core aspect of the adaptation mechanism is the automatic recognition and interpretation of game situations and player actions in Section 5.5 by GameAdapt.KOM. The definition of the game interface is provided in Section 5.6. The actual adaptation selection is presented in Section 5.7, where the metrics to rate adaptations related to player, learner, and interaction models are explained in detail and the resulting adaptation selection algorithm is explained. The GameAdapt.KOM system architecture is provided in Section 5.8.

The concept of GameAdapt.KOM directly addresses the question asked in RQ1: How can an optimal decision be made regarding the adaptation of a multiplayer (Serious) game depending on a given game situation with regard to players' traits, levels of knowledge, learning styles, and interaction?

5.1 GAMEADAPT.KOM FUNCTIONALITY

GameAdapt.KOM periodically checks which adaptation would improve the current group model the most and triggers the respective adaptation, given its improvement is greater than a defined threshold. Therefore, it needs to receive all game relevant information in order to be able to calculate the group model. Further, it needs to know how the game can be adapted, i.e., which adaptations are available and what their effects are. The frequency of the adaptation cycles depends on the underlying game and its speed.

GameAdapt.KOM updates the group model whenever it receives a game update. The evaluation of adaptations is performed based on the metrics described below. To ensure that adaptations are not executed infinitely or alternating, an adaptation's gain needs to be above a predefined threshold. Moreover, a cool-down can be specified for an adaptation to prevent repeated execution of adaptations within a short amount of time.

5.2 REQUIREMENTS

As stated above, GameAdapt.KOM needs knowledge about the game state and player actions. Thus, the adaptation algorithm needs to know about all GM-relevant *game variables* (see Section 4.2.3). and available *actions* and *events* to be triggered (*Game Interface*). The game needs to send updates to the adaptation module whenever a relevant action is performed or a game variable changes (*Information Object*). The game further needs to execute actions received by the adaptation module (*Adaptation Objects*). Figure 19 illustrates the interaction between the game and GameAdapt.KOM in a simplified way.

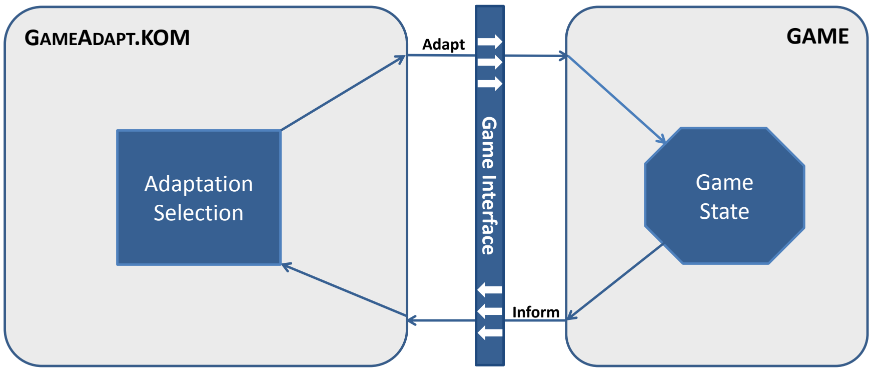

Figure 19: Simplified interaction model between GameAdapt.KOM (left) and the game (right).

5.3 ADAPTATION GOAL

The goal of the adaptation selection algorithm is to adapt the game in a way such that it better supports player, learner, and interaction features of the players/learners and optimizes the challenge. In terms of gaming, this means the game should be optimized to maximize player motivation. Therefore, again, an (almost) optimal challenge needs to be targeted in order to keep players in the *flow channel*. Moreover, the player models should be met as players tend to have more fun if the game style fits their preferences.

The learning process and learning success should be optimized. As shown before, this refers to presenting knowledge to players from the *outer fringe* of the knowledge graph. Therefore, adaptations should be selected in a way such that players are confronted with the most fitting piece of knowledge at most points throughout the game.

Optimizing teamwork, collaboration, and communication requires recognizing deficits in those aspects. Therefore, criteria for bad communication or teamwork need to be defined by subject matter experts. Those need to be measurable through player behavior and the game state. This is described in more detail later. Moreover, there need to be suitable adaptations to improve the probability of functioning teamwork and communication. Those adaptations need to emulate actions taken by real instructors in collaborative learning scenarios. Therefore, they need to be defined by subject matter experts, too. Examples are notifications and hints sent to players with meaningful situational content.

5.4 PLAYER PERFORMANCE ASSESSMENT

As stated above, it is necessary to recognize when players are over-challenged or unchallenged. The question is, how to measure how players are performing.

In order to be able to measure player performance, it is necessary to define performance criteria. Those should be defined by subject matter experts. Criteria for player performance can have various types like the time needed to solve a problem, the number of points achieved for a task, or the number of trials needed to solve a task. It could also be related to player values, like health. A low health value could be an indicator for some mistakes a player made in the past. This, however, is strongly depending on the purpose of the related variable and needs to be interpreted by experts.

The approach presented here makes use of expert defined criteria with which player performance can be measured. Such criteria are used to define game situations which help define whether players perform well in terms of learning, gaming, or interaction.

5.5 RECOGNITION OF GAME SITUATIONS

As stated above, it is vital to recognize game situations, i.e., to recognize in what situation players are at a certain point in the game. In order to be able to recognize player performance, deficits, or problems with concrete tasks or general understanding of the game goal and what to do to achieve it, it is necessary to have knowledge about what players are doing. This, however, is not trivial. Elementary actions can easily be recognized by the game, such as walking, using an item, activating or manipulating a game object, trading items, or using skills. Those actions are directly triggered by a user input like a mouse click, a key, or pressing an in-game button. Thus, they can directly be recognized. More complex player activities, like searching for an item in a 3D world, following somebody for a game-relevant reason, or tactical movement is hard to recognize. In the prototype *Escape From Wilson Island*, players need to surround an animal to force it over a cliff. Therefore, it is required that players move in a coordinated way. There is, however, no game action which starts the hunting, like pressing a 'hunt'-button. A human might easily be able to conclude what the players are doing using observation and human reasoning. Yet, it is necessary to recognize such situations automatically to be able to judge how good the players are solving this task. It needs to be possible to define how much time players are spending on the task and how many trials they took.

The situation recognition mechanism developed in this thesis uses elemental game data like game variables, player parameters, or elemental game actions, like moving, or triggering events by pressing a button. Based on those, *game states* and *tasks* are defined. Game *situations* are defined using different kinds of *criteria* which indicate that the situation is currently present. Finally, all situations are evaluated periodically, calculating a probability for each situation to be present at a certain point in the game session.

5.5.1 *Basic Definitions*

Following, the elemental parts of the situation recognition are defined. This includes a definition of a game variable, the game state, an action, a shape, an task, a region, the term criterion and the various types of criteria, and finally a situation.

Definition: Game Variable (repetition). *A game variable* $v \in V$ *is an elemental piece of information about the game state. The set* V *contains all game variables. The function* ω *assigns a value of* v*'s codomain:* $\omega : v \rightarrow \{\mathbb{N}, \mathbb{R}, \mathbb{B}, [\mathbb{N}, \mathbb{R}, \mathbb{B}]^n\}$*.*

Game variables can change either through game events or through player *actions*.

Definition: Game State. *The game state* GS *is a concrete allocation of all game variables* V *in the game:*

$$GS = \begin{pmatrix} v_0 \\ v_1 \\ ... \\ v_{|V|} \end{pmatrix} \tag{19}$$

Hence, the game state changes whenever a game variable $v \in V$ changes.

Definition: Action (repetition). *An action* $a \in A$ *is an elemental player activity which has a well-defined effect on the game state* GS*.* A *is the set of all actions. The effect* e *on* GS *is defined as a manipulation* α *of a subset of game variables* $V' \subseteq V$*, with:*

$$\alpha : GS, a \rightarrow GS \text{ with } GS + e_a = GS' \tag{20}$$

The effect e_a of action a is a vector of $|V|$ variables $d_0 ... d_{|V|}$ describing how the variables of the game state change:

$$e = \begin{pmatrix} d_0 \\ d_1 \\ ... \\ d_{|V|} \end{pmatrix} \tag{21}$$

An action can be as simple as 'move forward' triggered by pressing the respective key, or 'gather berries' triggered by clicking on a berry bush game object. Note: Although in a game an action might have a number of prerequisites which have to be fulfilled for the action to be executable (e.g. a player needs to be within a certain distance to a game object in order to trigger the object), this is no relevant information here. The game itself takes care about checking the prerequisites. GameAdapt.KOM only needs to know when an action was performed.

Definition: Shape. *A shape* $s \in S$ *describes a boolean expression defining game variable states.* S *is the set of all shapes.*

A shape is a boolean expression over a set of game variables which evaluates to *true* or *false*. Shapes are used to express the combination of a set of game variables to form conditions. For example, if it should be expressed that some game action should be performed when a player's satiety is lower than 30% while the player has berries in his/her inventory, this could be expressed via $s_x = (satiety \leqslant 30 \wedge berriesInInventory > 0)$. A shape s is considered *active* if its boolean expression evaluates to true, else *inactive*. The values of the variables used in the boolean expression are taken from the current game state GS as it contains all game variables. The function f assigns {active, inactive} to a shape s depending on the value of its boolean expression.

$$f : (s, GS) = \text{ active, if } s(GS) = true, \text{ else inactive} \tag{22}$$

According to the above definition, at every point in the game, n shapes can be active simultaneously with $n \in [0, |S|]$.

Definition: Task. *A task* $\tau \in \mathcal{T}$ *is defined as the tuple* $\tau = (\Sigma, \sigma_s, \sigma_a, \delta)$ with $\sigma_a \in \Sigma$, $\delta : \Sigma, A, S \to \Sigma$. *The set of tasks* $\mathcal{T}$ *contains all tasks.*

A *task* is a well-defined assignment which has to be fulfilled by one or more players. A task consists of a set of states Σ, a start state $\sigma_s \in \Sigma$, an active state $\sigma_a \in \Sigma$, and a transition function δ which defines the successor state for each state in Σ based on an action or event and a required configuration of variables s (shape), i.e., $\delta(\sigma, a, s) \to \sigma'$ with $\sigma' \in \Sigma$. Whenever an action or event a_{now} is executed, GameAdapt.KOM checks whether a task τ's active state σ_a changes.

Definition: Region. *A Region is a well-defined area in the game world.*

The specific definition of a region is left to the game. The relevant information about a region is whether one or more game-relevant entities are inside an area. This information needs to be accessible by GameAdapt.KOM. A region can be described as a cube or sphere which can be reduced to squares or circles on the game terrain, if players cannot move in the third dimension independently from the terrain (e.g., fly).

In order to be able to evaluate which situations are currently present, a set of criteria is used. A situation is defined to be true if all of its criteria are fulfilled.

Definition: Criterion. *A criterion* $c \in C$, *whereas* C *is the set of all criteria, is an item which defines if a game condition (see Section 4.2.3) is fulfilled.*

For each criterion, an evaluator function *ev* continuously evaluates to which extend the criterion is fulfilled. Many criteria can either be 0 or 1 as they are evaluated in a binary way (*true* or *false*). In the following section, the various types of criteria and their codomain are explained.

5.5.2 *Criteria*

Atomic Criterion

A criterion which is directly retrievable from a game variable, e.g., 'Is player x moving?'. An atomic criterion refers to the question if a game variable has a specified value. Thus, it can only be evaluated to exactly 0 or 1, respectively *false* or *true* for the related condition:

$$ev(c) = 1, \text{if the respective game variable has the desired value, else 0.} \tag{23}$$

Local Criterion

A *local criterion* is considered fulfilled if the respective player is in the related *region*. A player can either be in the specified *region* or not. Hence, the *local criterion* can only be evaluated to exactly 0 or 1, respectively *false* or *true* for the related condition:

$$ev(c) = 1, \text{if the a player is in the specified area, else 0.} \tag{24}$$

Global Criterion

A *global criterion* is based on the *local criterion* type. It contains a set of *regions* and defines the relationship between them. Possible relationship types are:

1. Visiting a set of specified *regions* (either by one player or by several players).
2. Being in a set of specified *regions* at the same time (several players).

For the former relationship, the following evaluation function is used:

$$ev(c) = \frac{|regions\ visited|}{|regions\ to\ be\ visited|} \text{ with } |regions\ to\ be\ visited| > 0 \tag{25}$$

For the latter relationship, the criterion can be either fulfilled or not, thus:

$$ev(c) = 1, \text{if all areas are being occupied by at least one player, else 0.} \tag{26}$$

Distance Criterion

A *distance criterion* is based on the distance between a player and another player/object/region. Therefore, a target (player, object, or region) and a value $distance_{max}$ is specified. The distance criterion then evaluates using the current distance $distance_{cur}$ between the player and its target and the maximum distance $distance_{max}$ to its target.

$$ev(c) = 1 - \frac{\min(distance_{cur}, distance_{max})}{distance_{max}} \tag{27}$$

Time Criterion

The *time criterion* can only be fulfilled at a certain point in time (this point can actually be a time span, like e.g., 'at night'). Therefore, two points in time are defined: t_{min} and t_{max}. Let t_{now} denote the current point in time. Then

$$ev(c) = 1, \text{if } t_{min} \leqslant t_{now} \leqslant t_{max} \tag{28}$$

Note: t_{now}, t_{min}, and t_{max} can refer to a continuous time scale (global time) or recurring time (e.g., time of day). If it refers to a continuous time scale, there can only be one time interval where $ev(c) = 1$. Otherwise, there can be a time span with $ev(c) = 1$ every cycle.

INTERVAL CRITERION
The *interval criterion* is used to measure the time distance since the last occurrence of an event or action. Therefore, an action or an event is specified, as well as a $max\ distance$ value. Let t_{now} denote the current point in time, and $t_{lastocc}$ the point in time when the action or event occurred the last time. Then:

$$ev(c) = \frac{\min((t_{now} - t_{lastocc}), 1)}{max\ distance} \tag{29}$$

SHAPE CRITERION
Uses a concrete *shape*, i.e., is considered fulfilled if the respective *shape* is active:

$$ev(c) = 1, \text{if the shape is 'active', else 0.} \tag{30}$$

ACTION CRITERION
Is considered true if the respective action is performed or the respective event is triggered:

$$ev(c) = 1, \text{if the action or event is true, else 0.} \tag{31}$$

ATTRIBUTE CRITERION
The *attribute criterion* is based on *player attributes*. If a *player attribute* has a certain value, is above or below a certain threshold, or is in a certain range, the criterion is considered fulfilled. Therefore, x_{min} is specified as a lower threshold, and x_{max} is specified as an upper threshold. Let x_{now} be the current value of the player attribute.

$$ev(c) = 1,\ \text{if}\ x_{min} \leqslant x_{now} \leqslant= x_{max}, \text{else 0.} \tag{32}$$

TASK CRITERION
The *task criterion* is used to make a task a prerequisite of a situation. The *task* needs to be either in the state 'not started', 'ongoing', or 'finished'. Therefore, $x_{task} = \{not\_started, ongoing, finished\}$. Let x_{now} be the current state of the task:

$$ev(c) = 1,\ \text{if}\ x_{task} = x_{now},\ \text{else 0.} \tag{33}$$

5.5.3 *Situations*

Definition: Situation. *A situation* sit *is a point of interest in the game which can be considered interesting or relevant due to its meaning for the game or game purpose. A situation is defined via a set of criteria* C_{sit}*:*

$$sit := C_{\text{sit}} \subseteq C \tag{34}$$

A situation is considered partially present if at least one of its criteria is fulfilled. A situation is either caused by one player or by the whole group.

In contrast to shapes, situations are not easily tangible by defining a boolean expression over a set of concrete game variables. Rather, situations are used to describe vague incidences during the course of the game. Those incidents would usually be categorized by a human person which is able to recognize and judge player behavior and game situations.

Note: in contrast to a shape, which has two concrete states: *active* and *inactive*, a situation has a continuous value range of $[0, 1]$ indicating the probability that the situation is present. Table 9 contains the data structure for a situation in GameAdapt.KOM. A situation is described by a unique identifier (name). It contains a semantic annotation as an explanation, describing what game situation is to be captured (description). It contains the field 'Caused by' which defines if the situation can be created by a single player or by the whole group. Further, it contains a list of criteria including a weighting factor for each criterion.

	Explanation of field	Type
Name	Unique Identifier	String
Description	Description of the Situation	String
Caused by	'Player' or 'Group'	String
$[w_0]\mathrm{Criterion}_0$ ... $[w_n]\mathrm{Criterion}_n$	Criteria to be fulfilled for this situation to be true (weighting w in [0,1])	$C_{sit} \subseteq C$

Table 9: Situation data structure.

5.5.4 *Situation Recognition Metric*

For each situation, GameAdapt.KOM calculates how likely it is that the situation is currently present. Therefore, GameAdapt.KOM evaluates the situation's criteria. For each criterion, an evaluator function ev assigns a value between $[0, 1]$ (see above):

$$ev : C \longrightarrow [0, 1] \tag{35}$$

Additionally, each criterion has a weighting modifier w to express that different criteria have a different impact on a situation. It is possible to mark criterion as obligatory, hence stating that the criterion needs to be fulfilled for the situation to be present regardless of other criteria. Therefore, a criterion is marked with a weighting of 0. The function obl evaluates if at least one of a situation's obligatory criteria is not fulfilled. For an obligatory criterion to be considered fulfilled, $ev(c) \geqslant 0.5$ is required.

$$\mathrm{obl}(c) = 0, \text{ if } ev(c) < 0.5, \text{ else } 1. \tag{36}$$

Hence the term

$$\widehat{\mathrm{OBL}} = \prod_{i=0}^{|C_{sit}|-1} \mathrm{obl}(c_i) \tag{37}$$

evaluates to 0, if at least one obligatory criterion is not fulfilled. The situation sit's degree of presence $p(sit)$ thus is:

$$p(sit) = \frac{\sum_{i=0}^{|C_{sit}|-1} w_i * ev(c_i)}{\sum_{i=0}^{|C_{sit}|-1} w_i} * \widehat{OBL} \tag{38}$$

5.5.5 *Situation Examples*

Following, two examples will be presented showing the use of criteria to define a situation:

Example 1: Game Skyrim - Buying a house in the city of Whiterun:
Table 10 shows a situation describing whether a player is about to buy a house in the Skyrim city of Whiterun using the criteria c_0 to c_4.

Name	Buying house in Whiterun
Description	The player is about to buy a house in the city of Whiterun
Caused by	Player
$[3]c_0$:	Local Criterion: 'Is the player in Whiterun?'
$[1]c_1$:	Task Criterion: 'Does the player hold the title 'Thane' of Whiterun?'
$[1]c_2$:	State Criterion: 'Did the player already slay the first dragon?'
$[1]c_3$:	Distance Criterion: 'Is the player close to the NPC Proventus Avenicci?'
$[1]c_4$:	Inventory Criterion: 'Does the player possess more than 5000 pieces of gold?'

Table 10: Situation example - buying a house in the game *Skyrim*.

If all of the criteria listed above are fulfilled, there is a high probability that the player is on his/her way to buy a house in Whiterun. If, however, only some of the criteria are fulfilled, the player however is not Thane, it is still possible that he/she is trying to buy the house, but just does not yet know that he needs to be Thane. Or, he/she approaches the NPC because he/she wants to buy something else. Subsequently, the probability that the player is about to buy the house is still relatively high.

Example 2: Game EFWI - Building the log hut:
Table 11 shows a situation describing if the players are about to build the log hut using the criteria c_0, c_1, and c_2.

If all of those criteria are fulfilled, it can be assumed with a very high probability that players are trying to finish building the log hut. If, however, only a part of those criteria is fulfilled, the players are probably trying something else (like building the

Name	Building log hut
Description	The players are about to build the log hut
Caused by	Group
[3]c_0 :	Local Criterion: Are the players close to the hut building area?
[1]c_1 :	State Criterion: Is the first part of the hut built already?
[1]c_2 :	Atomic Criterion: Is a large palm being carried?

Table 11: Situation example - building the hut in the game *EFWI*.

raft, or carrying logs to make firewood) or they, for example, do not know where they have to build the hut. Therefore, the probability that the players are actually building the log hut is lower, but significantly above zero.

5.5.6 *Situation Object*

A *situation object* defines the consequences of the presence of *situations*. It is used to formally describe effects and consequences of problems and poor player performance. Those consequences can be the change of player models, learner models, flow models, or interaction models. In detail, it can mean that player traits, challenge, learner skills, or interaction skills of one or more players have changed. $f : sit \mapsto IMP$

It is possible to define a condition which is repeatedly applied after a specified cool-down ('Cool-down' field). Therefore, an integer is assigned defining the cool-down. If *cool-down* is 0, the situation can only occur once. If it is >0, the situation can re-occur every '*cool-down*' seconds, i.e., the consequences are applied every *cool-down* seconds. The cool-down mechanism prevents consequences to be applied every frame, if a situation is evaluated to be present for a longer duration. GameAdapt.KOM keeps track of how many times a situation was present, and how long it was present in total (weighted by the probability that it was fulfilled). Thus, it is possible to react to problems or poor player performance which is usually characterized by not being able to solve tasks in the given time. In Table 12, the data structure of a *situation object* is shown. The effect on the player model is described in form of a vector of length k with k being the number of traits in the player model. For each trait t_i with i in $\{0, k\}$, a modifier mod_i^t specifies how the related trait should be modified. mod_i^t hereby specifies a value in $[-1, 1]$ which is added to the current value of t_i.

Analogously, the effect on the flow model is specified in form of a vector of length m with m being the number of challenges in the flow model. For each challenge c_i with i in $\{0, m\}$, a modifier mod_i^c specifies how the related challenge should be modified. mod_i^c hereby specifies a value in $[-1, 1]$ which is added to the current value of c_i. For increased readability, challenges with modifiers of 0 are omitted in the vector.

The effect on the learner model is also specified in form of a vector of length l with l being the number of skills in the learner model. For each skill σ_i with i in $\{0, l\}$, a modifier mod_i^σ specifies how the related skill should be modified. mod_i^σ hereby specifies a value in $[-1, 1]$ which is added to the current value of σ_i. For increased readability, challenges with modifiers of 0 are omitted in the vector.

Analogously to skills, the effect on the interaction model is specified in form of a vector of length 2, as there are exactly two interaction skills. For both interaction skills, a modifier mod_i^{η} specifies how the related interaction skill should be modified. mod_i^{η} hereby specifies a value in $[-1, 1]$ which is added to the current value of η_i with i in $\{0, 1\}$.

An example can be found in Section A.5.

	Explanation of field	Value
Situation Name	Ref. to the name of the situation	String
Cool-down	Cool-down on effects applied?	$\mathbb{N}$
Effect on player model	How does the occurrence of the situation effect player models?	$\begin{pmatrix} \mathrm{mod}_0^t \\ \mathrm{mod}_1^t \\ \ldots \\ \mathrm{mod}_{\lvert T\rvert-1}^t \end{pmatrix}$
Effect on challenge	How does the occurrence of the situation effect challenge?	$\begin{pmatrix} \mathrm{mod}_0^c \\ \mathrm{mod}_1^c \\ \ldots \\ \mathrm{mod}_{\lvert C\rvert-1}^c \end{pmatrix}$
Effect on associated skills	Which skills are affected by this situation and how?	$\begin{pmatrix} \mathrm{mod}_0^{\sigma} \\ \mathrm{mod}_1^{\sigma} \\ \ldots \\ \mathrm{mod}_{\lvert \Sigma\rvert-1}^{\sigma} \end{pmatrix}$
Effect on associated interaction skills	Which interaction skills are affected by this situation and how ?	$\begin{pmatrix} \mathrm{mod}_0^{\eta} \\ \mathrm{mod}_1^{\eta} \end{pmatrix}$

Table 12: Situation object data structure.

5.6 GAME INTERFACE

The game interface describes the connection and exchange of information between the game and GameAdapt.KOM.

Information exchange happens in two phases:

1. Before the game
2. During the game

In Phase I, the game the game tells GameAdapt.KOM

- The set of relevant game variables and player parameters
- The set of relevant game entities
- The set of trigger-able actions and events, i.e., game rules

Game variables', game entities', actions' and events' structures are described in Section 4.2. Actions and events are special game rules. Player parameters are parameters which describe a player's state in terms of the game (i.e., health, score, etc.).

In addition to this data, the player models, learner models, and interaction models need to be inititalised. If this information is available - either due to a prior assessment via tests, questionnaires, etc. or via the instructor's estimation of the players/learners - it can be configured. Otherwise neutral values will be used at the start of the game. For skills it then is assumed that they are not known, i.e., 0 is used as standart value. The same goes for the interaction skills. For the player models, all traits are set to 0.5. All challenge values are always initialized with the neutral 0 (optimal challenge).

This knowledge is required for defining meaningful adaptations and situations in a game. Experts can use available game variables, game entities, actions, and events and design relevant game situations and useful adaptations which use and adapt exactly those game elements.

In Phase II - during the game's run-time - the game continuously sends updates to GameAdapt.KOM informing it about changes to values of game variables, entities, etc. or the execution of actions or events. GameAdapt.KOM sends adaptations to the game when necessary, telling it to update game variables, entities, or to execute actions or events.

5.6.1 *Game Element Negotiation*

As stated above, at the start of the game (Phase I) the game tells GameAdapt.KOM about relevant game elements. Thereto, the following structure is used:

	Explanation of field	Value
Name	Unique identifier	String
Description	Description of the game variable; Purpose in terms of gameplay, etc.	String
Type	Game variable / player action / game event	<String, Float, Integer, Boolean> / Action type / Event type
Value range / Effect	Describes the valid range of the parameter (i.e., minimum, maximum) or the action/event's effect	Range described by [min; max] or a discrete set of valid values

Table 13: Game element object data structure.

A game element has a unique identifier 'name', a description of what information it contains, a type, and a value range of the type. If 'type' = *game variable*, the value is of type <String, Float, Integer, Boolean>. If 'type' = *player action*, the value is of type

<Action type>. If 'type' = *game event*, the value is of type <Event type>. The game defines the set of relevant player *action* and *event* types which exist in it. An example can be found in Section A.1.

5.6.2 *Information Objects*

Information objects are the game's way to update GameAdapt.KOM about changes. They are sent at run-time (Phase II) to inform GameAdapt.KOM about changed game variables, parameters, actions, or events. This information is then used by GameAdapt.KOM to calculate *situations* and to subsequently update the group model. An information object provides information about changes of relevant game elements like game variables, actions, or events.

	Explanation of field	Value
Name	Unique Identifier	String
Value	Game variable value	<String, Float, Integer, Boolean>
	Player action parameters	Action type
	Game event parameters	Event type

Table 14: Information object data structure.

An information object is described by a unique identifier 'name', which refers to the unique name of the *game variable, action, or event* which it updates/executes. If it refers to a game variable, then the value contains the new value of thatgame variable. If it refers to an action or event, the value field contains possible action/event parameters.

5.6.3 *Adaptations*

From the knowledge about existing game variables, game entities, and game rules, Game-Adapt.KOM is able to derive in what ways the game can be adapted. Based on this knowledge, *adaptations* can be defined. An adaptation is either a concrete manipulation of a game fact or an entity, or the triggering of an event, or the execution of a concrete action. An adaptation is described using the following data structure as shown in Table 15. The data structure contains the fields 'name', 'description', 'game element', and 'parameters'. The name of an adaptation is a unique identifier. The description explains the function of the adaptation. The game element specifies if the adaptation references a game variable, a game action, or a game entity. If the game element specified is a player, this means that the adaptation only concerns that player. This is used for most messages containing hints or tips for a player. If the specified game element is a non-player game element or variable, it is considered to affect the whole group. The required parameters specify detailed input data about the adaptation. For each parameter, either a concrete value (e.g., 0.5) or a modification of the current value (e.g., +0.1) are possible if the parameter's value type is Float or Integer.

An example for an action can be found in Section A.2. An example for an adaptation can be found in Section A.6.

	Explanation of field	Value
Name	Unique identifier	String
Description	Description of the adaptation; Purpose in terms of gameplay, etc.	String
Game Element	Game variable name Game action name Game entity name	Reference to a concrete game variable, game event, or game entity (e.g. player)
Parameter(s)	Required parameters	For each required parameter a concrete value or modifier

Table 15: Adaptation data structure.

5.6.4 *Adaptation Objects*

Adaptation Objects (AOs) are the counterpart to information objects. Whereas information objects provide GameAdapt.KOM with information and status updates, GameAdapt.KOM uses *adaptation objects* to manipulate the game in the desired way. The structure is shown in Table 16. *Adaptation objects* are described using the following fields: 'Name', a unique identifier, the name of the adaptation object. The 'Description' field is a semantic annotation containing the AO's purpose and desired effect. The 'Prerequisite' field contains an optional prerequisite condition which needs to be fulfilled for the AO to be executable. The prerequisite condition is a boolean function over various game variables, acquired skills, or situations. Only if the function evaluates to *true*, the AO is allowed to be used. The 'Player Model' field denotes in what situation an adaptation object should be used. Therefore, adaptation objects are attributed related to the player model to be able to state for which player types it fits best. If the AO does not consider the player model (i.e., player model is irrelevant) this can omitted. 'Effect on Challenge': This field contains the AO's effect on the game's challenge. It can be omitted if the AO does not affect challenge. The associated skills field contains all skills which are affected (including social skills) by this AO and the effect as a modifier (e.g., $+0.1$) of the current skill value. For simplicity, only skills which a modifier $\neq 0$ are listed. The adaptation field is a reference to the related adaptation which will be executed by the game if this AO is used. An example can be found in Section A.6.

5.7 ADAPTATION SELECTION

The metrics to decide about appropriateness of adaptation objects are described next. For each adaptation object ao, a fitness value F is calculated depending on the current group model IPM. The adaptation object with the highest fitness value is chosen. For simplicity the value domain of F is restricted to $[0, 1]$ so that an optimal value is $F = 1$ and the worst result is $F = 0$.

	Explanation of field	Value
Name	Name of the adaptation object	String
Description	Description of the Adaptation Object	String
Prerequisite	A condition which needs to be fulfilled for this AO to be executable	<Condition>
IsGlobal	Defines whether this adaptation object is applied to the group or to a player	Boolean
Player Model	Suitability regarding the player model (note: for each trait t_x there is a modificator a_x^t; a_x^t can be 0)	$a^t = \begin{pmatrix} a_0^t \\ a_1^t \\ \dots \\ a_{\|T\|-1}^t \end{pmatrix}$
Effect on Challenge	How does the AO effect challenge?	$a^c = \begin{pmatrix} a_0^c \\ a_1^c \\ \dots \\ a_{\|C\|-1}^c \end{pmatrix}$
Associated Learning Skills	Which learning skills are affected by this AO and how?	$a^\sigma = \begin{pmatrix} a_0^\sigma \\ a_1^\sigma \\ \dots \\ a_{\|\Sigma\|-1}^\sigma \end{pmatrix}$
Interaction Skills	Which interaction skills are affected by this AO and how ?	$a^\eta = \begin{pmatrix} a_0^\eta \\ a_1^\eta \end{pmatrix}$
Adaptation	Game Variable / Action / Event /Game Entity	<Adaptation Identifier>

Table 16: Adaptation object data structure.

For the four dimensions learning, gaming, challenge and interaction, a fitness value depending on the player model (PM_p), flow model (FM_p), learner model (LM_p), or interaction model (IM_p) is calculated for each player p.

If the adaptation object is not global (i.e., IsGlobal = false), the overall fitness F_p value regarding one player p of the adaptation object ao is the weighted sum of the fitness values in terms of playing (G_P), challenge (C_P), learning (L_P), and interaction (I_P):

$$\begin{aligned} F_p(ao, IPM) =& \alpha \cdot G_p(ao, PM_p) + \beta \cdot C_p(ao, CM_p) \\ &+ \gamma \cdot L_p(ao, LM_p) + \delta \cdot I_p(ao, IM_p) \\ &\text{with } \alpha, \beta, \gamma, \delta \in [0, 1] \text{ and } \alpha + \beta + \gamma + \delta = 1 \end{aligned} \tag{39}$$

If the adaptation object is global, it is not evaluated for each individual player. Instead, the overall fitness value for the group F_G is the weighted sum of the weighted fitness values in terms of playing P_G, challenge C_G, learning L_G, and interaction I_G with

$$\begin{aligned} P_G &= \frac{1}{n}\sum_{j=1}^{n} P_i(ao, PM_j), \\ C_G &= \frac{1}{n}\sum_{j=1}^{n} C_i(ao, CM_j), \\ L_G &= \frac{1}{n}\sum_{j=1}^{n} L_i(ao, LM_j), \\ I_G &= \frac{1}{n}\sum_{j=1}^{n} I_i(ao, IM_j). \end{aligned} \tag{40}$$

$$F_G(ao, IPM) = \alpha \cdot P_G + \beta \cdot C_G + \gamma \cdot L_G + \delta \cdot I_G \tag{41}$$

5.7.1 *Player Model Metric*

The goal of this part of the metric is to evaluate an adaptation object in relation to the group's player model (i.e., $(P = 1)$ if it fits well to the group's player model). Thus, the discrepancy between the group model and the adaptation object's suitability vector a^t (see Table 16) should be minimal for an optimal value.

For measuring how close two vectors are, there are different metrics.

The discrepancy can be measured using the Manhattan-metric d_M, which simply is the sum of the absolute values of the differences between the two vectors v and w:

$$d_M(v, w) = \sum_i |v_i - w_i| \tag{42}$$

Alternatively, the squared euclidean distance d_E can be used which sums the quadratic values of the differences between the two vectors v and w. This punishes bigger differences harder:

$$d_E(v, w) = \sum_i (v_i - w_i)^2 \tag{43}$$

It should be noted that the squared euclidean distance is not considered a metric because it does not satisfy the triangle inequality.

Further, the cosine similarity is a measure of similarity between two vectors. It measures the cosine angle between the two vectors with -1 being the maximum distance between the two vectors, and 1 meaning the two vectors are identical.

$$d_C(v, w) = \frac{v \cdot w}{\|v\|\,\|w\|} = \frac{\sum_i v_i \cdot w_i}{\sqrt{\sum_i (v_i)^2} \cdot \sqrt{\sum_i (w_i)^2}} \tag{44}$$

As there is no reason to punish bigger differences much stronger, it was decided to use a modified version of the Manhattan metric which subtracts the difference from 1, such that a minimal distance results in a maximum value:

$$d'_M(v, w) = \frac{1}{n}\sum_{i=1}^{n}(1 - |v_i - w_i|) \tag{45}$$

Therefore, the resulting metric to calculate the appropriateness of an adaptation object ao in terms of gaming for *one player* p is:

$$\begin{aligned} P_p(ao, PM_p) &= \frac{1}{|T|}\sum_{i=1}^{|T|}\left(1 - \left|a_i^t - t_i\right|\right) \\ &a_i^t\text{: suitability of } ao \text{ regarding trait } t_i; \\ &|T|\text{: number of traits in } PM; \end{aligned} \tag{46}$$

The metric to calculate the appropriateness of an adaptation object ao in terms of gaming for *the group* is the weighted sum of the appropriateness $F_G(a, PM_j)$ over all n players affected by ao:

$$\begin{aligned} P_G(ao, PM_G) &= \frac{1}{n}\sum_{j=1}^{n} P_j(ao, PM_j) \\ &n\text{: number of players affected by } ao \end{aligned} \tag{47}$$

5.7.2 *Challenge Model Metric*

The goal of this part of the metric is to make sure that an adaptation object is considered useful in terms of challenge. It needs to express both when a challenge is too high and when it is too low. Hence, the domain for each dimension of challenge is $[-1, 1]$ (see Section 4.3). c_x^{new} denotes the new value of challenge c_x, if the AO ao is applied, limited to the domain $[-1, 1]$.

$$\begin{aligned} c_x^{new} &= (c_x^{now} + a_x^c)\big|_{[-1,1]} \\ c_x^{now}&\text{: the current value of challenge} \\ a_x^c&\text{: the modification of the current challenge value } c_x^{now} \end{aligned} \tag{48}$$

To consider that an adaptation could change a challenge value from too easy to too difficult (or vice versa), if the adaptation is stronger than the challenge difference from 0, the term c_x^δ describes an absolute improvement of challenge c_x through AO a towards o:

$$c_x^\delta = |c_x^{now}| - |c_x^{new}| \tag{49}$$

The appropriateness calculation should reflect that larger variations from 0 should be weighted stronger (i.e., it is more important to correct a challenge which is much too high than a challenge which is only a little bit too high). Hence, c_x^δ is weighted by multiplying it with $(1+|c_x^{now}|)^2$. With $c_{now} \in [0,1]$, follows $(1+|c_x^{now}|)^2 \in [1,4]$. Therefore, the result is subtracted by 1 and divided by 3, so that $C_p(a^c, c)$ is in $[0,1]$.

The appropriateness value C considering *one challenge* c_x therefore is:

$$\begin{aligned} C(a_x^c, c_x) &= \frac{1}{3}\left(|c_x^{now}| - \left|(c_x^{now} + a_x^c)|_{[-1,1]}\right|\right) \cdot \left((1+|c_x^{now}|)^2 - 1\right) \\ &= \frac{1}{3}\left(|c_x^{now}| - |c_x^{new}|\right) \cdot \left((1+|c_x^{now}|)^2 - 1\right) \\ &= \frac{1}{3} c_x^\delta \cdot \left((1+|c_x^{now}|)^2 - 1\right) \end{aligned} \tag{50}$$

The appropriateness value *for all challenges* for *one player* p is the sum of all appropriateness values for the challenges c_x with $a^{c_x} \neq 0$:

$$C_p(ao, CM_p) = \frac{1}{3} \sum_{\substack{0<i<|C| \\ i|a_i^c \neq 0}} c_i^\delta \cdot \left((1+|c_i^{now}|)^2 - 1\right) \tag{51}$$

a_i^c: the influence of a on the challenge value c_i

$|C|$: set of challenges

The overall appropriateness value of the AO ao for *the group* is then the weighted sum of all appropriateness values of all players affected by ao:

$$C_G(ao, CM_G) = \frac{1}{n} \sum_{j=1}^{n} C_j(a, CM_j) \tag{52}$$

n: number of players affected by ao

5.7.3 *Learner Model Metric*

When deciding about how good an *adaptation object* is fit for the current state of knowledge of the group, the associated skills of the AO are taken into account. As shown before, it should be considered that players are only confronted with new knowledge from the *outer fringe* of their state of knowledge. They should not be confronted with knowledge concerning skills which are considered too difficult. Also, they should not be confronted with knowledge about skills they are considered to have learnt already. Figure 20 explains which skills in the skill graph are part of the outer fringe, already learned, or too difficult. To respect this, a modification to a skill caused by an AO ao.

The term gain defines what is gained AO regarding a $skill_x$ on the *outer fringe*.

$$gain(skill_x, a_x^\sigma) = \begin{cases} a_x^\sigma & ,\text{if } \sigma_x + a_x^\sigma \leqslant 1 \\ 1 - \sigma_x & ,\text{if } \sigma_x + a_x^\sigma > 1 \end{cases} \tag{53}$$

σ_x: the current value of $skill_x$

a_x^σ: modification of σ_x caused by ao

The metric diff denotes how good an AO ao is fit for a skill. For a skill on the *outer fringe*, this is the calculated gain. For a skill which is considered too difficult

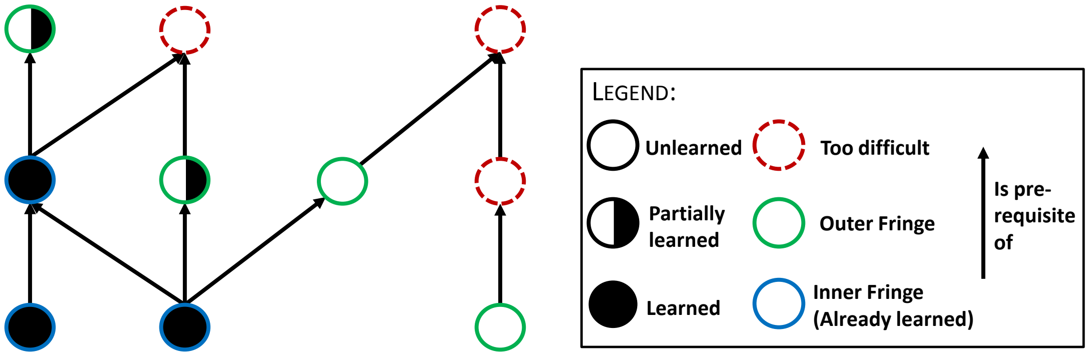

Figure 20: Outer Fringe of a knowledge graph.

at the current state of knowledge, this is $-a_x^\sigma$. This gives a penalty in the amount of taught knowledge which is considered too difficult. It is neutral (0) for all other skills (i.e., skills already learned).

$$\mathrm{diff}(\mathrm{skill}_x, a_x^\sigma) = \begin{cases} \mathrm{gain}(\mathrm{skill}_x, a_x^\sigma) & \text{, for all skills } \in \textit{ outer fringe} \\ -a_x^\sigma & \text{, for all skills } \in \textit{ too difficult} \\ 0 & \text{, else} \end{cases} \tag{54}$$

σ_x: the current value of skill_x

a_x^σ: modification of σ_x caused by ao

The overall appropriateness value in regard of the learner model of *one player* p is then the sum of the appropriatess values for all skills skill_x with $a_x^\sigma \neq 0$:

$$L_p(ao, LM_p) = \sum_{\substack{0<i<|\Sigma| \\ i|a_i^\sigma \neq 0}} \mathrm{diff}(\mathrm{skill}_i, a_i^\sigma) \tag{55}$$

a_i^σ: the influence of ao on the skill value σ_i

$|\Sigma|$: set of skills

The overall value for the learner model of the group is then the weighted sum over all specified players affected by ao:

$$L_G(ao, LM_G) = \frac{1}{n} \sum_{j=1}^{n} L_j(ao, LM_j) \tag{56}$$

n: number of players affected by ao

5.7.4 *Interaction Model Metric*

In terms of interaction, AOs are evaluated on the fact of how they affect the interaction skills. Analogous to the learner model, for each interaction skill in IM the new value $\eta_x + a_x^\eta$ is calculated which represents the new skill value after the adaptation was performed.

The term $gain$ defines what is gained by AO regarding a $skill_x$.

$$\mathrm{gain}(\mathrm{skill}_x, a_x^\eta) = \begin{cases} a_x^\eta & \text{if } \eta_x + a_x^\eta \leqslant 1 \\ 1 - \eta_x & \text{if } \eta_x + a_x^\eta > 1 \end{cases} \tag{57}$$

η_x: the current value of $skill_x$

a_x^η: modification of η_x caused by a

The average value of all new interaction skills is the measure for F_I. Thus, the resulting metric to evaluate F_I for *one player* p is:

$$I_p(ao, LM_p) = \mathrm{gain}(\mathrm{skill}_{Teamw.}, a_{Teamw.}) + \mathrm{gain}(\mathrm{skill}_{Comm.}, a_{Comm.}) \tag{58}$$

The resulting metric to evaluate F_I for *the group* is then:

$$I_G(ao, IM_G) = \frac{1}{n}\sum_{j=1}^{n} I_j(IM_j, a) \tag{59}$$

n: number of players affected by ao

5.7.5 *Adaptation Selection Algorithm*

The adaptation selection algorithm periodically runs the three steps shown in Figure 21.

1. Filter adaptations based on their preconditions
2. Rate remaining adaptations according to the metric above
3. Select the adaption with the highest rating and execute it

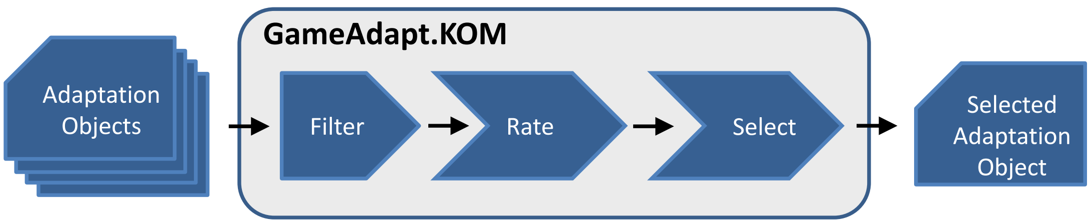

Figure 21: GameAdapt.KOM adaptation selection procedure.

Listing 1 shows the adaptation selection algorithm.

Listing 1 Adaptation Selection

```
Require: A, FA, EA          ▷ A are all adaptations, FA are all filtered adaptations
  for all a ∈ A do          ▷ EA is the map of adaptations and their respective values
    if precon(a) = true then
      FA = FA ∪ a
    end if
  end for
  for all a ∈ FA do
    Calculate F(IPM, a)
    EA = EA ∪ {a, F(IPM, a)}
  end for
  return MAX(EA)            ▷ return the adaptation with the highest value
```

5.7.6 *Adaption Selection Computing Time Estimation*

In the following section, an estimation of the upper bound for the computing time for the adaptation selection will be made.

From the adaptation selection algorithm it can be seen that computing time can be split into three parts:

1. Adaptation filtering
2. Adaptation fitness calculation
3. Sorting

Adaptation filtering scales with the number of adaptations $|\mathcal{A}|$ as each adaptation is checked once per adaptation cycle for its condition. It further scales with the number of game variables in an adaptation's condition. This number can vary widely. However, an upper boundary is the total number of game variables in the game. The number of possibly used game variables in an adaptation's condition is $\mathcal{O}(|V|)$. This upper boundary is still valid when assuming that variables can be used more than once in a condition. Hence, adaptation filtering can be performed in

$$\mathcal{O}(|A| \cdot |V|). \tag{60}$$

Adaptation fitness calculation also scales with the number of adaptations $|\mathcal{A}|$ as each adaptation is evaluated once per adaptation cycle for its fitness to be executed. In each fitness evaluation, the adaptation's fitness regarding the player model $F_G(PM, a)$, the flow model $F_F(FM, a)$, the learner model $F_L(LM, a)$ and the interaction model $F_I(IM, a)$ are evaluated. Hence, adaptation selection can be performed in

$$\mathcal{O}(|A| \cdot |P| \cdot (|T| + |C| + |\Sigma| + |\mathcal{I}|)) \tag{61}$$

, whereas $|P|$ is the number of players.

As $|P| \leqslant 6$ can be assumed for the scenario described in this thesis, it can be omitted in the $\mathcal{O}$ calculus:

$$\mathcal{O}(|A| \cdot (|T| + |C| + |\Sigma| + |\mathcal{I}|)) \tag{62}$$

The selection of the best suitable adaptation with fitness calculated can be done in n steps, whereas n is the number of evaluated adaptations. At a maximum, this can be $|A|$ adaptations. Hence, selecting can be performed in $\mathcal{O}(|A|)$.

Altogether, an upper boundary for the adaptation selection is:

$$\begin{aligned}\mathcal{O}(\mathrm{filter} + \mathrm{fitness} + \mathrm{select}) &= \mathcal{O}(|A| \cdot |V| + |A| \cdot (|T| + |C| + |\Sigma| + |\mathcal{I}|) + |A|) \\ &= \mathcal{O}(|A| \cdot (|V| + (|T| + |C| + |\Sigma| + |\mathcal{I}|) + 1))\end{aligned} \tag{63}$$

As the number of traits $|T|$, the number of challenges $|C|$, the number of skills $|\Sigma|$, and the number of interaction skills $|\mathcal{I}|$ is constant, this term can be simplified to $\mathcal{O}(|A| \cdot |V|)$.

5.8 GAMEADAPT.KOM ARCHITECTURE

Figure 22 provides an overview over the relevant elements of communication between the game and GameAdapt.KOM.

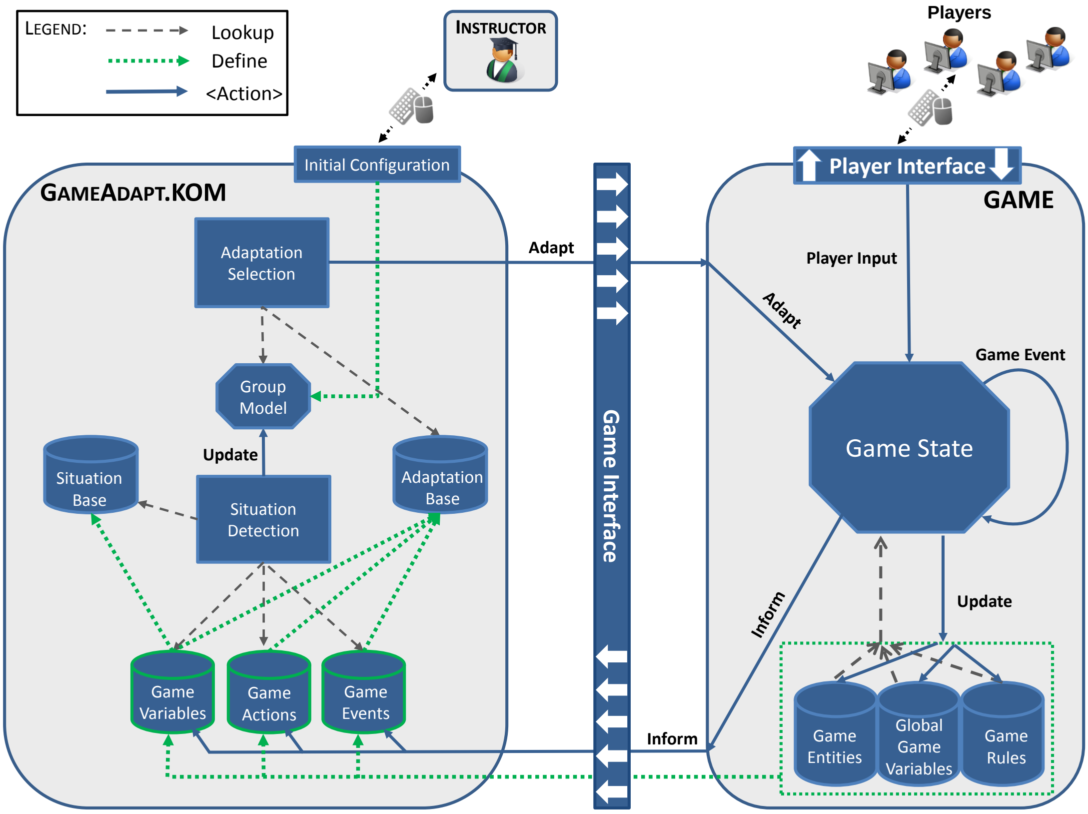

Figure 22: GameAdapt.KOM architecture and interaction with game.

The blue box on the right side represents the game. The game contains the sets of *game entities*, global *game variables*, and *game rules* (see Section 4.2). Moreover, the game maintains the *game state* for which it continuously looks up the *game entities*, global *game variables*, and *game rules*. The *game state* is influenced by *player input*, *game events*, and *adaptations*. Whenever the game state changes, the game informs GameAdapt.KOM and updates the sets of *game entities*, global *game variables*, and *game rules*. For communication with the player, the *player interface* is used. For communication with GameAdapt.KOM, the *game interface* is used.

The blue box in the left represents GameAdapt.KOM. The core entities are the *situation detection*, and the *adaptation selection*. GameAdapt.KOM holds an instance of *game variables*, *game actions*, and *game events* which are defined in Phase I (green). Based on those, an expert can define the set of situations and adaptations (also in Phase I, green). Moreover, GameAdapt.KOM holds the *group model*. The *group model* is updated continuously whenever the *situation detection* recognizes the presence of one or more *situations*. The *adaptation selection* periodically evaluates the set of *adaptations*. Therefore it looks up the set of *adaptations* in the *adaptation base* and the current state of the *group model*. If an adaptation should be executed, it sends the selected *adaptation* to the game.

Figure 23 depicts the interaction between GameAdapt.KOM and the game as a sequence diagram for:

1. Initialization: receiving game relevant entities, variables, and actions
2. Game Updates: the game tells GameAdapt.KOM that something relevant happened
3. Adaptation selected: GameAdapt.KOM sends an adaptation to the game

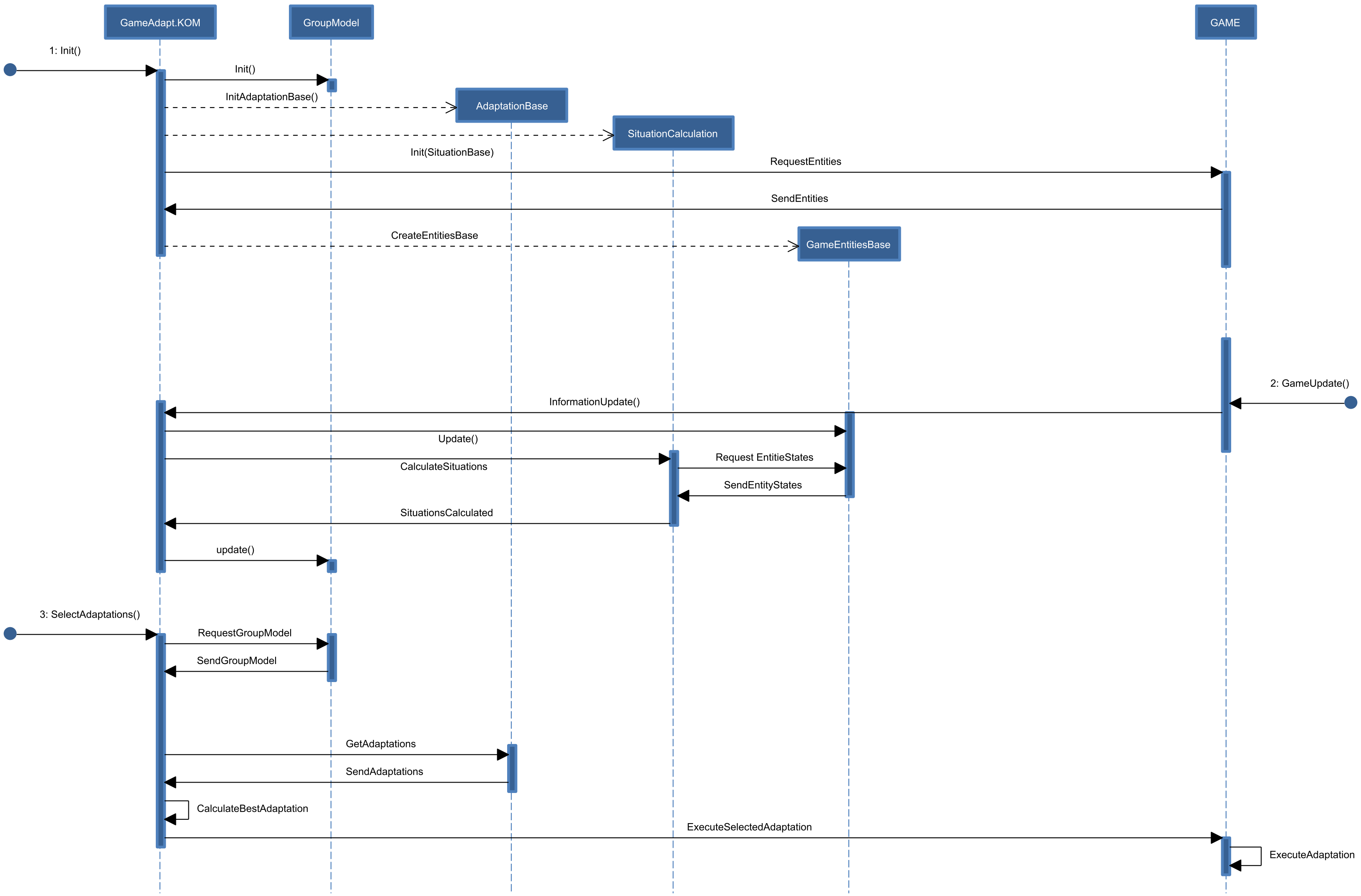

Figure 23: GameAdapt.KOM adaptation selection sequence diagram. The first part shows the initialization phase. The second part shows how game updates are processed. The third part shows the adaptation selection.

The GameAdapt.KOM adaptation mechanism is modelled as a stand-alone component which interacts with the related game via the game interface. Whereas this already enables an expert to use game variables, entities, and rules to model situations, situation objects, adaptations and adaptation objects via a minimalistic user interface, it might be useful to implement this user interface in the related game. This allows to have a unified Graphical User Interface (GUI) for the instructor when using both the Game Master interface (see Chapter 6) and the adaptation interface.

5.9 CHAPTER SUMMARY

This chapter covers the concept of automatic game adaptation - GameAdapt.KOM. GameAdapt.KOM represents a novel approach to automatically adapt collaborative multiplayer Serious Games based on the state of the game in order to optimize player experience, the learning process, and the interaction process. Thus, it directly addresses Research Question 1. The core functionality is described in Section 5.1. Requirements regarding the game to be adapted are identified (Section 5.2). Based on that, a formalization of accessibility requirements of game elements is presented.

The adaptation goal is formally described as an optimization of adaptable game elements with the goal of optimizing player experience, the learning process, and the interaction process. In terms of gaming, this means the game is to be optimized to maximize player motivation and to optimally fit gaming preferences. In terms of learning and interaction, this means that knowledge is presented in a way that presents an optimal learning path for players. Therefore, again, challenge needs to be optimized, whereas challenge refers to player skills and knowledge (see Section 5.3).

To measure and quantify how well players are performing, criteria for player performance aspects are defined. These criteria are used as prerequisites for adaptations to be executable (see Section 5.4). A prerequisite to measure player performance is the ability to recognize what players are doing in the context of the game.

A major contribution in this context is the conceptualization of a situation recognition mechanism in collaborative multiplayer Serious Games, which algorithmically detects in which situation players are at a certain point during the game (Section 5.5). The situation recognition automatically derives player intentions and goals from the detected situations. This information is used to update the collaborative player model.

Situations are formally defined using definitions for *game variables*, *player parameters*, the *game state*, *actions*, *states*, *tasks*, *regions*, and *criteria*. A situations data structure is formally defined using the situation specification model introduced in this section. A situation detection metric is developed to assign a probability value to each possible situation indicating the likeliness of that situation to be present in the current game state.

Situation objects represent a formal specification of a situation's effects on challenge, the player model, and the learner model of the group. The situation recognition concept allows specific game situations to be algorithmically detected throughout a game. Usually, for this task, human perception and reasoning would be required as player actions and movements cannot easily be matched to player intentions. That is to say that in an open game world with players moving around freely, it is often a complex task to decide what players are doing and why. The situation recognition concept is a first approach to recognize this algorithmically without the need for human reasoning.

A formal definition of an information exchange interface between the game and GameAdapt.KOM is derived in Section 5.6 based on requirements of the adaptation process. Thus, it defines how information is exchanged between these two components. A structure for adaptations is defined to meet the descriptive requirements of the adaptation's purpose and uses the game model's structure to define how the adaptation impacts the game. With the developed concept to formally define adaptations and their impact on the game, it is now possible to define a set of adaptations using only the existing and accessible elements, rules, and variables of the game and to define how the state of the game will be influenced by executing the respective adaptation. Therefore, again, only the available and accessible game elements, rules, and variables, as well as the group model, are necessary. This provides a mechanism for manipulating a game in a desired way based on defined situations by using only those parts of the game which are made accessible via the game interface without deeper knowledge about internal processes and game mechanics.

The second major contribution is the development of the adaptation selection algorithm (Section 5.7). A metric was developed to rate adaptations for their suitability considering the current state of the game and the collaborative group model. Further, an adaptation selection algorithm first filters adaptations based on preconditions, rates them using the adaptation selection metric, and executes the optimal adaptation. The developed adaptation selection algorithm makes it possible to automatically decide when and in what way a game should be adapted based on the current state of the game and the current group model. Hence, it is now possible to adapt a game with the goal of optimizing game experience, learning success, interaction skills, and challenge automatically - i.e., without the need to have a human person observe the game process, judge player performance and possible problems, and decide whether and how to counteract or interfere.

Lastly, to comprehensibly describe the interaction and interdependencies between the single parts described in this section, Section 5.8 introduces the GameAdapt.KOM system architecture and explains its components and the relations between them.

Summarizing, the contribution of this chapter is the development of a concept to adapt a game automatically. This includes the development of a method to automatically recognize game situations based on the current game state and group model, as well as a concept for the definition of adaptations, including their impact on the game and an optimal selection of the most suitable adaptation using an adaptation selection metric that takes into account players' preferences, knowledge states, and interaction skills, as well as individual challenge. Hence, research question 1 is addressed in this section.

6 GAME MASTERING INTERFACE

»I am always ready to learn although I do not always like being taught.«

— Winston Churchill

This chapter covers the conceptualization of the Game Mastering interface. As a first step, the requirements towards the interface are identified based on typical instructor tasks in collaborative learning and gaming scenarios (Section 6.1). Resulting from those requirements, the information (Section 6.2) and adaptation (Section 6.3) interfaces are designed. Finally, it is shown how the concept of Game Mastering is integrated into GameAdapt.KOM (Section 6.4).

6.1 REQUIREMENTS

In order to be able to provide an instructor with the necessary means for orchestrating a collaborative multiplayer Serious Game, it needs to be clarified,

- what information about a game, the players and the course of the game need to be made available to the GM,
- what options of influencing and adapting a game need to be made available to the GM for the GM to be able to optimize the learning session according to his/her professional opinion.

This directly referes to RQ3. Hence, the design of the Game Mastering interface is driven by an analysis of requirements for manual Game Mastering on collaborative multiplayer Serious Games. In Section 2.1, instructor tasks are elaborated. A result from that analysis is a classification of instructor tasks. In collaborative learning scenarios, instructor tasks include moderation, monitoring, coaching, analysis, and intervention (see [61], p. 51). In general, instructor tasks can be divided into two categories: *observation* and *adaptation*. Observation refers to tasks which require acquisition and processing of information. Adaptation refers to influencing the group and the learning process. Thus, tasks like monitoring or analysis can be classified as observation, whereas moderation, coaching, guidance, and intervention are classified as adaptation (see Table 17).

Observation	Adaptation
Monitoring Analysis	Moderation Coaching Intervention

Table 17: Instructor task classification.

Transferred to game-based learning scenarios, those tasks implicate several requirements:

1. A Game Master needs to be able to obtain relevant information from a game.
2. A Game Master needs to be able to influence the learning process in the game.

In Section 3.5, the role of a Game Master is discussed. The term Game Master is originally taken from pen&paper role-playing games where the GM obtains the role of a story facilitator. The GM in a pen&paper role-playing game is responsible for promoting the narration, while including players into the story progression. The core problem is the so-called *Narrative Paradox*, the conflict between player freedom and pursued narration. Whenever players, in the role of actors in a narration, have complete freedom regarding the choice of their actions, it is nearly impossible to keep a the narration going to where the author intended it to go beforehand. To solve this conflict, a good GM needs one vital skill: The ability to improvise. A good Game Master is able to adapt the previously designed story in a way such that the overall narration still goes into the desired direction while not making players feel like they were mere puppets. It is of utmost importance that players have the feeling that their actions make an impact on the course of the game and narration. In order to be able to improvise, the Game Master in a pen&paper role-playing game needs to have

- full control over the game,
- rich background knowledge about the game, the story, and the characters,
- and knowledge about the current state of the game.

Resulting from that rich background knowledge, the GM is able to estimate consequences of his/her actions enabling him/her to make suitable decisions.

This concept of Game Mastering in pen&paper role-playing games can be used in collaborative multiplayer Serious Games to identify requirements of a Game Master interface. The assumption made in this thesis is that a GM overseeing a collaborative multiplayer Serious Game has similar needs and requirements as Game Masters in pen&paper role-playing games and instructors in collaborative learning scenarios. Therefore, the GM in a collaborative multiplayer Serious Game also needs to be provided with the same

- full control over the game,
- rich background knowledge about the game, the story, and the characters,
- and knowledge about the current state of the game.

Whereas 'full control over the game' refers to meaningful adaptation mechanisms and possibilities. The two remaining bullet points refer to gathering information about the game and presenting it to the GM. The goal of the Game Master interface hence is to provide the GM with that information.

6.2 INFORMATION INTERFACE

Basically, there are two ways of providing the Game Master with information about the state of the game, the players, and what is happening in the game.

The first is providing the GM with an in-game view in form of a special camera view. The idea is to provide the GM with a camera view with which he/she can observe the game either in a bird's-eye perspective or another freely controllable perspective to observe the game world or at least the relevant part of the game world. This provides an overview over game entities. However, this method needs to be implemented entirely in the game.

The second way is using information sent by the game. This information is processed to calculate situations and to evaluate a useful adaptation. The results of the situation recognition as well as suggested adaptations can be made available to the GM. Moreover, game variables can be provided directly. They can be visualized in a simple fashion. In a similar way, game rules can be visualized. However, visualization might be a little more complex. This, however, is rather a problem of visual presenting of information and will have to be solved in the concrete implementation. The group model can be presented to the GM in order to provide him/her with more insight about the group's player model, state of learning, and interaction profile. For all of those methods, visualization can be done in a generic Game Master front-end. However, as there is already a need to implement a Game Master view in the game itself, to provide the bird's-eye or similar camera perspective, it seems appropriate to provide all of that information inside the game in a dedicated GM front-end graphical user interface.

In summary, the following information provision methods have been identified (see Table 18):

	Method	Provided Information
In-game	(Bird's-eye) camera	Game world Game entities
Via GameAdapt.KOM	Situation recognition	Player actions
	Group model	Player characteristics
	Game variables and events	Game state
	Game rules	Current game rules

Table 18: Game Master information provision methods.

Whereas graphical information about the game (via camera perspectives) needs to be provided by the game itself, all other pieces of information can be provided by GameAdapt.KOM which gathers and aggregates all of that information (game variables, state of the game and players, situation recognition, etc.).

6.3 ADAPTATION INTERFACE

For provision of adaptation mechanisms, it stands to reason to use the already in GameAdapt.KOM defined and encapsulated adaptations. As GameAdapt.KOM contains a set of available adaptations, this repository can be made available to the Game Master. However, some of those adaptations are defined directly for changing game variables, like improving a game variable about a fixed factor. Instead of providing this type of adaptations, it appears to be more intuitive to provide direct access on the underlying game variable. Moreover, for some adaptations, ease of use might be improved if there was some kind of support for it in the game. For example, placing or moving a game entity can be done via an interface provided by GameAdapt.KOM where the GM specifies position parameters. However, placing or moving an object in a game, i.e., in a 3D environment can be done more intuitively when the GM can do it directly in the 3D world.

It is further possible to recommend suitable adaptations to the Game Master, as GameAdapt.KOM does evaluate and select suitable adaptations, anyway. Instead of triggering the selected adaptation, the adaptation - or a set of best fitting adaptations - can be suggested to the GM. The GM can then decide if one of those adaptations should be used or not. Similar to the provision of information, adaptations mechanisms can be provided to the GM via a GameAdapt.KOM front-end. Yet, again, it seems appropriate to integrate a front-end into the game itself which makes use of the GameAdapt.KOM Game Mastering interface.

In summary, the following adaptation mechanisms can be provided to the GM using GameAdapt.KOM (see Table 19):

	Method	Allows Adaptation of
In-game	(Support of adaptation placement	Game entities
Via GameAdapt.KOM	Direct access to game variables	Game state Players
	Direct access to game rules	Game state Players
	Adaptation recommendation	–

Table 19: Game Master adaptation methods provision.

6.4 GAMEADAPT.KOM GAME MASTERING INTERFACE

In the following section, the enhancement to GameAdapt.KOM for enabling a human GM to use GameAdapt.KOM to gather information about the game and to manually adapt the game is explained. This contains the interface between GameAdapt.KOM and the GM as well as access to GameAdapt.KOM information (i.e., game variables, situations, group model) and GameAdapt.KOM adaptations. This interface enables a Game Master to manually adapt a game according to his/her professional opinion. Figure 24 shows the enhanced GameAdapt.KOM architecture.

As before, GameAdapt.KOM receives information from the game via the game interface. GameAdapt.KOM keeps that information locally, i.e., stores game variables, game actions, and game events. This information is directly presented to the GM, possibly aggregated, if necessary. Further, information about the likeliness of situations to be present is forwarded to the GM. The group model is updated as before and presented to the GM. Moreover, the GM can access the set of adaptations. The adaptation selection module, however, is disabled, as adaptations are only triggered by the GM.

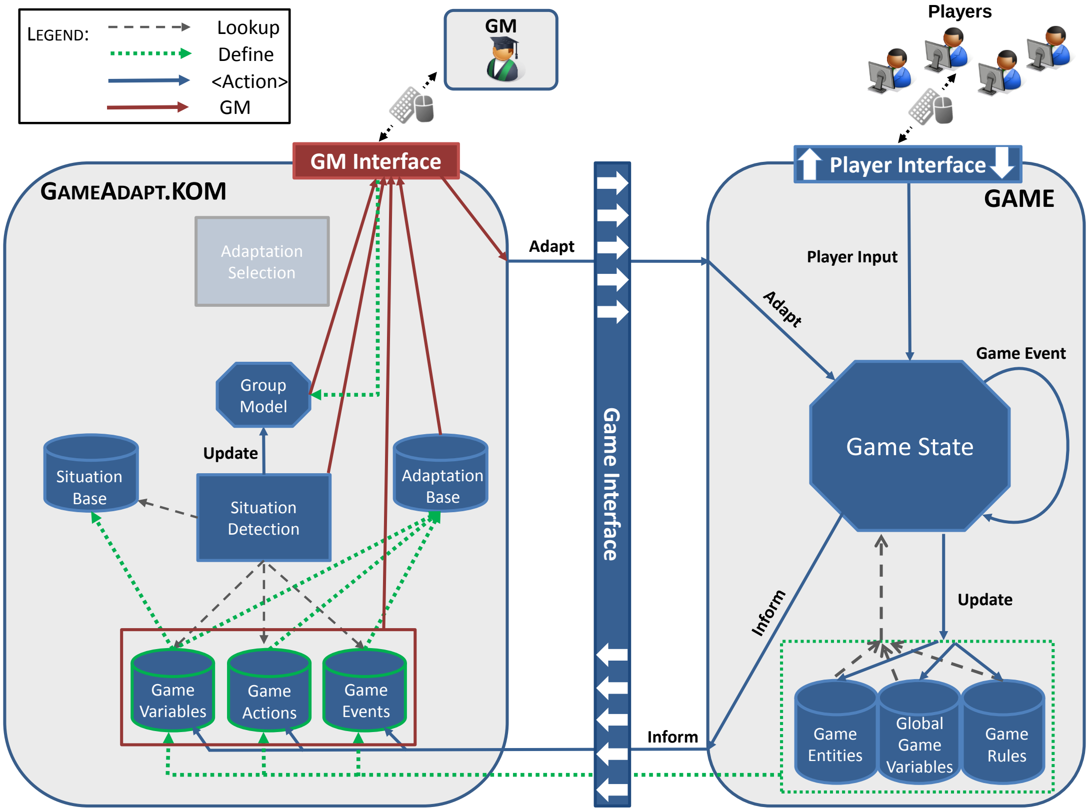

Figure 24: Enhanced GameAdapt.KOM system architecture. Shows interaction between GameAdapt.KOM and game. Changes to the previous architecture shown in red.

Thus, GameAdapt.KOM functions as a black-box providing and aggregating information about the game for the GM, as well as providing adaptation possibilities for the GM.

Although the Game Master interface is logically modeled as a frontent component for GameAdapt.KOM, it might be more practical to have an actual implementation of the frontend as an implementation in the underlying game to enable access to the in-game camera, and to enable the GM to move within the game world.

6.5 CHAPTER SUMMARY

This chapter describes the concept for Game Mastering in collaborative multiplayer Serious Games, which represents an enhancement of GameAdapt.KOM. The questions of what information about a game is required and which means of adaptation need to be provided for a GM, are addressed. Thus, this chapter addresses Research Question 3. Based on the literature review from Chapter 3, instructor tasks in collaborative learning scenarios, as well as Game Mastering concepts, are defined and summarized in Section 6.1.

In Section 6.2, the interface to provide information retrieved from the game, is defined. The design of the interface is motivated by a specification of the requirements a GM has when overseeing a game session. The requirements are considered from a collaborative learning scenario perspective as well as from a pen-and-paper roleplaying game perspective. From the aggregated requirements, the goal of the GM interface is derived as providing the GM with the means to gather complete information about the game state and the players and to provide him/her with the ability to adapt the game such that the GM can be considered to have full control over the course of the game.

A concept is developed to enhance the functionality of GameAdapt.KOM to make available adaptation mechanisms accessible to the GM. Section 6.3 explains how GameAdapt.KOM can be used to enable the GM to use adaptation mechanisms available in GameAdapt.KOM manually. This contribution could be classified in the field of Human-Computer-Interaction (HCI). Hence, it is not a core contribution of this thesis. Nevertheless, it is a necessary aspect that needs to be addressed in order to be able to apply the underlying methods and concepts for Game Mastering developed here.

In Section 6.4 the enhanced GameAdapt.KOM architecture is presented. The enhanced architecture includes the Game Master frontend which provides the GM with gathered and aggregated information about the game as well as available adaptation options.

7 PLAYER SIMULATION CONCEPT

»I had discovered that learning something, no matter how complex, wasn't hard when I had a reason to want to know it. «

— Homer Hickam

For a comprehensible evaluation of the developed concepts, a multitude of participants would be required. Moreover, it is necessary to evaluate depending on participants' player models, learner models, and interaction models. It is, however, hardly possible to compose groups of players with desired player, learner, and interaction models. Therefore, to be able to comprehensively evaluate GameAdapt.KOM, a concept was developed to simulate player/learner behavior with configurable player traits, knowledge, as well as social skills to define their behavior in the dimensions learning, gaming, and interaction. The player simulation developed in this thesis is designed with the goal of being able to simulate realistic behavior of players/learners. The chosen approach is to observe real players playing the game and to model goals and plans from the observed behavior for the simulated players. The simulated player agents then rate their available goals based on their player, learner, and interaction model, choose the best available plan for the goal and execute it. Validity of the behavior is evaluated by comparing the resulting behavior with the expected behavior considering the player, learner, and interaction model configuration. In Section 7.1, the agent-based player concept is described including the agent components and the agent's player, learner, and interaction model. The simulation process including the plan selection algorithm is explained in Section 7.2.

7.1 AGENT-BASED PLAYER

Collaborative multiplayer (Serious) games can be characterized as highly dynamic. Thus, player goals might constantly change. In Chapter 3, different player simulation models were discussed. In detail, four different agent-based models were presented: the simple reflex agent, the model-based agent, the goal-based agent, and the utility-based agent. The utility-based agent working towards a desirable world state is well fit for changing player goals. Therefore, players are modeled as a utility-based agent since this kind of agent can deal best with those circumstances.

7.1.1 *Basic Definitions*

Definition: Game Variable (repetition). *A game variable $v \in V$ is an elemental piece of information about the game. Game variables can change either through game mechanics and events or through player actions.*

Definition: Player Action (repetition). *A player action $a \in A$ is an isolated action in the game which can be performed by the player. The action has a well-defined effect on the game and/or the player(s).*

Examples are 'walk to', 'gather berry', or 'fell palm'. All available *Player Actions* are defined via the game interface including relevant parameters and effects.

Definition: Skill (repetition). *A skill* $\mathtt{skill}_x \in \Sigma$ *is a game relevant piece of knowledge (e.g., 'I can eat berries to increase my satiety') or motor skill (e.g. carrying palm). The function* $f : \Sigma \to [0;1]$ *assigns a value to a skill* $\mathtt{skill}_x$*, such that* $\mathtt{skill}_x = 0$ *means that the skill is not learned and* $\mathtt{skill}_x > \gamma$ *means the skill is learned, whereas* γ *is a predefined threshold (e.g.,* 0.75*).*

Skills can be learned (i.e., the skill value increases) by doing related actions or by gathering relevant information. Teamwork and communication between players is modeled using two special skills 'Teamwork' and 'Communication'. The 'Teamwork' value is multiplied by the respective skill related to a collaborative task. The 'Communication' value decides about with what probability a player passes on new knowledge to other players.

Definition: Player Goal. *A player goal* $g \in G$*, with* $g = (PI, K \subseteq \Sigma, GC' \subseteq GC, P' \subseteq P \setminus \{\emptyset\})$ *is an elementary objective which the player pursues. Player goals define what a player needs to do in order to successfully play the game. They consist of a player interest* PI*, a set of knowledge preconditions* $K \subseteq \Sigma$*, a goal condition* $GC' \subseteq GC$ *, and a set of plans* $P' \subseteq P$ *fulfilling the goal.*

Definition: Player Interest. *The player interest* PI*,* $PI^T = (pi_1, pi_2, ..., pi_{n-1}, pi_n)$*,* $pi_x \in [0;1]$ *is a vector defining how a goal matches to a player's player model shaping. The player interest is an n-dimensional vector which assigns a value of* $[0;1]$ *corresponding to each player model trait.*

Example: $PI = (0.0, 1.0, 0.5, 0.5, 0.0)$ (using a player model with the five traits: curious, ambitious, acting, interacting, moving) would mean that the *player goal* would be pursued strongly by players who have high ambition shaping, and moderately by players with a high acting and interacting shaping in their player model.

A *player interest* can be defined statically (like in the previous example) or dynamically. A dynamic *player interest* is changed by a multiplier depending on game variables. For example, the 'Increase Satiety'-goal has a $PI = (0.5, 0.5, 0.5, 0.5, 0.5)$ *Player Interest*. The *Player Interest* is changed with respect to the current satiety such that a lower satiety results in a higher rated goal for all player model traits.

Definition: Goal Condition. *A goal condition* $gc \in GC$ *is a game condition which needs to be fulfilled for the goal to be accomplishable. The information is gathered from the game itself (using game variables) and stored in the world state. The goal condition is formulated as a boolean expression using a set of game variables* $V' \subseteq V$*.*

Definition: Knowledge Precondition. *A knowledge precondition* $K \subseteq \Sigma$ *is a set of player (knowledge) skills which needs to be fulfilled for the goal to be accomplishable, i.e., the player 'knows' the required skills.*

Definition: Plan. *A plan* $p \in P, p = (A' \subseteq A, K \subseteq \Sigma, w)$ *is a tuple consisting of player actions* $A' \subseteq A$ *which are to be executed in a defined order, a set of knowledge preconditions* $K \subseteq \Sigma$*, and a weighting value* w *to prioritize the execution of plans when more than one plan for a goal is executable.*

7.1.2 *Agent Components*

The player agent architecture (see Figure 25) mainly consists of three modules: the perception module, the planning module, and the control module. On top of that, the player agent contains an AI player model, keeps an internal world state, and has a repository of plans and goals.

The underlying game uses the same model as in Chapters 4 and 5. In the core, the game state is influenced by game events and player actions. Available player actions, however, need to be known by the player agent.

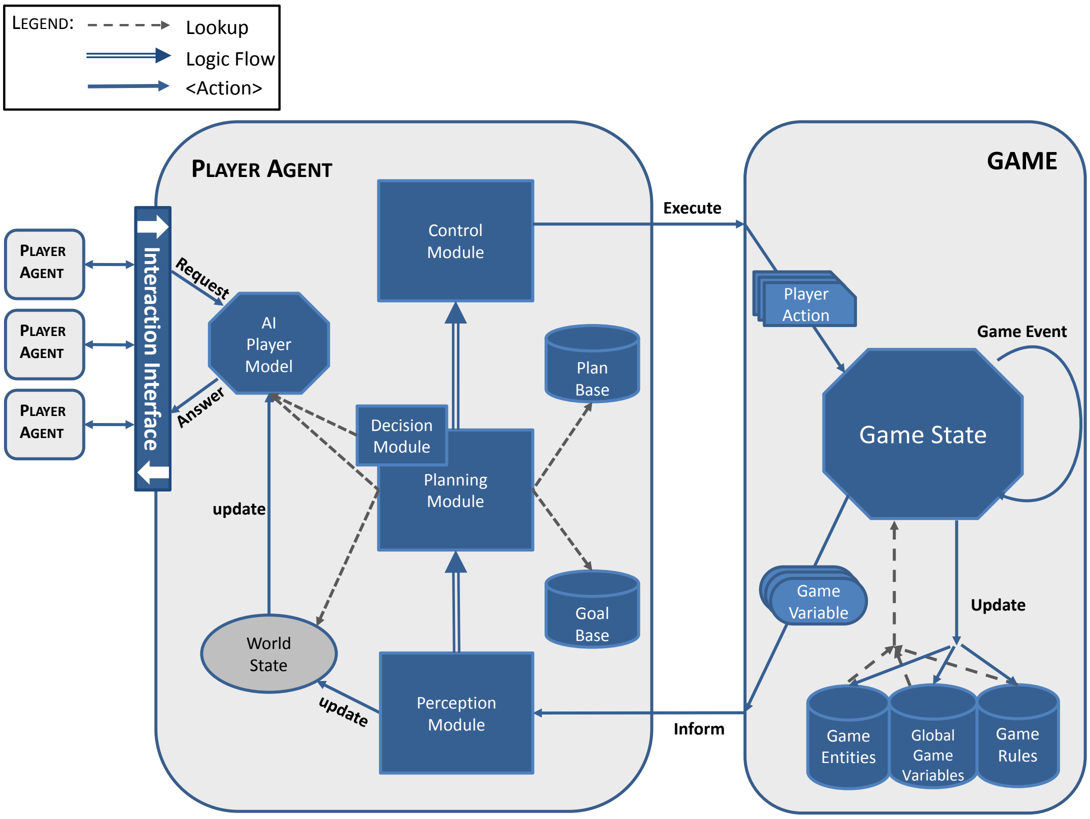

Figure 25: Player simulation architecture showing the player agent (left) and its interaction with the game (right).

Perception Module

Perceived information is processed in the *perception module*. The *perception module* is responsible for processing the information gathered from the game and for updating the state of the world (*world state*).

Planning Module

The *planning module* periodically evaluates the *simulated player model* and current information about the game world in order to decide which plan should be executed next. Therefore, a plan library is used which holds a set of pre-defined plans. In each plan evaluation cycle the planning module evaluates all plans for applicability and chooses those plans which can be applied for the current player goal.

Control Module

The *control module* decides which game actions the simulated player should execute based on the current plan and *simulated player model*. Thus, the control module is responsible for communication with the game as it sends messages to the game triggering player actions.

AI Player Model

The *AI player model* contains the current player model, learner model, and interaction model as defined in Chapter 4. It is updated when relevant game states change, like when the player receives new knowledge, etc.

World State

The world state is is the agent's local representation of relevant information about the game. This contains the set of variables which are input variables for goals and plans. Thus, the planning module makes use of the world state when deciding which plan to execute next.

7.1.3 *AI Player Model*

The simulated player model consists of three elements: The *player model*, the *learner model*, and the *interaction model*, representing the simulated player in terms of game-play preferences, knowledge, and collaboration/teamwork skills.

7.1.3.1 *Player Model*

The player model represents the simulated player's preferences in terms of play style. Depending on the underlying Serious Game, a set of traits is defined which represent possible player preferences. Those traits can represent global player preferences like 'action-oriented' or 'defensive'. Or those traits refer to more fine-grained traits like 'prefers to use ability x'.

The traits defined in the player model are identically equal to the traits of a player interest of a goal. Thus, it is possible to define a metric to calculate how attractive a goal is for a player depending on his/her player model.

7.1.3.2 *Learner Model*

The learner model defines which game-relevant skills a player has. Skills are modeled following the *Competency-based Knowledge Space Theory* [92] which uses Hasse diagrams to order skills hierarchically and to model prerequisites between skills as relations. Thus, the learner model is described by the partially ordered set of skills and dependencies between those skills. Each skill is initially set to a value in $[0, 1]$, with 0 meaning the player does not have the skill, and 1 meaning that the player has completely acquired the skill. Learning is modeled in form of modifications of a skill value. Whenever a player learns something, the related skill is increased about a specified amount related to what was learnt.

7.1.3.3 *Interaction Model*

The interaction model represents how well players can communicate with each other and to which extend they are able to perform in a team. A good communication means that players recognize relevant pieces of information and moreover, recognize that they should forward that information to one or more team members. Thus, information is modeled as a skill (in the learner model). Having the skill learned means knowing the information. Communication is modeled as a special skill defining to which extend (i.e., probability) the player forwards information once he/she receives them (i.e., learns the respective skill). Teamwork is also modeled as a special skill which is used as a multiplier for skills which require collaboration.

Interaction between players is modeled within plans. Whenever a goal requires collaboration, the plan contains an 'AskForHelp' action. The player will then ask a suitable fellow player to 'help'. That player will evaluate its current goal against the goal it is asked to help. Players with a higher teamwork skill are more willing to help than players with a low teamwork skill. On the other hand, players with a high skill in communication get a bonus when they ask for help. The asked player is more willing to help if the asking player has a high communication skill. This reflects the fact that a player with good communication skills is better able to explain the necessity of the requested action to its teammates.

A high communication skill moreover makes players more likely to share knowledge with fellow players. This is to reflect that players with good communication skills share critical knowledge about the game with their teammates once they receive it.

The asked player then answers with either 'yes', 'no', or 'why'. If the asked player evaluates the asked goal higher than its current goal, the agent answers 'yes'. If the asked player evaluates the asked goal lower than its current goal, the answer depends on how much lower it is evaluated. If it is evaluated lower than $\delta \cdot e(g_{now})$, whereas δ is a threshold and $e(g_{now})$ is the rating of the current goal g_{now}, the agent answers 'no', otherwise the agent answers 'why'.

Figure 26 shows a sequence diagram of the asking process of a player agent (with the answer being either 'yes' or 'no'). Figure 27 shows the process from the asked player agent's perspective (with the answer being either 'yes' or 'no').

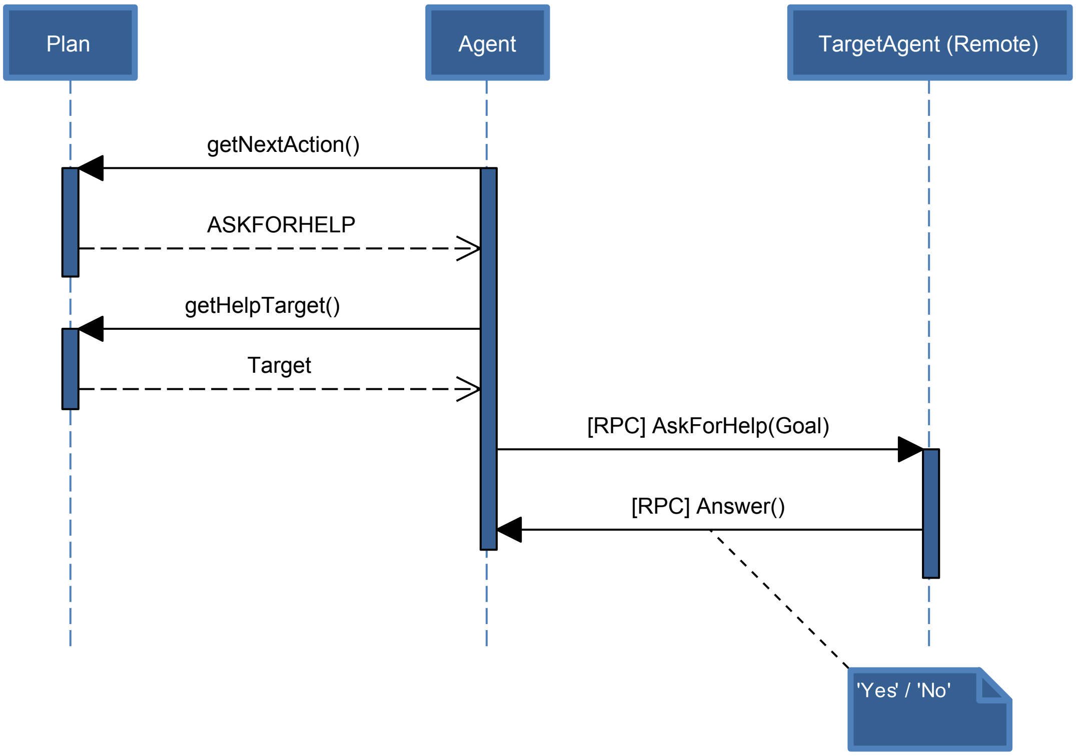

Figure 26: Sequence diagram: Player Agent asking for help (possible answers: 'yes' or 'no') from the asking agent's perspective.

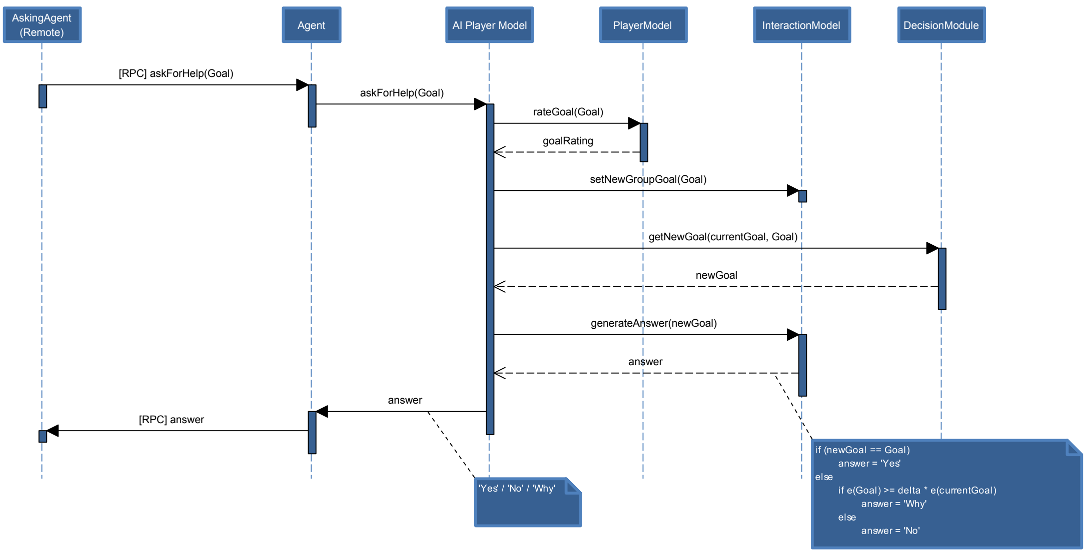

Figure 27: Sequence diagram: Player Agent asked for help (possible answers 'yes' or 'no') from the asked agent's perspective.

If the asked player responds with 'why', the asking player sends the current goal's higher goal. This is the goal which is ultimately to be achieved by achieving the current goal. The asked player then compares the higher goal with the current goal and decides whether to accept the request ('yes') or not ('no').

Figure 28 shows a sequence diagram of the asking process of a player agent (with the answer being 'why') and the asking players persuasion attempt. Figure 29 shows the persuasion process from the asked player agent's perspective.

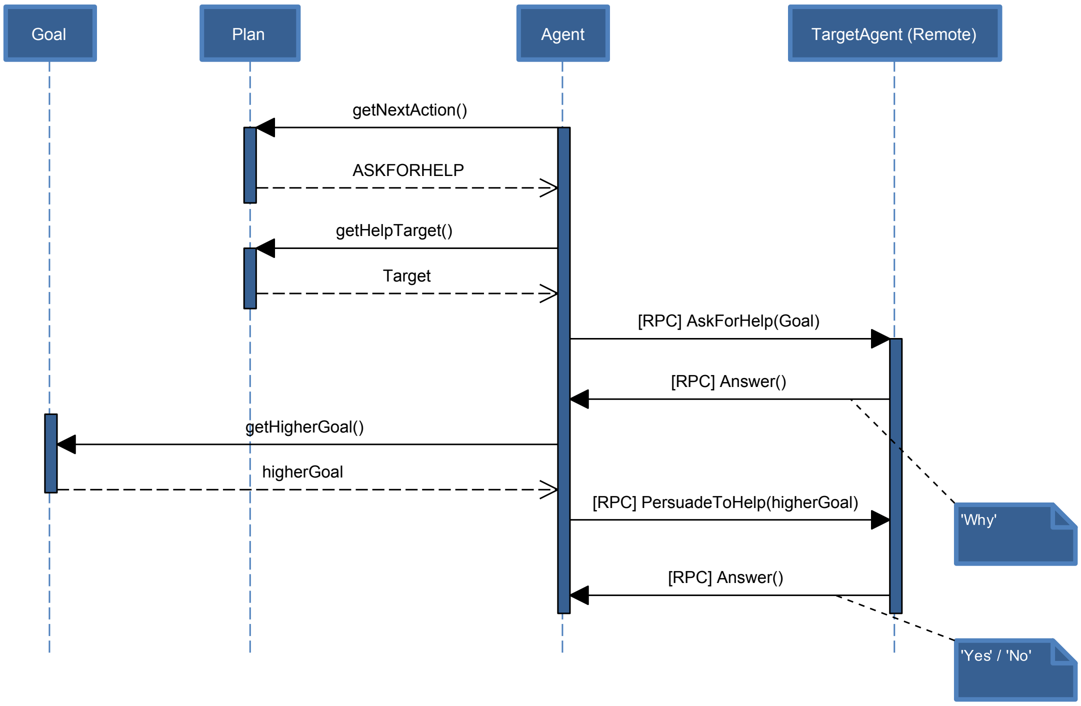

Figure 28: Sequence diagram: Player Agent asking for help (possible answer: 'why') from the asking agent's perspective.

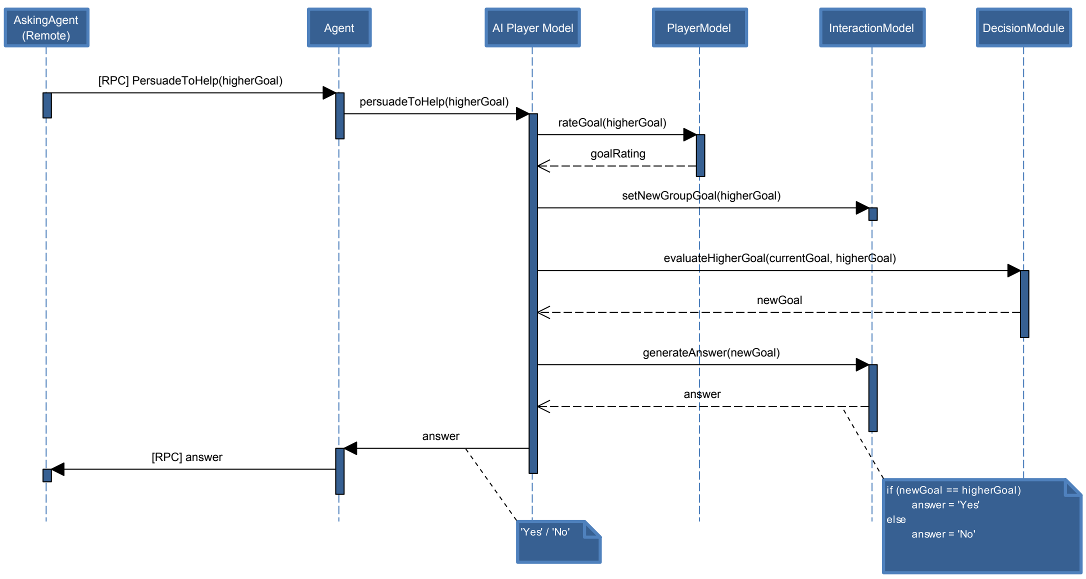

Figure 29: Sequence diagram: Player Agent asked for help (previous answer: 'why') from the asked agent's perspective.

7.2 SIMULATION PROCESS

In each simulation cycle, each simulated agent perceives information about the game (world), plans, and, if applicable, sends control information to the game.

For each simulated player, the simulation determines the next goal once the current goal is accomplished or when it cannot be accomplished any more. The next goal is determined by comparing the player model with the player interest for each goal with regard to goal conditions and knowledge preconditions. The most fitting goal is chosen according to a metric to define the appropriateness of a goal. First all goals are filtered, checking if their *goal conditions* and *knowledge precondition* are accomplishable. Each accomplishable goal's *player interest* vector is then compared to the player's player model vector using the scalar product. Thus, the goal with the *player interest* which is most similar to the current player model shaping will be chosen. The goal with the highest value is chosen and gets processed by executing one of its available plans. Available plans are filtered depending on their preconditions. The valid plan (i.e., all preconditions fulfilled) with the highest weighting w gets executed. Executing a plan means that the plan's actions are executed in the defined order. If no valid plan for the highest rated goal can be executed, the next best goal is selected and checked if one of its plans can be executed. This is repeated until either a plan is executed or no more goals are available. If no goal can be achieved, the simulated player is idle. Idle players periodically check for available goals.

7.2.1 *Agent Simulation Algorithm*

Listing 2 describes the algorithm which is executed in each simulation cycle to determine the simulated player's behavior.

Listing 2 Agent Simulation

```
Perceive()                    ▷ Receive game updates, i.e. update the set of variables V
if currentPlan = null then
    Find_Suitable_Plan()
else if currentAction! = null then
    if currentPlan.hasNextAction then
        Start_Next_Action()           ▷ Start execution of next action of current plan
    else
        Find_Suitable_Plan()
    end if
else
    Execute_Current_Action()          ▷ Go on with execution of current action
end if
```

7.2.2 *Plan Selection Algorithm*

Listing 3 shows the algorithm to find a suitable plan to be executed. It uses the function $\mathrm{RateGoal}(g)$, which rates how important the goal is for the simulated player based on its player model PM. $\mathrm{RateGoal}(g)$ rates the goal g using the function r, which rates the goal g's *player interest* PI_g in relation to the player agent's player model PM (see Equation 64):

$$\mathrm{RateGoal}(g) = r(PI_g, PM) = \frac{\sum_{i=0}^{|T|-1} 1 - |t_i - pi_i|}{|T|} \tag{64}$$

Function r returns 1, if the PlayerInterest is identical to the current player model. Differences in each trait are subtracted from 1, added up and divided by the number of traits. Thus, differences add negatively to r.

Listing 3 Find Suitable Plan

Require: G, $G_{possible}$, $P_{possible}$
  for all Goal $g \in G$ **do**
    if all knowledge preconditions K_g of g met **then**
      if all goal conditions GC_g of g met **then**
        RateGoal(g) ▷ rate the importance of goal g
        $G_{possible} = G_{possible} \cup g$ ▷ considering the current world state
      end if
    end if
  end for

  plan selection:
  if $G_{possible} = \emptyset$ **then**
    return 'no goal available'
  else
    Let g_0 be $\max(G_{possible})$
    for all Plan p in P_{g_0} **do**
      if all knowledge preconditions K_p of p met **then**
        $P_{possible} = P_{possible} \cup p$
      end if
    end for
    if $P_{possible} = \emptyset$ **then**
      $G_{possible} = G_{possible} \setminus g_0$
      goto plan selection
    end if
    return $\max(P_{possible})$
  end if

7.2.3 *Plan Selection Algorithm Complexity*

As the player simulation is potentially active in every frame during the game's run-time, in the following section a computing time estimation for the selection of the most suitable plan will be shown. There are basically four steps involved in finding the next suitable plan:

1. Rate all goals
2. Find the highest rated goal
3. Filter all plans of the highest rated goal

4. Find the highest rated plan

Thus, the complexity can be denoted as:
$\mathcal{O}(FindPlan) = \mathcal{O}(RateGoals) + \mathcal{O}(SortGoals) + \mathcal{O}(FilterPlans) + \mathcal{O}(FindBestPlan)$
To rate a goal, its PI is compared to the player agent's Player Model. Hence, $|T|$ traits are compared, whereas T is the set of player traits. This is performed for all goals in G.

$$\mathcal{O}(RateGoals) = \mathcal{O}(|G| \cdot |T|) \tag{65}$$

Finding the maximum from a list can be done in $\mathcal{O}(n)$, whereas n is the number of items. Hence,

$$\mathcal{O}(FindMaxGoal) = \mathcal{O}(|G|) \tag{66}$$

whereas G is the set of goals.

Regarding filtering plans, for each plan the plan's knowledge preconditions need to be checked. In the worst case, each plan could have all skills in the list of its preconditions:

$$\mathcal{O}(FilterPlans) = \mathcal{O}(|P| \cdot |\Sigma|) \tag{67}$$

Again, finding the maximum from a list can be done in $\mathcal{O}(n)$, whereas n is the number of items. In the worst case, all plans in P might need to be considered,

$$\mathcal{O}(FindBestPlan) = \mathcal{O}(|P|) \tag{68}$$

$$\begin{aligned} \mathcal{O}(FindPlan) &= \mathcal{O}(RateGoals) + \mathcal{O}(FindMaxGoal) + \mathcal{O}(FilterPlans) \\ &\quad + \mathcal{O}(FindBestPlan) \\ &= \mathcal{O}(|G| \cdot |T|) + \mathcal{O}(|G|) + \mathcal{O}(|P| \cdot |\Sigma|) + \mathcal{O}(|P|) \end{aligned} \tag{69}$$

The number of plans per goal is limited to $|P| \leqslant 5$ in reality. Hence, the estimation can be simplified:

$$\begin{aligned} \mathcal{O}(FindPlan) &= \mathcal{O}(RateGoals) + \mathcal{O}(FindMaxGoal) + \mathcal{O}(FilterPlans) \\ &\quad + \mathcal{O}(FindBestPlan) \\ &= \mathcal{O}(|G| \cdot |\mathcal{T}|) + \mathcal{O}(|G|)\mathcal{O}(|\Sigma|) \\ &= \mathcal{O}(|G| \cdot |\mathcal{T}| + 1) + \mathcal{O}(|\Sigma|) \\ &= \mathcal{O}(|G| \cdot |\mathcal{T}|) + \mathcal{O}(|\Sigma|) \end{aligned} \tag{70}$$

Thus, the goal selection runs in linear complexity regarding the number of goals, the number of traits in the player model, and the number of skills in the learner model.

7.3 CHAPTER SUMMARY

Motivated by the necessity of a multitude of players in order to be able to profoundly evaluate the methods and concepts developed, the need for players/learners simulation was identified. This is due to the need for a high number of participants, but even more due to the fact that it is hardly possible to compose groups of players representing the desired player/learner/interaction models. Therefore, to be able to comprehensively evaluate GameAdapt.KOM, a concept was developed to simulate player/learner behavior with configurable player traits, knowledge, and social skills. Players are simulated as utility-based agents.

In Section 7.1, the agent model is presented starting with a definition of core concepts. The terms *player action*, *skill*, *player goal*, *player interest*, *goal condition*, *knowledge precondition*, and *plan*, are introduced as basic definitions. Subsequently, the components of the player AI are explained: The player agent AI contains a perception module, a planning module, and a control module. It further contains a world state, an AI player model (containing a player model, a learner model, and an interaction model), and a set of plans and a set of goals. The perception module processes information about the game world and calculates a world state. When the world state is updated, the agent's player, learner, and interaction models are updated accordingly, if necessary. The planning module uses the world state and the AI player model to evaluate the set of plans and to decide which plan should be executed next. Therefore, it makes use of the set of goals. The control module decides which game actions should be performed next based on the current plan.

The simulation execution algorithm is described in Section 7.2. Each player agent determines the next goal once the current goal is accomplished or once it cannot be accomplished any more. Therefore, the AI player model is compared to the player interest for each goal with regard to goal conditions and knowledge preconditions. The most fitting goal is chosen according to a metric to define the appropriateness of a goal. The best-fitting goal is selected and processed by executing one of its available plans. Available plans are filtered depending on their preconditions. The valid plan with the highest weighting is executed: i.e., the plan's actions are executed in the defined order.

With this player simulation concept, it is now possible to simulate the behavior of players in a collaborative multiplayer Serious Game. This allows GameAdapt.KOM to be evaluated both with automatic adaptation and with Game Master functionality using an unlimited set of virtual players with configurable characteristics. The virtual player agents can be freely configured in terms of their gaming style and preferences (player model), their knowledge state and learning style (learner model), and their interaction skills (interaction model). Based on this, players will behave accordingly, thus being able to solve game tasks depending on their player, learner, and interaction models. Hence, challenge (flow model) can indirectly be configured, too.

The conceptualization of the simulated player model, as well as the agent simulation including goal and plan selection algorithms, is the main contribution of this chapter.

8 COLLABORATIVE MULTIPLAYER SERIOUS GAME PROTOTYPE

»Tell me and I forget, teach me and I may remember, involve me and I learn.«

— Benjamin Franklin

In this chapter, the development of the collaborative multiplayer Serious Game *Escape From Wilson Island* (EFWI) is covered, which is used as a simulation environment and testbed for the adaptation and Game Mastering concepts developed within this thesis. The Serious Game purpose and the resulting game design is explained (Section 8.1) based on guidelines on collaborative learning and gaming in the literature. Section 8.2 covers the collaborative gameplay concepts on which the game design is build upon. Section 8.3 explains how the concept of adaptation is implemented and Section 8.4 describes how the player, learner, and interaction models in EFWI are assessed. The integration of GameAdapt.KOM as adaptation mechanism is explained in Section 8.5. In Section 8.6, it is explained how the developed Game Mastering concepts are integrated into EFWI, focusing on the explanation of the required GUI. Section 8.7 covers the implementation of the player simulation in form of an extension of EFWI.

8.1 ESCAPE FROM WILSON ISLAND GAME DESIGN

To the best of the author's knowledge, there are currently only few collaborative multiplayer Serious Games available. Of them, only very few are designed for including a Game Master. One example of such a game is *ViPol*, a team operation training game for police and action force. However, *ViPol* is not freely available and its limitation to the police and action force sector complicates evaluations as those groups are rarely available. As there are no known adaptive *multiplayer* Serious Games available, it was decided to design and develop an own solution of a collaborative multiplayer Serious Game as a proof-of-concept: *Escape From Wilson Island*.

The characterizing serious goal is the improvement of of soft skills, more closely collaboration, teamwork, coordination, and intra-group communication. Hence, the developed collaborative multiplayer Serious Game was designed with the following goals in mind:

- Create a game with serious learning content
- Design for a small group of players
- Promote collaborative behavior

As genre, 1st/3rd person action adventure was chosen. This made is possible to represent players with avatars and to create a finite level on which players can navigate their avatar. Hence, the setting was chosen to be settled on an island. Thus, the game world could naturally be limited through the use of water as a natural border.

To create the game's goal, the narrative background was designed to create a common dilemma from which the players need to escape. Therefore, the Robinson Crusoenade was chosen as narrative background: Players strand on a deserted island from which they want to escape. A chain of tasks and obstacles needs to be overcome for the players to achieve their goal.

An NPC was included as a mentor and an in-game helping system. The NPC Hank is an eremite who is living on the island and can provide the players with valuable information about the island and how to survive.

It was decided to limit the number of players to exactly four players as this is a common number for Coop-games like e.g., *Left4Dead*, *Evolve*, or *Forced*. Morover, a fixed number of players limits design decisions for the collaborative tasks. Subsequently, all collaborative tasks are designed for a maximum of four players.

The game was designed in a way such that it does not require any additional instructions, like a tutorial. This, however, implied the necessity of an intuitive GUI and game controls. Moreover, it needs to be clear what players can or need to do. Therefore, the players can ask the NPC about all game relevant things.

The game was developed using the Unity3d game engine. The terrain was created using the integrated terrain tools. Models were created using Blender and integrated in the .fbx-format. Textures for the integrated models were also imported from Blender or created using Photoshop.

Game logic was completely implemented in C#, which is one of the supported scripting languages: Javascript, C#, and Boo. It was decided to use C# for all scripts, as it was considered the more 'clean' scripting language due to its explicit variable typing. As development environment Microsoft Visual Studio was used.

8.2 COLLABORATIVE GAMEPLAY CONCEPTS

Preparing a game-based collaborative learning setting which takes place in a game-based learning environment, yields additional requirements. As shown in Chapter 3, there are several requirements for collaborative learning to take place (e.g., requirements of cooperative working, see [76]). Furthermore, in a Serious Game, requirements regarding game design and learning content integration have to be met. In this approach, we integrated the requirements for cooperative working into the game design to create a Serious Game for collaborative learning based on one of the most popular Serious Gaming genres for learning: 3D virtual worlds. The goal is to make the collaborative tasks require teamwork, coordination, and intra-group communication skills for them to be solved by the team as well as require good overall task sharing to beat the game.

Resulting from the literature review, we developed the following concept ideas. They are designed in a way such that they match the necessary elements for cooperative working of Johnson & Johnson [76]: (*Positive interdependence, individual accountability, face-to-face promotive interaction, social skills*, and *group processing)*. Furthermore, they take into account the lessons and pitfalls of Zagal et al. [183] (see Section 3.1) and the design guidelines according to Padilla-Zea et al. [186].

8.2.1 *Collaborative Gameplay Concepts related to Padilla-Zea et al.*

Common Goal/Success
The goal of the game should be designed in a way such that players win or lose together.

Heterogeneous resources
Each player should have one unique tool or ability enabling him/her to perform unique tasks in the game which other players cannot perform, e.g., only the player with the axe can fell palms. Hence, only this player can cut firewood.

Refillable personal resources)
In order to create a certain tension, there should be certain refillable resources (e.g., a health or hunger value) which slowly deplete automatically or when players act recklessly. Furthermore, they should be influence-able in a way such that players can help each other (e.g., food could be gathered by one player and then be given to another player to prevent him/her from starving).

Collectable and tradeable resources
There should be resources available in the game world which are necessary for the players to win the game. These resources should be tradeable between players in order to create space for decisions to negotiate or collaborate (e.g., giving a resource to another player for the common good of the team or trading resources between players).

Collaborative tasks
If all tasks could be solved by one player, there would be no need to collaborate. So there should be tasks which are solvable only if players work together. Those tasks may include the heterogeneous resources described above to create a need for certain players to participate in team tasks. This may cause a need for discussion among players when the group depends on one individual.

Communication
It has been shown that communication is vital for collaborative learning. So the game should provide a way for players to communicate (e.g., chat system, voice communication). While voice communication might be easier for most players, a text-based chat system might be easier to evaluate. Also a third party tool for communication like Skype, TeamSpeak or Mumble could be used.

Ingame help system
The game should provide help to the players when they are stuck. The easiest way is a pop-up when players fail a task or it takes them too long to solve it. Furthermore, the help system should be trigger-able by the players. A more sophisticated but also more immersive way is to include help in the game itself, e.g., by having in-game characters (NPCs) providing help when needed.

Scoreboard

A scoreboard should show both individual efforts and team efforts at the end of the game. This may help players judge the overall success (e.g., by comparison with other teams or previous attempts) and each player's contribution to the team performance. The individual score may function as a motivator for selfish actions which helps to make collaboration not self-evident.

Trading system

Players should be able to trade items among each other. This creates space for decisions for or against collaboration.

8.2.2 *Collaborative Gameplay Concepts related to Zagal et al.*

As a next step, the game concept is discussed in relation to the lessons and pitfalls according to Zagal et al. (see Section 3.1):

Regarding lesson 1 (tension between individual and team utility)

By having an individual score board for each player, a competitive element is created. Individual scores can sometimes be achieved by helping the group (e.g., when participating in solving a task together), or they can be selfish (e.g., when gathering resources).

Regarding lesson 2 (individual decision-making)

By the nature of the game (3D third person, open environment), each player may move and act freely. No player is forced to perform any action, although some actions are not possible without other players' consensus.

Regarding lesson 3 (being able to link payoffs to own decisions)

The results of decisions are always visible to the players as they are immediate. A player may e.g., decide to help solving a task or to gather resources.

Regarding lesson 4 (different abilities and responsabilities)

By providing players with heterogeneous resources (tools), each player has a different ability and responsibility.

Regarding pitfall 1 (sufficient rationale for collaboration)

The nature of a 3D third person game makes it very difficult for one person to fell decisions for all players, however leader roles will certainly be possible and relevant.

Regarding pitfall 2 (players need to care about the outcome)

As all players either win or lose together, each player should care about the outcome assumed he/she has the proper motivation to play at all. Such a motivation is provided by the narrative background story.

Regarding pitfall 3 (replayability)

Although the core game itself will not change, when played repeatedly, the free game world and the free sequence of available actions can create completely different progressions of the game in different runs. Also, playing again with different players will be a completely different experience for a player. Furthermore, the difficulty of the game can be influenced by a teacher/trainer both before and during run-time, so that more experienced players will still find the game challenging.

8.2.3 *Collaborative Gameplay Concepts related to Johnson & Johnson*

Next, the game concept is discussed in relation to the cooperative working requirements by Johnson & Johnson:

Positive interdependence
As many tasks are only solvable if players work as a team, it is assumed players will realize quickly that they cannot succeed alone.

Individual accountability
By introducing an individual scoreboard, the game can assess each players personal performance. As the scoreboard is visible to the whole group, the results are given back to both the group and the individual.

Face-to-Face Promotive Interaction
Although the game itself does not encourage promoting behavior like encouraging or praising fellow players, the game enables players to do so. Players can help, which improves chance for success. Chat allows players to praise or encourage other players. Furthermore, players can help other players through their actions. A player who decides to help his/her fellow players, will significantly improve his/her chances of success.

Social Skills
The game provides lots of opportunities for practicing social skills both in speech (chat) or behavior (gameplay).

Group Processing
The game provides the players with the possibilities to discuss their progress and relationships (chat) and to reflect on them (scoreboard).

8.2.4 *Resulting Game Design Decisions*

In the following list, it is shown how the design guidelines are applied.

Common Goal/Success
Players can only escape together, not alone. It is not possible to evade the island alone. An outro will be played at the end of the game as a reward if the game was finished successfully.

Heterogeneous resources
Each player has one unique tool (axe, map, whistle, watch) enabling him/her to perform unique tasks in the game which other players cannot perform The axe is

required to fell palms. The map is required to be able to steer the raft through currents in the water. The whistle is required to attract the herons on the island to hunt them. The watch is used to trigger sleeping and to configure the hours slept per night which impacts the amount of hours available per day at the expense of regeneration.

Refillable personal resources

There are three personal resources: satiety, health, and fitness. The players' avatars need to eat from time to time in order not to starve. Furthermore, they have a fitness value which depletes through walking or running and regenerates when sleeping. A lack of fitness slows players down. The health value is negatively influenced when players are starving or when they are drowning. It is regenerated when eating or sleeping.

Collectable and tradeable resources

Three collectable and tradeable resources are integrated into gameplay: Wood, berries, and meat. There are two ways to get food: A player can gather berries from a bush which restores a small amount of satiety, or the players can hunt a heron, which will give them some pieces of meat that can be cooked. Each piece of cooked meat restores a large amount of satiety, making heron meat a lot more valuable resource than berries. Wood can be gathered from felled palms.

Collaborative tasks

1. Building the log hut: To build the log hut, first palms need to be felled. Only the player with the axe can do this. This implements the heterogeneous resources. Palms require three players to be carried, one at each end and one in the middle of the palm. Players need to synchronize their movements to not drop the palm. Figure 30 shows three players carrying a palm in EFWI.
2. Hunting herons: Herons can only be hunted in team, as they have to be surrounded, which needs at least 3 players but is easier if more players take part in it. Surrounding the heron such that it does not have any escape route requires players to coordinate their movement.
3. Steering the raft: The raft can only be steered when all players are participating, as each player can only sit in one corner, steering the raft towards his/her corner when paddling. So the players have to coordinate their actions when steering the raft. On top of this, only one player is able to see the currents in the sea. Thus, communication is vital for this task since this player needs to tell the other players where to steer. Figure 31 shows the four players steering the raft over the sea.

Communication

Players are able to communicate with each other via an integrated chat. It is possible to chat with only one other player, with a set of other players, or with all other players. The chat window is always visible in the lower left corner. Of course, it is also possible to allow players in the same room to talk with each other or to use a third party tool for communication like Skype, TeamSpeak, or Mumble.

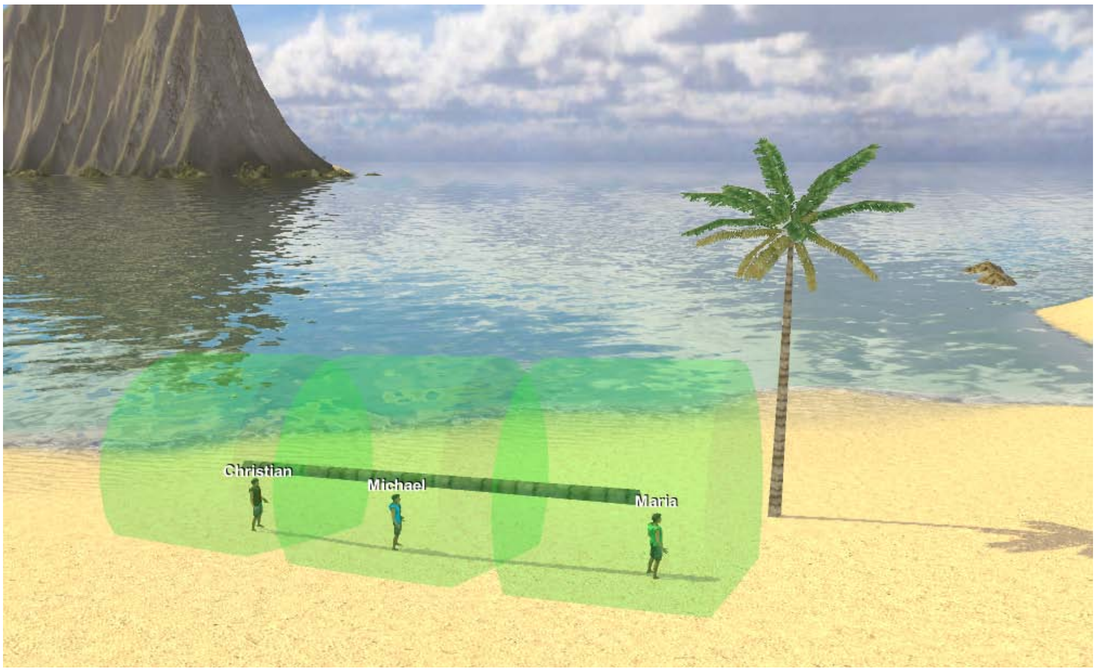

Figure 30: Screenshot of EFWI - three players are carrying a palm. The green areas around the players indicate the tolerance radius which they may not leave.

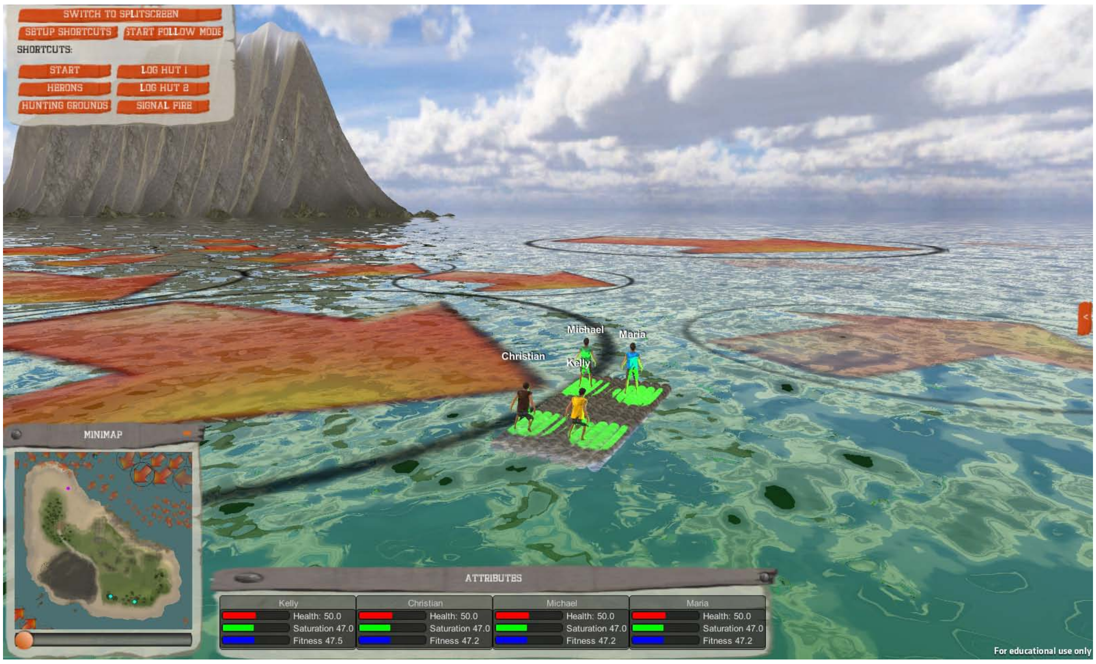

Figure 31: Screenshot of EFWI - four players steering the raft. The Game Master can see the currents in the sea (arrows).

Ingame help system

An NPC eremite named Hank which is living on the island was integrated. The NPC's task is to guide the players through the game, giving them hints when necessary and answering some game related questions. The NPC communicates with the players via a predefined structured chat system or can be controlled by the Game

Master. The GM then is able to talk to the players via a chat system or the instructor can design structured dialogs ad-hoc at run-time.

Scoreboard

At the end of the game, each player will have an individual score that is visible to the whole group. The score depends on the number of (potentially) meaningful actions performed during the game like gathering berries, carrying wood, building the raft, helping to catch a heron, etc. This is intended to create a tension between individual performance and team performance. Moreover, players can always decide to keep the valuable food for themselves, rather than to give it to another player.

Trading system

Every player has a personal inventory where he/she can place items gathered throughout the game. Those items can be given to other players by placing them in a community box accessible by every player. Access to the chest is trust based, there is no control mechanism to prevent someone from taking something out of the chest.

8.3 ADAPTATION IN EFWI

In this section it is shown how the adaptation concepts developed in Chapter 5 are implemented in EFWI. This contains an explanation of the EFWI player, learner, and interaction model as well as how game situations in EFWI are recognized.

8.3.1 *EFWI Player Model*

The player model representing EFWI player traits is designed in a way such that it represents all relevant play styles and player preferences.

The player model is strongly related to Bartle's player model, as the genre is similar. However, some of the traits need to be interpreted slightly different as the genre of a 1st/3rd person action adventure is similar to a 1st/3rd person role-playing game (for which Bartle's player model originally was designed), but there are some differences. Many game elements are similar to typical role-playing games, like playing a character which is represented by an avatar in a virtual world. In both genres, there are usually NPCs and other players to interact with. However, there is no skill system which is fundamental to most role-playing games. In the case of EFWI, there is also no combat or fighting system which is elemental to many role-playing games. Also, the narrative development throughout the game is rather limited in EFWI. This is a core aspect in RPGs.

It was decided not to take the four traits of Bartle's player model as it has been shown recently that players usually can not be sorted into one category of Bartle's model. It is generally assumed that the player types (traits) might be overlapping or correlating with each other [37]. Thus, classifying players along the axes acting $\longleftrightarrow$ interacting and player $\longleftrightarrow$ world might not result in independent player types. In contrast to that, for the player model used here, traits on axes which are orthogonal to each other are necessary, such that having a high value in one trait is not a contradiction to having a high value in another trait. The chosen traits are the five axes as shown in Table 20.

Trait			Analogy to Bartle
Curious	⟷	Anxious	Explorer (1)
Ambitious	⟷	Phlegmatic	Achiever
Acting (= interacting with world)	⟷	Passive	Killer, Achiever
Interacting (with players)	⟷	Solitary	Socializer
Moving	⟷	Stationary	Explorer (2)

Table 20: EFWI player traits.

8.3.2 *EFWI Challenge Model*

The challenge model in EFWI is connected tightly to the tasks. For each task defined in EFWI, there is a value in the challenge vector. EFWI tasks are described below (see Section 8.3.4). The following list describes the challenge vector which is defined in EFWI:

Challenge	Explanation
Keep_Satiety_High	refers to the player parameter 'satiety'
Keep_Health_High	refers to the player parameter 'health'
Keep_Fitness_High	refers to the player parameter 'fitness'
Carry_Palm	refers to the ability to carry a palm
Build_LogHut	refers to the ability and time needed to build the log hut
Build_Raft	refers to the ability and time needed to build the raft
Hunt_Heron	refers to the ability and time needed to hunt the heron
Get_Seamap	refers to the ability and time needed to receive the sea-map
Fill_Bottle	refers to the ability and time needed to find and fill the bottle
Escape_Fast	refers to the time needed to successfully complete the game

Table 21: EFWI challenges.

This list covers all goals and tasks in EFWI. Basically, players should keep their vital values (health, satiety, and fitness) high. Thus, each of these goals is formulated as a task with a challenge value in the challenge vector. The speed of achieving the other goals (building the log hut, building the raft, hunting the first heron, acquiring the sea-map, and finding the bottle) is a good indicator for player performance and, subsequently, for the challenge. If they can achieve those goals too fast, the game might be too easy and vice versa. The same goes for the overall game progress, expressed through the task *Escape_Fast*. An important aspect here is the reference point which indicates an optimal challenge. A subject matter expert (i.e., the instructor) needs to define the reference challenge based on domain knowledge.

8.3.3 *EFWI Learner Model - Skill Graph*

Generally, there are two types of skills modeled here.

1. Knowledge skills: Skills which refer to the knowledge about tasks
2. Mechanical skills: Skills which refer to how a good a player's mechanics are concerning a task

Figure 32 contains the skill graph which is used in EFWI. On the lowest layer, there are the skills *KnowledgeRaft*, *KnowledgeWilson*, *KnowledgeHunt*, *KnowledgeWood*, *KnowledgeBerries*, *KnowledgeGeysir*, and *KnowledgeBottle*. Those refer to the basic knowledge skills. Players need to know that they have to build a raft, that they need to find the volleyball 'Wilson' to get a rope for the raft, that they can hunt herons, that they can fell palms (and no other trees), that they can gather berries from certain bushes, that they need to find a geyser for special gas for the signal fire, and that they need to find a bottle to fill the gas into. When a player knows that he/she can gather berries and how to do that, it should be taught how to eat. Only when a player knows about the fact that palms can be carried, he/she should learn about carrying palms. When a player knows how to carry palms, it is reasonable to teach him/her how to build the log hut and raft. For the raft, however, the player also needs to know about the rope which he/she will get from the NPC for giving him his volleyball 'Wilson'. Moreover, the player needs to know that he/she needs a raft to reach the second island. Once the player knows how to build the log hut, he/she should be taught how to sleep and why this is useful (in terms of gameplay). Similar, it should be taught how to build a fire and what it is good for. If the players know that they can get food by hunting the heron, they need to know how to hunt. If players know how to hunt and how to build a fire, they should learn about how to cook heron meat. Once players know that there is a bottle to find, they should be taught how to find it (i.e., by scanning the beach). Once players know about how to find the bottle and how to find the geyser, they should learn how to fill the bottle (as there is a special procedure required). The skills directly refer to one or more challenge as shown in Table 22.

Teamwork and *Communication* are listed as two special skills here representing the interaction skills. They are not directly related to the learning skills, but are listed here for the sake of completeness.

8.3.4 *EFWI Interaction Model*

EFWI implements the interaction model as described in Section 4.3. Thus, the interaction skills used are *Communication* and *Teamwork*.

8.3.5 *Game Situation Recognition*

Following, the game interface for EFWI will be described. First, EFWI game variables and parameters, actions, and events are listed.

The Tables 31, 32, and 34 in Appendix A show the game variables, player parameters, game actions, and events provided bei EFWI. Those tables are a consolidation of the EFWI game element objects. The Tables 35, 36, 37, and 43 in Appendix A show

CHALLENGE	SKILLS
Keep_Satiety_High	KnowledgeBerries, KnowledgeEat KnowledgeHunt, HuntHeron, CookHeron
Keep_Health_High	KnowledgeBerries, KnowledgeEat KnowledgeHunt, HuntHeron, CookHeron KnowledgeWood, CarryPalm, BuildHut, SleepInHut, BuildFire
Keep_Fitness_High	KnowledgeWood, CarryPalm, BuildHut, SleepInHut, BuildFire
Carry_Palm	KnowledgeWood, CarryPalm
Build_LogHut	KnowledgeWood, CarryPalm, BuildHut
Build_Raft	KnowledgeRaft, KnowledgeWilson, BuildRaft KnowledgeWood, CarryPalm
Hunt_Heron	KnowledgeHunt, HuntHeron
Get_Seamap	KnowledgeRaft
Fill_Bottle	KnowledgeBottle, FindBottle, KnowledgeGeysir, FillBottle
Escape_Fast	all skills

Table 22: EFWI skill - challenge interdependencies.

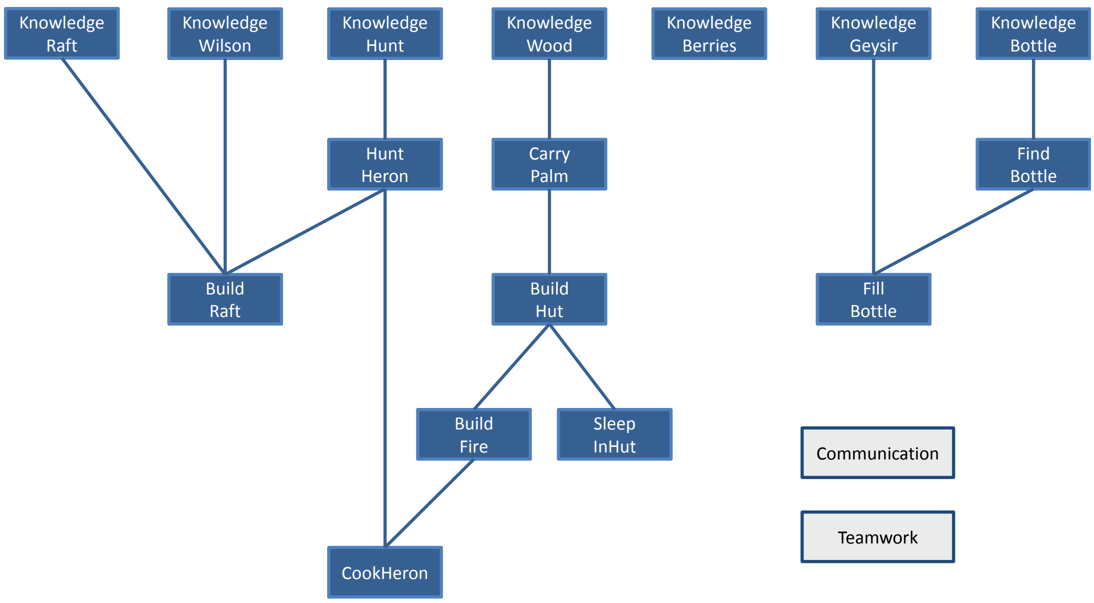

Figure 32: EFWI skill graph.

the tasks, regions, situations, and situation objects defined in GameAdapt.KOM for EFWI based on the information provided by EFWI.

8.3.6 *EFWI Adaptation Objects*

A complete list of adaptation objects and their respective adaptations used in EFWI is presented in Section A.6 in Appendix A (see Tables 46, 47, 48, 49, 50, 51, 52, and 53). The adaptations use the EFWI adaptation variables shown in Table 33. The list contains adaptations which have been defined specifically for EFWI with the goal of having meaningful methods to adapt the game's difficulty and to react to possible problems by players. Based on that, adaptation objects were defined (see Tables 54, 55, and 56 in Appendix A).

8.4 GROUP MODEL ASSESSMENT IN EFWI

To assess the player model, learner model, and interaction model of a player in EFWI, the following mechanisms are used.

8.4.1 *Player Model Assessment*

The player model in EFWI is directly extracted from player behavior, i.e., from a player's actions. Player models are initialized with the following neutral trait vector: $(0.5, 0.5, 0.5, 0.5, 0.5)$. Player actions are assessed using situation objects (see Section 5.5). The situation object's effect on the player model is applied whenever a situation object's situation is present.

The effect can be a modifier which is applied to the player's player model whenever the situation is considered true (trimmed between $[0, 1]$). This is used for discrete actions (e.g., gather berries). For continuous situations like walking or searching, the total modifier is calculated based on the relative time the player spent doing that action. For example, players are assumed to be walking about 40% of the game time. Hence, a player moving 40% of the current game time on average should have a (Moving $\longleftrightarrow$ Stationary) value of 0.5 which reflects the average action rate φ for that action. A value greater than 0.5 indicates that the player walks more than average, a value lower than 0.5 indicates that the player walks less than average. In general, a continuous action's overall modifier s^{mod} to a trait is calculated using the following formula:

$$s^{mod} = \frac{\frac{t_{action}}{t_{total}}}{2 \cdot \varphi} \qquad \begin{aligned} &\text{with } t_{action}\text{: total time the action was used,} \\ &\text{and } t_{total}\text{: total game time,} \\ &\varphi\text{: average action rate;} \end{aligned} \tag{71}$$

If $t_{action} \to 0 \Rightarrow \frac{t_{action}}{t_{total}} \to 0 \Rightarrow s^{mod} \to 0$.
If $t_{action} \to 2 \cdot t_{total} \Rightarrow s^{mod} \to 1$.
If $t_{action} \geqslant 2 \cdot t_{total} \Rightarrow s^{mod} = 1$ (due to the limitation of s^{mod} to $[0, 1]$.

8.4.2 *Learner Model Assessment*

The learner model is assessed prior to the game start. Therefore, players are asked if they played the game before. Three cases are differentiated: *new players*, *returning* *players*, and *experienced players*. Players are considered to be *new players* if they never played the game before. If players played the game 1-2 times without having it completed, they are considered *returning players*. If players finished the game at least once or played it partially for at least three times, players are considered *experienced players*. New players are assumed to have no game knowledge at all. Hence, all their skill values set to 0. Returning players are assumed to have knowledge about the core gameplay elements, especially those which players are confronted in the first part of the game. Hence, the skill configuration shown in Table 23 is used. Experienced players are considered to know the game. Thus, all their skill values are set to 1. Similar to the player model assessment, the learner model is updated by the situation recognition.

Skill	Value
KnowledgeWood, KnowledgeHunt, KnowledgeBerries, KnowledgeGeysir, KnowledgeBottle	1
KnowledgeRaft, KnowledgeWilson, CarryPalm, HuntHeron, FindBottle	0.5
BuildRaft, BuildHut, FillBottle, SleepInHut, BuildFire, CookHeron	0.0

Table 23: EFWI Skill configuration for 'returning players'.

8.4.3 *Interaction Model Assessment*

The interaction model is initialized with a neutral vector for each player, i.e., $(0.5, 0.5)$. Again, like the player and the learner model, it is updated by the situation recognition.

8.4.4 *Challenge Model Assessment*

All challenge values are initialized with 0.0. The challenge codomain ranges from $[-1, 1]$ with $[-1, 0]$ indicating 'too easy' and $[0, 1]$ indicating 'too hard'. Thus, 0.0 is the neutral value (neither 'too hard', nor 'too easy'). Challenge values are also changed by situation objects. The effect of an active situation on the challenge vector is specified in the associated situation object and is applied either event-based (for discrete actions) or as a continuously calculated value for continuous actions.

8.5 THE ADAPTATION PROCESS - GAMEADAPT.KOM AND EFWI

The adaptation mechanism as described in Chapter 5 is implemented as an extension of EFWI. It is configured with the player, learner, and interaction model as described in Section 8.3 after the initial models are assessed as shown in Section 8.4. Two GameAdapt.KOM processes are running:

1. Situation recognition
2. Adaptation selection

The situation recognition periodically evaluates present situations in a 1-second-cycle and updates the integrated group model. The adaptation selection evaluates the defined adaptations in a 1-second-cycle based on preconditions and the current integrated group model and selects adaptations to be applied, if any. The cycle length of one second was chosen as both situation recognition and adaptation selection are supposed to capture the game as closely as possible. However, doing it on a frame basis would be too time consuming in regards to computation time. Therefore, one second turned out to be a good tradeoff between accuracy and work load.

8.6 GAME MASTERING IN EFWI

GameAdapt.KOM is able to provide a human Game Master with the necessary tools to enable him/her to perform his/her tasks in a game-based collaborative learning scenario. Therefore, GameAdapt.KOM needs to contain an interface for the GM providing necessary information about the game process and methods to adapt the game at run-time.

There is no question that the presence of a qualified instructor in the role of a Game Master should always have a positive influence on the learning, gaming, and interaction in such a game. The GM should have access to the same options as the automatic adaption engine, made understandable and comfortably usable for a (qualified) human instructor.

In order to be able to provide the Game Master with the ability to assess the game state and to be able to adapt it according to their professional opinion, the *Game Master Toolkit* has been designed and developed. The *Game Master Toolkit* is the interface for the instructor to the collaborative Serious Game. It makes use of the *Collaborative Multiplayer Serious Game Model* and the game interface which is based on it.

The Game Master Toolkit consists of three elements:

- Information module
- Adaptation module
- GM observation frontend

The *Game Master Toolkit* provides a GUI for the GM inside the actual game session. This front-end provides the relevant information about the game in a meaningful way. It also provides means to trigger the available adaptations.

Relevant information is taken from the *information module* and contains information about the game state (game variables, game actions and events), current situations, and the group model. All information is directly taken from GameAdapt.KOM. This represents a major advantage compared to traditional game-based learning scenarios, where the instructor usually observes players through their view, i.e., by 'looking over the shoulder'. The group model is displayed in a special window providing overview over each player's player model, learner model, and interaction model (see Figure 33).

The GM can view game entities directly in the game world and see their parameters in the settings window. Apart from collected data like the group model, it is useful for the instructor to be able to directly observe the gaming progress [61], possibly

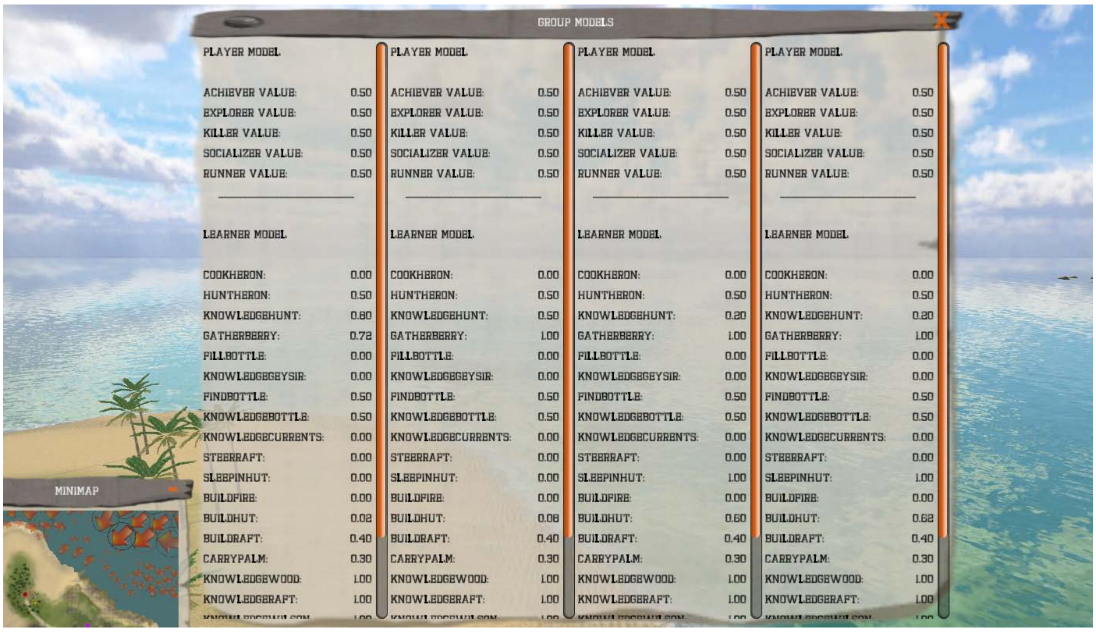

Figure 33: EFWI screenshot of the Game Master GUI - group model.

using aggregated data, in order to be able to extract information about the collaborative learning process. The data aggregation can be performed by GameAdapt.KOM to facilitate the assessment process. Moreover, it might be useful for recognition of problems players might have at certain points in the game and a resulting notification for the GM. Therefore, it seems useful for the GM to be able to move freely in the game world, i.e., have a free camera perspective (see Figure 34).

Moreover, in order to be able to oversee what all players are doing at a certain point of time when players are split up, a split-screen camera is provided. Each of the split screens can be set to either a fixed place or to automatically follow one player (see Figure 35). Apart from this, the GM is provided with the information of the group model.

On top of this, the results of the situation recognition are visualized (see Figure 36). The defined situations are presented to the GM in form of a list of present situations ordered by likeliness. The GM can adjust how many of the most significant situations hs/she wants be have displayed for each player and for the group.

The *adaptation module* enables the GM to adapt the game. The set of adaptations is made available to the GM from GameAdapt.KOM, so that he/she can use them according to his/her professional opinion. The GM can access available adaptations via a special menu in the GUI (see Figure 37). From there it is possible to adjust game variables via sliders and to trigger actions. A special menu to send custom messages to players is available. And, to improve immersion, the GM is able to create dialogues using the NPC. This way it is possible to emulate conversations rather than dropping messages from nowhere onto the players. It is also possible to give items to players or trade items using the NPC. Thus, the GM is able to manipulate relevant 3D objects, game rules (i.e., interaction rules, rules for collaboration, game actions), and difficulty in terms of gaming, or learning.

Figure 34: EFWI screenshot of the Game Master GUI - Game Master perspective. In the left top corner, the camera option menu is located. The minimap is provided in the lower left corner. In the top right corner, recent actions and events are logged and below that the situation recognition window is located. The player attributes are displayed in the bottom center of the screen.

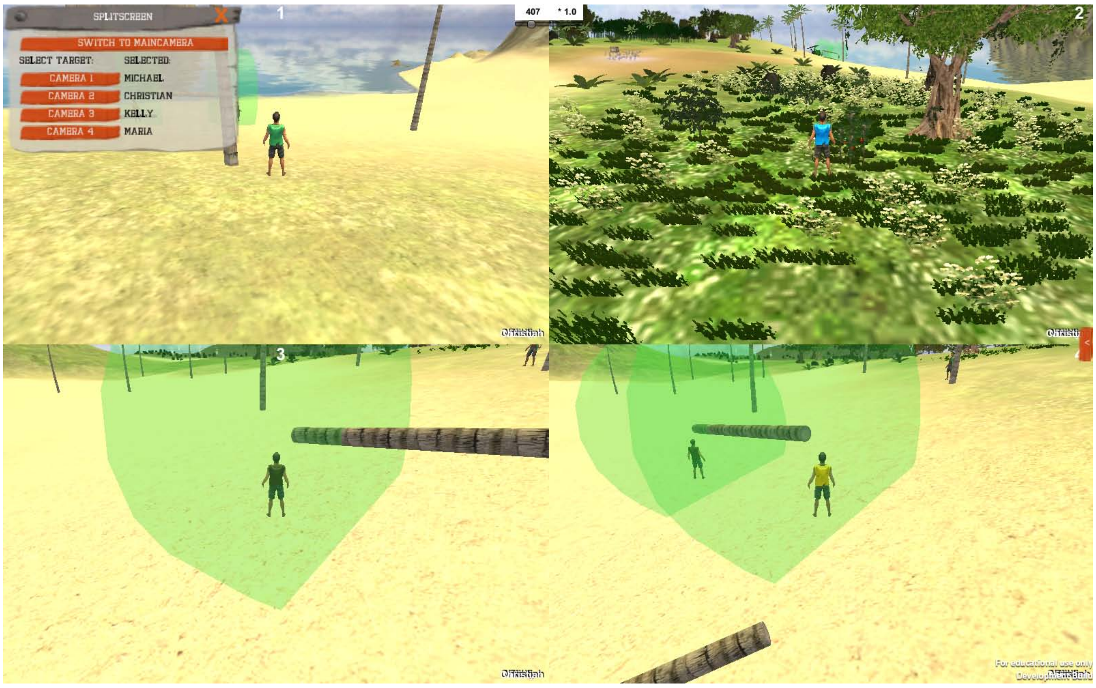

Figure 35: EFWI screenshot of the Game Master GUI - Game Master camera options.

8.7 PLAYER SIMULATION IN EFWI

The player simulation concept was implemented as an extension of *Escape From Wilson Island*. A graphical interface was elaborated and implemented as a frontend view

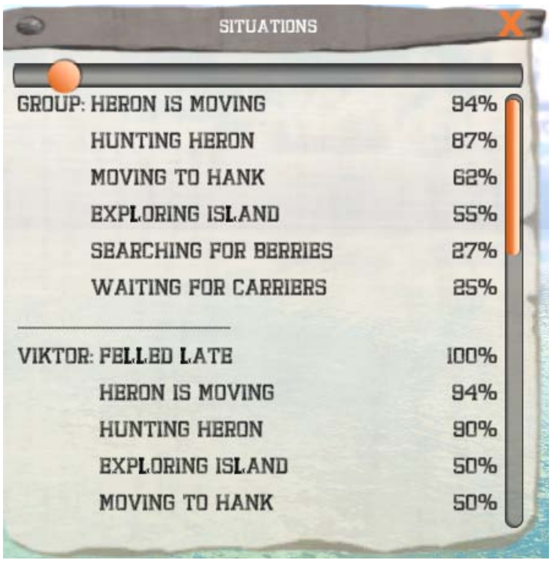

Figure 36: EFWI screenshot of Game Master GUI - situation recognition.

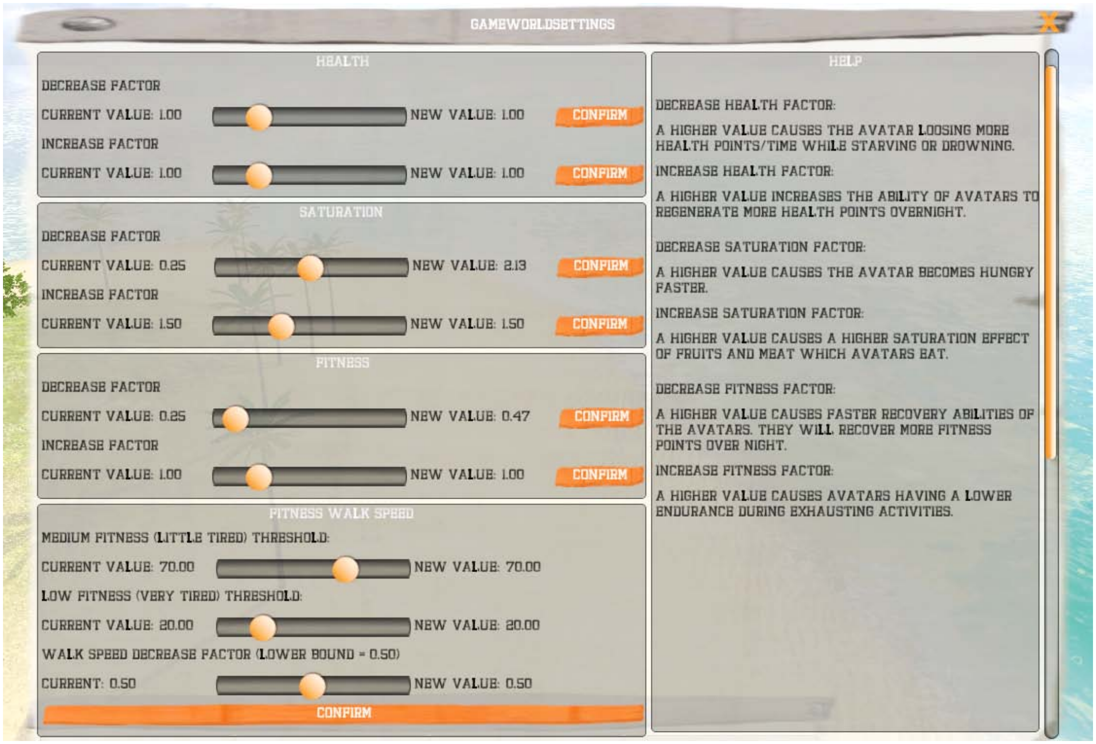

Figure 37: Screenshot of the EFWI adaptation interface.

on EFWI to facilitate the configuration of player traits and player, learner, and interaction model at the start of a game session. Following, the skills, actions, and goals for the simulated players are described as well as the used player, learner, and interaction model.

8.7.1 *Player Simulation Actions*

The simulated players can perform all possible player actions like real players, e.g., fell a palm, gather berries, or carry a palm, etc. Some of the available actions, like felling a palm, do not require a skill since the player only has to decide to execute this action for it to be executed. There is no failure possible. Actions like carrying a palm however require the players who are carrying the palm to stay inside a certain area around the palm while carrying. Those skills are denoted as *motor* skills. Each motor skill is implemented with a failure probability which depending on the skill, i.e., players can fail at an action according to their respective motor skill. For example, carrying a palm is simulated with a randomized offset on the walking direction. If one player steps out of his/her area, the palm is dropped. The offset is multiplied with $1 - \mathtt{skill}_{\mathtt{carryPalm}}$ so that there is no offset when the players 'know' how to carry the palm. Hence, it will not be dropped. The lower the skill level of the player the higher the possible random offset which lets the player more likely step out of his/her carrying area and therefore drop the palm. Furthermore, the *teamwork* skill also influences this performance, giving good team players a bonus and a penalty to bad team players.

8.7.2 *Player Simulation Goals*

In EFWI, the players have several goals that they can or must achieve during the game. Those goals are described in Section 8.2 and mapped to a challenge vector in Section 8.3. For each challenge in the flow model, one goal is defined. For each goal, at least one plan is defined. A plan is a sequence of game actions to be performed by one or more players. For more details on goals and plans, as well as on goal and plan selection, see Section 7.2. A screenshot of the goal overview window in EFWI is provided in Figure 38.

8.7.3 *Player Simulation Group Model*

The player simulation uses the EFWI player, learner, and interaction model, which is explained in detail in Section 8.1.

The implemented interaction model provides the players with the capabilities to share knowledge and ask for help, as described in Chapter 7. Teamwork is modeled as a special skill which is multiplied with the required skill whenever a collaborative task is solved by the team (see above).

8.7.4 *Process of Learning Skills*

Motor skills are increased automatically by small amounts when the player is executing the related task. Motor skills can also be increased when other players or the NPC share knowledge about this skill, e.g., the NPC tells the player how to hunt a heron.

Skills that represent knowledge can be gathered from other players or the NPC and for some skills also by walking around the island. For example, whenever a

Figure 38: EFWI screenshot of the player agents' goal overview.

player is close to a berry bush, *GatherBerries* knowledge is increased to simulate that a real player would notice the highlighted bush and try to interact with it.

The skill *Teamwork* is increased whenever a player executes a teamwork task. The skill *Communication* is increased when a player shares knowledge with others or when he/she receives knowledge from other players.

As some skills do not have predecessors, they represent leaves in the skill graph (see Figure 32). These skills can be learned as soon as a learning condition is met, e.g., gathering berries can be learned whenever a player is close to a berry bush. Hence, whenever a player agent passes a berry bush, the *KnowledgeBerries* skill is increased about 0.05. Although it is not realistic that players slowly learn by passing a bush, but rather would at some point examine one bush and find out that there is an action available ('gather berries'), this is a pretty good emulation of the learning process which includes the need to recognize an object which repeatedly appears in the game world.

The knowledge, that the player can interact with trees is a prerequisite to learning how to carry a palm. Learning that palms can be felled is performed analogue to learning that berry bushes can be harvested. As soon as the player has learned this skill he/she can learn how to carry a palm. Carrying the palm is a motor skill. Motor skills are learnt by executing the related action (e.g., carrying the palm) and by failing at executing the action. In the case of carrying the palm, the player agents 'learn' the skill with 0.01 per second while carrying the palm. Further, whenever they drop the palm, they 'learn' the skill about 0.10. This should reflect a person to become better at a task while the person is actually performing it, and that usually something can be learned from failure.

Being able to carry the palm, in turn enables the player to gather knowledge about building the hut and the raft. If the player knows about building the hut, he/she can learn from the NPC about sleeping in the hut and that this helps recovering energy.

Apart from that built-in learning methods, the simulated player agents can learn from adaptations. This is to reflect that players can be given information, tips, or hints to improve their knowledge regarding tasks or skills. This is done via informative adaptations, adaptations which provide the player with knowledge in form of a message (i.e., a hint or tip). Whenever a message is received, it is considered that the message improves the receiving player's related skill.

Figure 39 shows a screenshot of the player and learner model overview window in EFWI.

Figure 39: EFWI screenshot of agents' player and learner model.

8.8 CHAPTER SUMMARY

As a proof-of-concept for the developed methods and concepts, the collaborative multiplayer Serious Game *Escape From Wilson Island* has been conceptualized, fully implemented, and evaluated in collaboration with AVM Rüsselsheim and the Fachhochschule of the bfi Wien. The intention behind the game design decisions of EFWI was to design and implement a Serious Game for a small group of learners with a focus on collaborative gameplay. The Robinson Crusoenade was chosen as the narration concept to set the background story and player motivation based on that narrative. The game is implemented with the 3D game engine Unity3D (see Section 8.1).

EFWI was designed according to design guidelines for collaborative games [183] and for fostering collaborative behavior in learning scenarios [77] found in the literature. The gameplay was designed to foster collaboration and focus on collaborative gameplay (Section 8.2).

EFWI adaptations and adaptation objects are defined in Section 8.3. The EFWI player model is designed using the player model of Houlette. Five traits are chosen: Curious vs. Anxious, Ambitious vs. Phlegmatic, Acting (interacting with the world) vs. Passive, Interacting (with players) vs. Passive, and Moving vs. Stationary. To model challenge, the flow model contains one vector for each task in EFWI. For learner modeling, a skill graph was designed based on all tasks and required skills for EFWI and their interdependencies. The EFWI skill graph contains 18 skills that were identified as the relevant learning content. Additionally, the two skills *Teamwork* and *Communication* were defined for the interaction model.

For situation recognition in EFWI, first the set of relevant game element objects is defined, followed by information objects. States, tasks, regions, and finally situations and their criteria are listed. Based on these, adaptations and adaptation objects for EFWI are defined.

Section 8.6 covers the implementation of the EFWI Game Mastering concepts, especially the implementation of the GM front end in EFWI. The graphical user interface (Game Master front end) was elaborated and prototypically implemented on top of EFWI. It enables both observation and monitoring of the game process and facilitates manipulation and adaptation of the game by a human instructor.

The agent-based player simulation was implemented to be able to simulate virtual learners/players in EFWI (Section 8.7). Goals and plans for the virtual players were defined to emulate realistic behavior. Moreover, a graphical interface was elaborated and implemented to facilitate the configuration of player features and player, learner, and interaction models at the start of a game session.

The main contribution of this chapter is the design and the development of a Serious Game prototype that allows the implementation of the developed methods and concepts in GameAdapt.KOM. Thereby, the game design followed design guidelines from the literature regarding collaborative learning, multiplayer games, and especially collaborative gaming. Using this Serious Game prototype, an implementation of the GameAdapt.KOM adaptation mechanism, the Game Mastering concept, and the player simulation concept was realized, enabling an evaluation of the elaborated concepts.

9 EVALUATION AND RESULTS

»Play is the highest form of research.«

— Albert Einstein

This chapter covers the evaluation of the concepts proposed in this thesis. As a first step, the player simulation is evaluated for soundness (Section 9.1). This includes testing if the developed player, learner, and interaction model is able to describe reasonable and configurable behavior. The situation recognition is evaluated in Section 9.2 using both simulated player agents and real players. Section 9.3 covers the system parameter evaluation for the automatic game adaptation and in Section 9.4 the evaluation of the automatic game adaptation using simulated players is described, followed by the evaluation of the automatic game adaptation using real players in Section 9.5. The effect of a Game Master on the performance and learning success of players when using the GameAdapt.KOM framework to orchestrate a game session is assessed in Section 9.6, again using real players.

9.1 SIMULATED PLAYER MODEL EVALUATION

As described in Chapter 7, the approach for the simulation of players is to observe real players while playing the game and to model goals and plans from the observed behavior for the simulated players. The simulated player agents then rate their available goals based on their player, learner, and interaction model, choose the best available plan for the goal and execute it. Validity of the behavior is evaluated by comparing the resulting behavior with the expected behavior considering the player, learner, and interaction model configuration. Hence, the goal of this evaluation is to evaluate the impact of the parameters of the player, learner, and interaction model developed in Chapter 7. So, the simulated player model is evaluated for the meaningfulness of player agent behavior regarding the chosen parameters. Simulated players are considered to behave meaningful if modifications to their models imply recognizable and understandable changes of their behavior.

9.1.1 *Experiment Design and Setup*

To decide whether player agent behavior is meaningful, it is judged according to various criteria. These criteria are based on observations of real players playing the game. The observations were matched to player, learner, and interaction models. Simulated players should behave according to their player, learner, and interaction models, i.e., they should behave similar to a real player with the same model. However, it needs to be taken into account that a direct comparison to a real player is hardly possible, as a real player's player, learner, and interaction model is only rarely known and almost never well-marked in the various traits or skills, which, when quantifying a real player's model, would rarely result in clear characteristics like 0.0 or 1.0.

Rather, real players represent a mixture in their characteristics of traits and skills, e.g., it is more likely that a real player has a player model of $(0.3, 0.4, 0.8, 0.2, 0.7)$ than $(0.0, 0.0, 0.0, 1.0, 1.0)$. Hence, when evaluating simulated player behavior, it is compared to idealized criteria, i.e., to how a player with that model should behave. Player agent behavior should be reflected

- in their preferences of pursuing goals (player model)
- in their (mechanical) ability to solve tasks (learner model)
- in their knowledge about tasks (learner model)
- in their sharing of knowledge and willingness to help (interaction model)

Thus, simulated players should differ in the order they pursue tasks and in their success at solving those tasks. This is reflected in their performance regarding those tasks and the overall game.

For a quantitative evaluation, player behavior is measured based on logged game data. Therefore, the game state and player states are logged periodically and player actions are logged event-based. From this log data, knowledge about player behavior can be extracted.

Player health, player satiety, and player fitness are logged periodically (once per second): Average health, satiety, and fitness over the whole game session can be calculated for each player from this. Moreover, the progression of those three values for each player can be plotted from this data. Other raw game data is logged event-based (see Table 24). The total number of berries gathered, herons hunted, meat gathered, palms felled, palms dropped and interactions with the NPC can be calculated from that. Furthermore, the timestamp of the first occurrence for each of those events can be taken from that information.

Collected Raw Data	Logging point	Derived information
Player health	periodically	Average player health
Player satiety	periodically	Average player satiety
Player fitness	periodically	Average player fitness
Talked to NPC	event-based	Total NPC interaction
Berries gathered	event-based	Total berries gathered
Heron hunted	event-based	Total herons hunted
Gathered meat	event-based	Total meat gathered
Empty bottle found	event-based	Time needed to find the bottle
Bottle filled	event-based	Total time needed to fill the bottle
Palm dropped	event-based	Number of 'carry palm' attempts
Log hut finished	event-based	Time to build log hut
Raft built	event-based	Time to build raft
Game won	event-based	Time to finish game

Table 24: Overview over the logged simulation data. Data is either logged event-based or periodically.

Table 24 summarizes the logged data gathered from the game and their purpose for evaluation of the player agent AI. Thus, independent variables (stimuli) are:

1. The player model configuration of player agents
2. The initial skills of player agents

The five player model traits were varied between $[0, 1]$. Likewise, the 18 skill values were varied between $[0, 1]$.

The observed and via log data measured player agent behavior is the response (dependent variables). In detail, dependent variables are:

- Time needed to find the bottle
- Time needed to fill the bottle
- Time needed to build the log hut
- Number of berries gathered
- Number of herons hunted
- Number meat roasted
- Average satiety
- Average fitness
- Average health

It is expected that player agents with a higher initial knowledge perform better on the time based items as they do not need time to first acquire the required knowledge. Instead, they can include the respective tasks into their initial planning. It is further expected that a team consisting of players with high values in the traits 'curious' or 'moving'[1] players will tend more to exploring the island, thus finding the empty bottle faster and finding berry bush locations, thus being able to gather berries faster even without the initial knowledge skill.

'Ambitious' players are considered to try to maximize measurable achievements, i.e., the number of berries gathered or herons hunted. Hence, it is expected that teams with lots of achievers tend to have high values on those items.

'Acting' players tend to fight or hunt where possible. This is reflected best in hunting herons. Subsequently, it is expected that 'acting' players tend to have higher values on the number of hunted herons.

'Interacting' players are expected to interact with the NPC as soon as possible and as much as possible.

'Anxious' players will probably stay away from the water or cliffs.

'Stationary' players might tend to stay near the spawn point or near the log hut which are focus points of the game.

The evaluation was conducted using the following setup: An instance of the game was started and and configured to function as a server in the simulation mode. The player agent configuration was set up on the server instance. Next, four other instances of the game were started and connected as clients. Once all clients were connected, the server instance started the game. The simulation ran for 2730 seconds, which are 45 minutes (2700 seconds) + 30 seconds to start the game.

For the skill configuration, skills were clustered according to their related task (compare Figure 32). The cluster topics are shown in Table 25. In order to reduce the number of parameters, the simulation runs do not include the second part of the game (i.e., the second island). Hence, building the raft is not part of the simulation runs. Thus, cluster E is omitted. Resulting, there are four clusters, forming the four parameters of a $2^k \cdot r$ factorial design with replications, with $k = 4$ and $r = 3$. Each

1 the player model traits used in this evaluation are explained in Section 8.3

Set	Task	Skills
A	Berries	KnowledgeBerries
B	Palm	KnowledgeWood, CarryPalm, BuildHut, SleepInHut
C	Bottle	KnowledgeGeysir, KnowledgeBottle, FindBottle, FillBottle
D	Hunt	KnowledgeHunt, HuntHeron, CookHeron, BuildFire
E	Raft	KnowledgeRaft, KnowledgeWilson, BuildRaft

Table 25: EFWI skill clusters.

cluster hereby is treated as one input parameter. For the lower value of each factor (−), all skills of the cluster are set to 0. For the higher value of each factor (+), all skills of the cluster are set to 1. The player models of all four players are set to a neutral shape of $(0.5, 0.5, 0.5, 0.5, 0.5)$. The interaction skills are set to $(1.0, 1.0)$.

For the player model configurations, all skills were fixed to 0.5 and the player model traits were varied. The player model configuration parameters were chosen as shown in Table 26. Sets 1 through 5 each have one trait set to 1.0 for all four players, whereas the other traits are set to 0.0. In set 6 each player has a different trait set to 1.0, so that each trait is set to 1.0 at exactly one player (Note: as there are only four players, but five traits, one player had both 'curious' and 'moving' set to 1.0 to represent an 'explorer'-like player type). Set 7 is the average over the previous $2^k \cdot r$ factorial design runs where the player model was set to 0.5 for all traits of all players. All configurations were repeated three times.

	Curious	Ambitious	Acting	Interacting	Moving
Set 1	1.0	0.0	0.0	0.0	0.0
Set 2	0.0	1.0	0.0	0.0	0.0
Set 3	0.0	0.0	1.0	0.0	0.0
Set 4	0.0	0.0	0.0	1.0	0.0
Set 5	0.0	0.0	0.0	0.0	1.0
Set 6	0.0/1.0	0.0/1.0	0.0/1.0	0.0/1.0	0.0/1.0
Set 7	0.5	0.5	0.5	0.5	0.5

Table 26: Player model configuration for the evaluation of player model influence on player behavior.

For the interaction model configurations, another $2^k \cdot r$ factorial design with replications with $k = 2$ and $r = 3$ is performed. The two factors are

1. the configuration of the two interaction (IA) skills
2. the configuration of all other skills

For the lower value of each factor (−), all skills are set to 0.0. For the higher value of each factor (+), all skills are set to 1.0. The player models of all four players are set to a neutral shape of $(0.5, 0.5, 0.5, 0.5, 0.5)$.

For all three simulation sets, the observed responses are shown in Table 27.

Name	Explanation
AverageHealth	average player health over time
AverageSatiety	average player satiety over time
AverageFitness	average player fitness over time
BerriesGathered	total number of berries collected
HeronsHunted	total number of hunted herons
MeatRoasted	total number of roasted meat
HutBuildTime	timestamp of the moment the hut is finished
FindBottleTime	timestamp of the moment the bottle is found
FillBottleTime	timestamp of the moment the bottle is filled

Table 27: Observed response values of the three studies of the simulated player model evaluation.

9.1.2 *Results and Discussion*

For each simulated game session, the items stated above were logged and the performance values were calculated. The average over the three sessions was calculated.

9.1.2.1 *Skill variation (learner model)*

a priori

Table 58 (see Appendix B) shows the proportion of variation of each of the four skill clusters (A, B, C, and D) for the eleven response values 'health', 'satiety', 'fitness', 'palms felled', 'berries gathered', 'herons hunted', 'meat roasted', 'palms lifted', 'hut build time', 'bottle found time', and 'bottle filled time'. Table 59 shows the absolute influence of the variation of the skill clusters in relation to the mean response of the eleven response values. The most significant skill clusters are drawn for each response value in Figure 40 compared to the mean response value.

The following observations can be made: Skill set A (KnowledgeBerries) has a visible effect on the average satiety and the number of berries gathered. It is responsible for 31% of the average satiety values. On average it improves the mean satiety of 51.96 about 10.38.

Skill set B (KnowledgeWood, CarryPalm, BuildHut, SleepInHut) has a very high impact on the average health (90%) and fitness (89%). This shows an average improvement of 17.91 on the average health (65.02) and of 12.06 on the average fitness (31.46). Subsequently, skill set B has a major impact on the hut build time (90%), as expected (457s better than the average of 1546). It also has a major impact on the number of gathered berries (63%), reducing the average number of berries gathered about 13.48 with an average number of 102.21 berries gathered.

Skill set C has a major influence on the time needed to find the bottle (86%) and to fill it (51%), as expected. The average time needed to find the bottle (612s) is reduced about 274s and the average time to fill the bottle (1133s) is reduced about 254s.

Skill set D does not have any notable effect. This might be explained by the fact that players have already learned those skills until they are needed in the game.

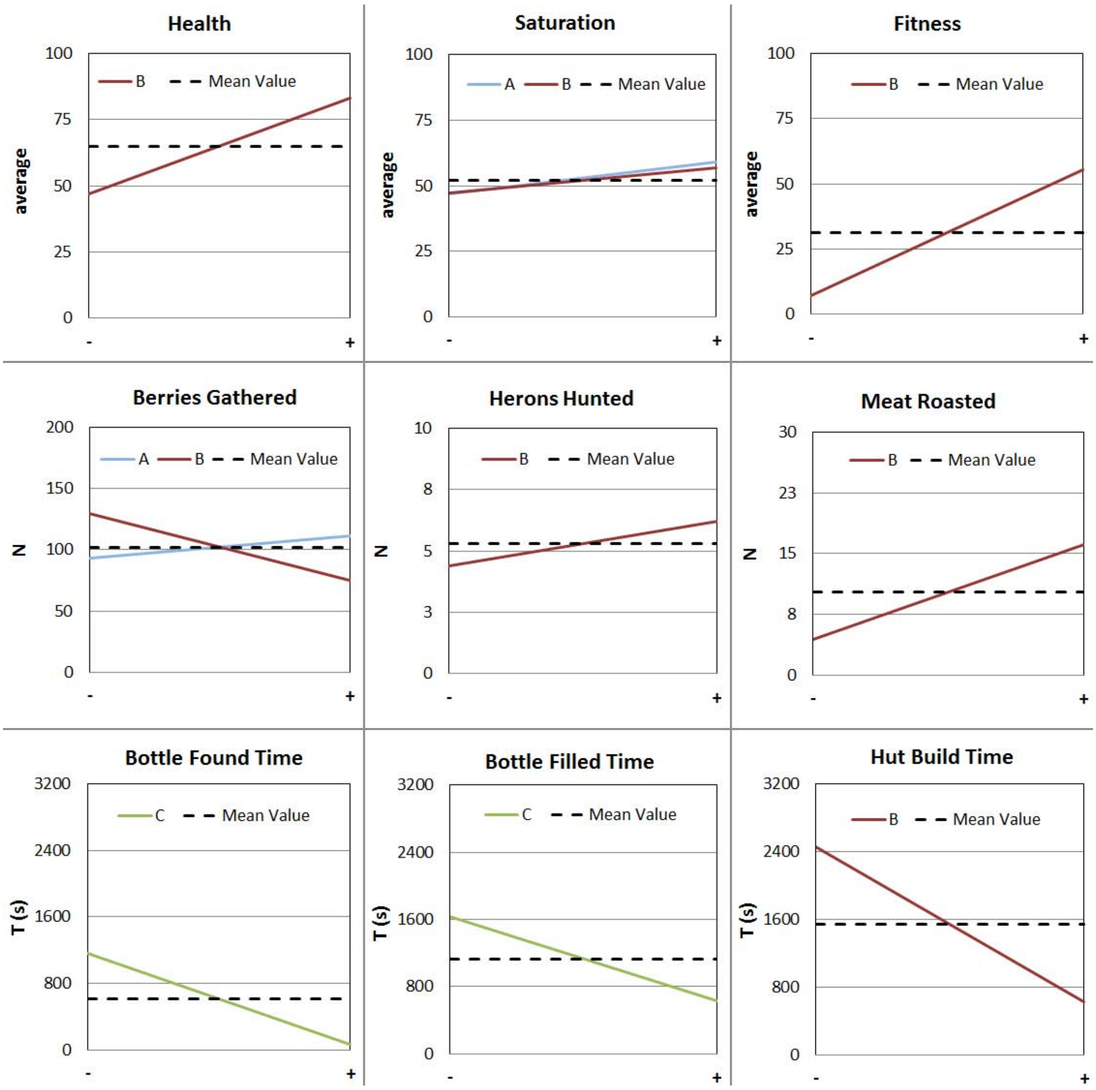

Figure 40: Visualization of the influence of the skill sets A, B, C, and D on the measured responses.

INTERPRETATION

Regarding skill set A, it can be concluded that changing the skill has the intended effect on the average satiety.

Regarding skill set B, the improvement of the average health and fitness can be explained with the fact that the related skills skills facilitate the hut building, which is a prerequisite to sleeping, which again enables regeneration of health and fitness. The impact on the number of berries gathered implies that on average, the players collect less berries when they are able to build the hut from the beginning. This can be explained with the fact that those players do not need to spend time to learn how to build the hut (in which they need to eat) and that players earlier get meat which makes gathering berries redundant.

Skill set C also has the expected influence on the time needed to find and fill the bottle.

Skill set D, however, does not have the expected impact.

Altogether, it can be stated that variation of the skill sets A, B, and C do have the predicted effect on player behavior.

9.1.2.2 *Player Model Variation*

A PRIORI

Table 60 (see Appendix B) contains the results of the player model parameter assessment. Figure 41 shows the mean response value for the chosen player model (with standard deviation) and the mean response (in red).

The player model configuration shows the following impacts on the player agent behavior and subsequently the player agent performance (only values which differ from the average are discussed explicitly):

Ambitious players are fastest at building the log hut (807s on average). Moreover, they show the best values of all configurations in health (74.10) and fitness (41.28) and an above average value in satiety. They further roasted the most meat (15.67). They show a below average number of gathered berries (101).

Curious players show high number of berries collected (140.33) and a short time to find (808s) and fill (1226s) the bottle, which is only outperformed by the set of mixed players (731s and 944s). However, they need longer than average to build the log hut (2161), or hunt less than average herons (6.00 meat chopped).

The set of 'acting' players shows high values for berries gathered (130.00), meat roasted (11.33) and subsequently, in satiety (71.90). Further, this set of players takes rather long to build the hut (1766s) or find (2078s) and fill (2504s) the bottle.

Interacting players show the smallest amount of gathered berries (97.33) and the most hunted herons (8.00). They tend to find the bottle late (1559s), and on average they fill the bottle within 209 seconds.

Moving players do find the bottle early (1082s) but on average fill it after 2194s. They also show a high number of collected berries (127.67).

INTERPRETATION

Ambitious players try to optimize their performance. Hence, they try to achieve available goals as fast as possible. This is shown in their prioritization if building the hut, their high average health and fitness values and the high number of roasted meat. The low number of gathered berries can be explained with those players prioritizing to get the better food (i.e., roasted meat). This meets the expected behavior.

Curious players tend to search the island. Thus, they are quickest to find berry bush locations or the bottle. This is reflected in the high number of berries collected and the short time to find and fill the bottle. However, they do not prioritize building the log hut or hunting herons. This behavior is congruent to the expectations.

The combination of the two traits 'curious' and 'moving' for one player in set 6 (mixed) appears to improve the need to search the bottle and subsequently let the players find the bottle even earlier. The combination of a player who is both 'curious' and 'moving' and another 'ambitious' player lets the team both find the bottle early ('curious'+'moving') and fill it as soon as possible ('ambitious').

The acting players tend to operate selfishly (analogy to Bartle's 'killer' type), which is reflected in their high number of berries gathered and meat roasted. Hence, they tend to pursue those goals which they can achieve alone (e.g., 'gather berries' and 'roast meat'). They also tend to keep their personal values (health, satiety, and fitness)

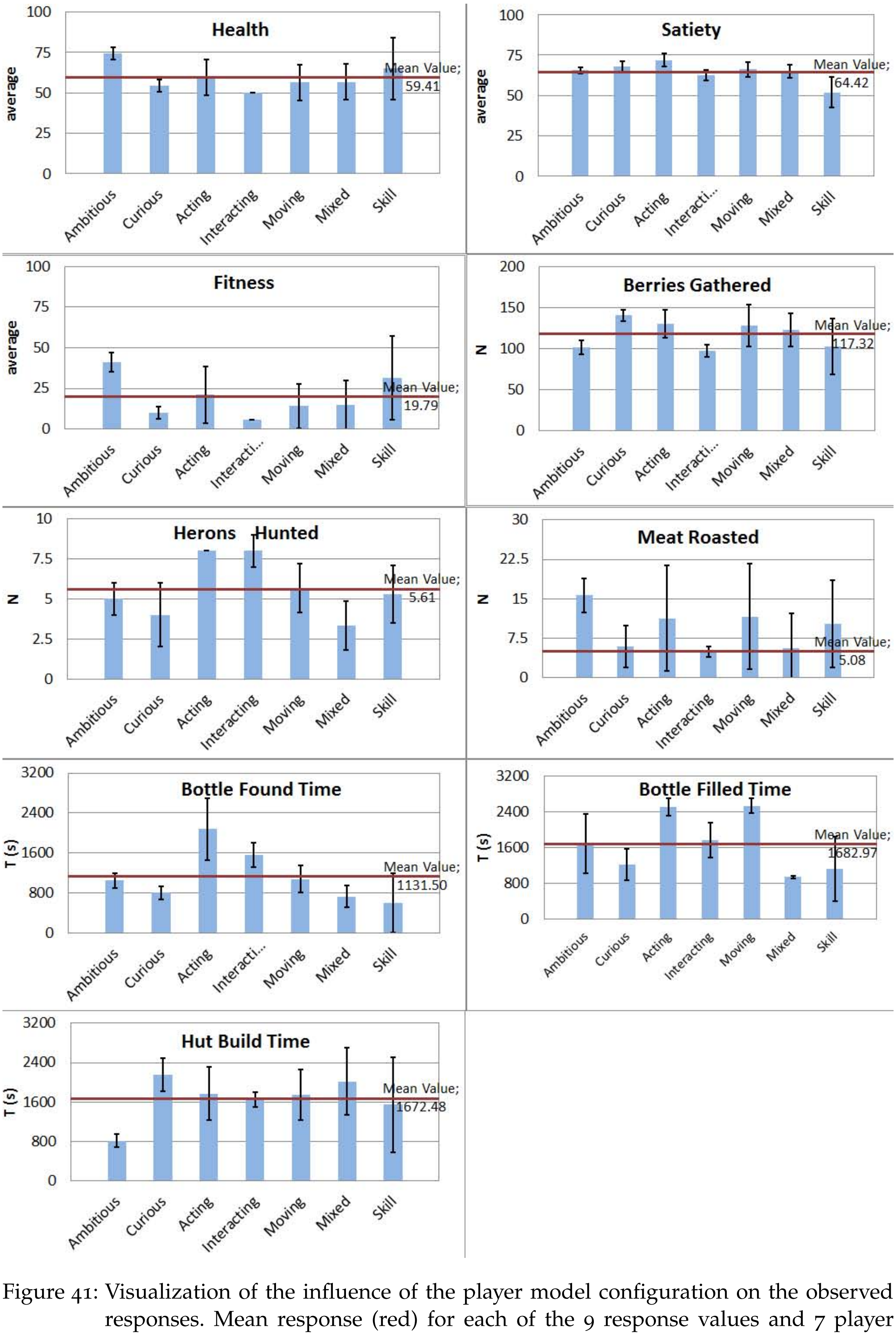

Figure 41: Visualization of the influence of the player model configuration on the observed responses. Mean response (red) for each of the 9 response values and 7 player model configurations (blue) with standard deviation.

high. Therefore, berries or meat is the best choice. As a consequence, this set of players takes rather long to build the hut or find and fill the bottle.

As 'gathering berries' is a task which requires no interaction with other players, it appears reasonable that interacting players show the smallest amount of gathered berries. 'Hunting herons' is a task which requires intense interaction, hence it is considered reasonable that they show the most hunted herons. Although interacting players tend to find the bottle late, they try to fill it as soon as possible once it is found. Thus, they try to carry out this interactive action as soon as it is available. This meets the expected behavior for interactive players.

Moving players do find the bottle early which appears reasonable as it is more likely to find the bottle when moving around the island. However, they do not care to fill it soon. Due to their movement, they also find berry bushes quickly which results in a high number of collected berries.

Summarizing, it can be stated that the players behave as expected in relation to their player model configuration.

9.1.2.3 *Influence model variation*

A PRIORI

Table 61 (see Appendix B) shows the proportion of variation for each of the two factors (interaction skill, all other skills). Table 62 shows the mean values for each response and each factor's impact. The influence of the two skills is drawn for each response value in Figure 42 compared to the mean response value.

It can be seen that the interaction skills do have a positive impact on the number of herons hunted (13%) increasing the average of 5.25 about 0.58, the time needed to build the hut (29%), lowering the average of 668s about 57s, and the time to fill the bottle (21%), lowering the average of 1335s about 312s.

The learning skills are responsible for 87% of the variation in average health (+15.47 compared to 59.80 on average) For satiety it is 69% (+7.49 compared to 48.52 on average), and for fitness 84% (20.36 compared to 25.91 on average) It impacts the variation of the number of berries gathered (65%), lowering the average of 107.08 about 22.58, the number of herons hunted (44%), increasing the average 5.25 about 1.08, and meat roasted (75%), increasing the average +7.92 about 7.92. Moreover, it positively affects the time needed to build the hut (59%) about −541s compared to 2008s on average), to find the bottle (87%) about −605 compared to 667 on average), and to fill the bottle (63%), about −534s compared to 1356s on average.

There is no notable interaction effect between the two stimuli (all $\leqslant$ 10%).

INTERPRETATION

The interaction skills positively influence those values which result mainly from collaborative actions (hunt the heron, build the hut, fill the bottle). This implies that players with high interaction skills are better at solving collaborative tasks which meets the expectations. The learning skills have a large influence on various performance values which confirms the results of the skill variation. The low interaction between the two stimuli indicates that there is no interaction between learning skills and interaction skills.

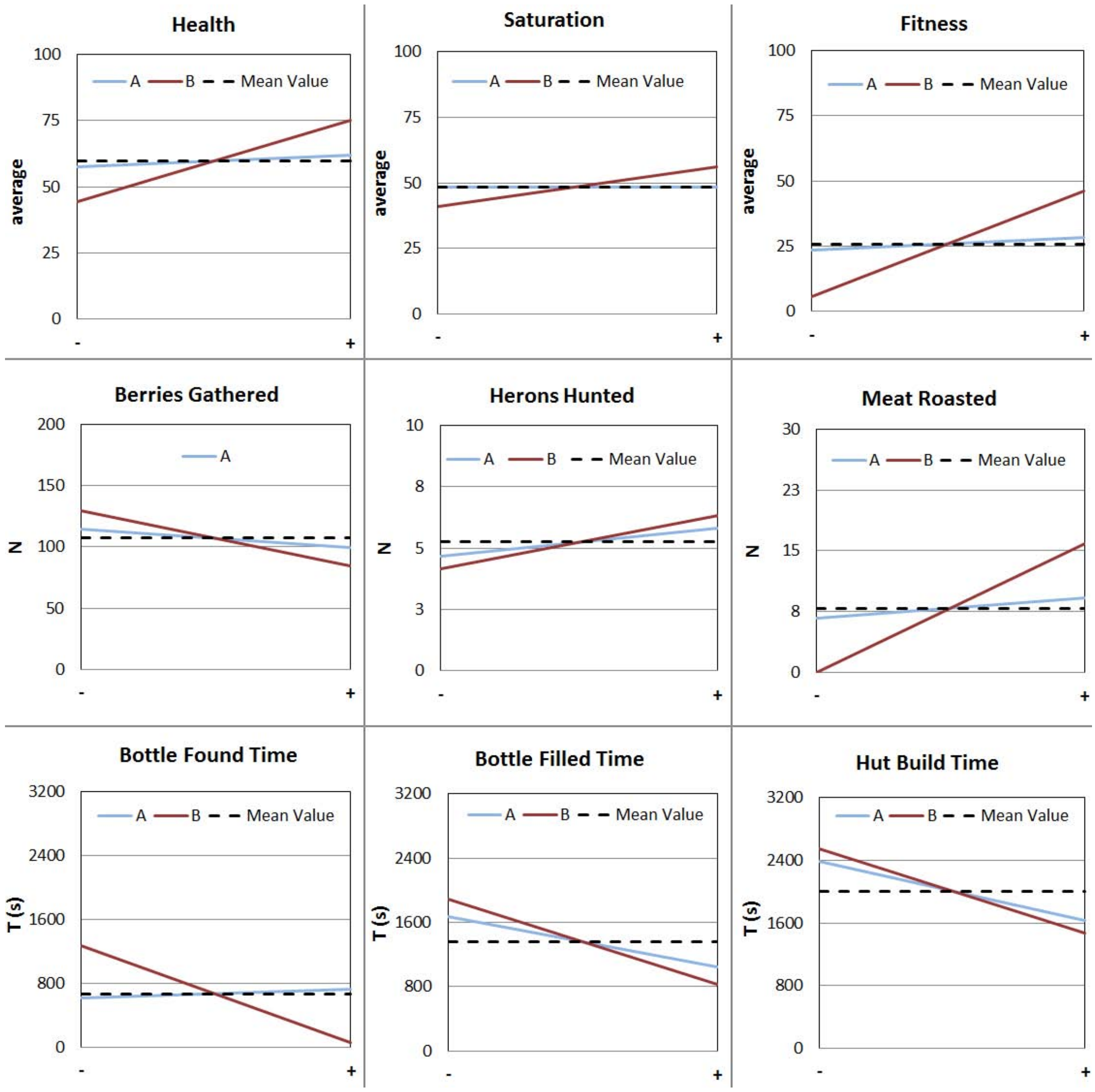

Figure 42: Visualization of the influence of the interaction skills (IM) configuration on the observed responses. 'A' refers to the interaction skills and 'B' to all other skills.

9.1.2.4 *Discussion*

Altogether, it can be stated that variation of the skill sets A,B, and C do have the predicted effect on player behavior. Skill set D, however, does not have the expected impact.

Moreover, it can be concluded that the configuration of the interaction skills does impact the player agent behavior and subsequently the simulated players' performance as intended. The variation of the interaction skills does have the expected influence on solving the collaborative tasks.

Summarizing, with respect to all five traits of the player model, it can be said that the trait configuration is reflected in player behavior and in player performance in an understandable way. Hence, it can be stated that the player model used enables to simulate players with a configurable player behavior with reasonable and understandable effects. Considering the learner model, it can be stated that it enables to configure player knowledge about the game EFWI resulting in understandable and reasonable player behavior.

Thus, it will further be used as a meaningful substitution of real players for further evaluations of the adaptation mechanism. This makes it possible to evaluate the adaptation mechanism with an unlimited number of (virtual) players using the desired player and learner model in contrast to real players whose player and learner model can be assessed but not configured as desired.

9.2 EVALUATION OF THE SITUATION RECOGNITION

In this section, the evaluation of the situation recognition is described. The purpose of this evaluation is to assess the quality of the situation recognition, i.e., how reliable it can correctly recognize situations.

9.2.1 *Experiment Design and Setup*

To measure the quality of the situation recognition, a twofold approach is chosen. As a first step, the results of the situation recognition are compared to the goals of the simulated players in a scenario with AI players. The AI goals accurately display what the respective player is doing. Hence, it is possible to compare the situation recognition with that data.

Second, a set of real players is instructed to play the various situations within a game. This allows to correctly define what players are doing and compare this information to the data from the situation recognition. It is assumed that this approach will produce more accurate results than observing a real game session and manually annotating what players are doing. The main problem with the second approach is its invasiveness. It would be necessary to constantly ask players what they are doing and with what goal in mind which would continuously disrupt their game. Hence, the first approach is chosen.

Appendix A contains a complete list of all situations defined in EFWI. Those situations are subdivided into three sets. The first set contains situations describing a state of action for the group, i.e., the group is just doing the related action. The second set indicates a problem due to the fact that the group did not achieve a task within the required time. The third set indicates a problem due to players solving a task too quickly or too easily.

For this evaluation, only the first set is relevant, as the two other sets are always recognized correctly with 100% accuracy, as they are based only on boolean expressions over game variables which evaluate to *true* or *false* at any point in the game, with *true* meaning the situation is present and *false* meaning the situation is not present. Hence, at every point during the game, it can be decided correctly if this situation is present or not). This can always be recognized correctly. Therefore, recognition of those situations is trivial and is not part of this evaluation.

Hence, only the following situations (i.e., the situations of the first set) are used for evaluation as recognition of these situations is not trivial:

- SearchingForBerries,
- SearchingForBottle,
- BuildingHut,
- TalkingToHank,
- HuntingHeron.

The following metrics are used to benchmark the quality of the situation recognition:

1. The number of correctly recognized occurrences of a situation; an occurrence is considered recognized correctly if it is recognized at least once while the situations was present.
2. The proportion of time the situation recognition correctly recognized a situation while it is present.
3. The proportion of time the situation recognition incorrectly recognized a situation while it is not present.

Regarding 1:
A situation is considered recognized if during the period where the situation is active $T^{sit}_{active} = t \mid t \in \left[t^{sit}_{start}, t^{sit}_{end}\right]$, the situation recognition at least once considers this situation to be present with a probability of $p(sit) \geqslant$ a threshold ρ. The function $r_{rec}(sit)$ hence returns 1, if the situation is recognized at least once, else 0.

$$r_{rec}(t, sit) = 1, \text{ if } \exists_{\geqslant 1} t \mid (t^{sit}_{start} \leqslant t \leqslant t^{sit}_{end} \wedge p(t, sit) \geqslant \rho_{sit}), \text{ else } 0 \tag{72}$$

Regarding 2:
Again, a situation is active in $T^{sit}_{active} = t \mid t \in \left[t^{sit}_{start}, t^{sit}_{end}\right]$.

The proportion of time, the situation is *correctly* recognized is $T^{sit}_{correct} = t \mid (t \in \left[t^{sit}_{start}, t^{sit}_{end}\right] \wedge p(t, sit) \geqslant \rho_{sit})$. The function $r_{prop}(sit)$ denotes the fraction of time the situation is recognized correctly:

$$r_{prob}(t, sit) = \frac{T^{sit}_{correct}}{T^{sit}_{active}} \tag{73}$$

$r_{prob}(t, sit)$ evaluates to 1, if the situation recognition recognizes the situation for the complete time span that it is present.

Regarding 3:
Let t^{obs}_{start} denote the start of observation, and t^{obs}_{end} the end of observation. $\hat{T}^{sit}_{active}$ denotes the total time the situation was active in $[t^{obs}_{start}, t^{obs}_{end}]$. $\hat{T}^{sit}_{inactive}$ denotes the total time the situation was not active in $[t^{obs}_{start}, t^{obs}_{end}]$ with $\hat{T}^{sit}_{active} + \hat{T}^{sit}_{inactive} = t^{obs}_{end} - t_{start}$ (at any point during observation, the situation is either active or inactive). The proportion of time, the situation is *incorrectly* recognized as present (false positive) $T^{sit}_{false} = t \mid (t \in \hat{T}^{sit}_{inactive} \wedge p(t, sit) \geqslant \rho_{sit})$. The function $r_{prop}(sit)$ denotes the fraction of time the situation is recognized incorrectly:

$$r_{false}(t, sit) = \frac{T^{sit}_{false}}{T^{sit}_{active}} \tag{74}$$

The evaluation was conducted using the following setup. An instance of the game was started and configured to function as a server in the simulation mode. The player agent configuration was set up on the server instance. Next, four other instances of the game were started and connected as clients. Once all clients were connected, the server instance started the game. The simulation ran between 900 and 1200 seconds, depending on how long the player agents need to finish the log hut.

The simulation runs were repeated 5 times using the configurations for the virtual players as shown in Table 28. Learning and interaction skills were fixed to 1.0 for all skills. This limits delay due to lack of knowledge/skills. Regarding the player model, the four players were set to randomly chosen values between 0.5 and 1.0 for each trait. (Note: While it is not realistic that players have high values in all traits, it ensures that they pursue all goals because due to the high traits they have a high interest in all goals. Hence, this ensures that all goals are pursued by the players and subsequently that the related situations will occur and can be observed.) The run was repeated five times with the configurations as shown in Table 28.

	IA Skills	Skills	Player Model
Set 1	1.0	1.0	$(0.5, 0.7, 0.8, 1.0, 0.6)$ $(0.8, 0.5, 0.5, 0.6, 0.7)$ $(0.5, 0.9, 0.6, 0.7, 1.0)$ $(0.7, 0.9, 0.8, 1.0, 0.7)$
Set 2	1.0	1.0	$(0.8, 0.9, 1.0, 0.7, 1.0)$ $(0.8, 0.9, 0.5, 0.6, 0.6)$ $(0.5, 0.9, 0.8, 0.9, 1.0)$ $(0.8, 0.8, 0.5, 0.7, 0.7)$
Set 3	1.0	1.0	$(0.7, 0.9, 0.5, 0.8, 0.5)$ $(0.8, 0.6, 1.0, 0.6, 0.6)$ $(0.6, 1.0, 0.7, 0.6, 0.5)$ $(0.8, 0.8, 0.6, 0.8, 0.6)$
Set 4	1.0	1.0	$(0.8, 0.7, 0.6, 0.5, 0.9)$ $(0.7, 0.9, 0.7, 0.9, 1.0)$ $(1.0, 0.7, 0.5, 0.7, 0.5)$ $(0.6, 0.7, 0.9, 0.5, 0.9)$
Set 5	1.0	1.0	$(0.6, 0.9, 0.9, 1.0, 0.9)$ $(0.5, 0.7, 0.7, 0.8, 0.6)$ $(0.6, 0.9, 0.7, 1.0, 1.0)$ $(0.8, 0.6, 0.5, 0.7, 0.5)$

Table 28: Player agent configuration for the situation recognition evaluation.

For all player agents, their active goal is logged in a 1-second interval. The recognized situations for all players and the group was also logged in a 1-second interval, synchronized with the player goal logging. Only goals are considered which were pursued for at least 3 seconds. If a goal was pursued for a shorter time, it is assumed that it was discarded again. As goals are not immediately pursued, thus reflected in behavior which can be recognized correctly, the first two seconds of the situation are considered a start-up phase. During this phase, the situation is not yet considered to be active. Similarly, at the end of the situation, a grace period is of two seconds is added reflecting that a change of goal, i.e., an end of a situation is not immediately reflected in a player's actions. During this grace period, the situation recognition results are not considered as false positive.

For the second part of the evaluation instructed 4 human players (3 male, 1 female, age ranging between 22 and 34) were recruited. Again, an instance of the game was started and and configured to function as a server. Next, four other instances of the game were started and connected as clients. Once all clients were connected, the server instance started the game.

The players were instructed to play the situations described above. The in-game timestamps at starting and finishing the situations were noted. The recognized situations for all players and the group was logged in a 1-second interval. Again, the runs were repeated 5 times.

9.2.2 *Results and Discussion*

A PRIORI

Table 29 shows the results of the aggregated data and the evaluation metrics for the 10 runs. An exemplary visualization of the situation recognition for each situations is shown in Figures 43 (Searching for berries), 44 (Searching for bottle), 45 (Hunting heron), 46 (Building hut), and 47 (Talking to Hank).

From Table 28, one can see that all situations were recognized correctly (metric 1).

The *SearchingForBerries* situation, moreover, was recognized correctly during 90% of the time it was active (metric 2) with 25% of false positive (metric 3) on average.

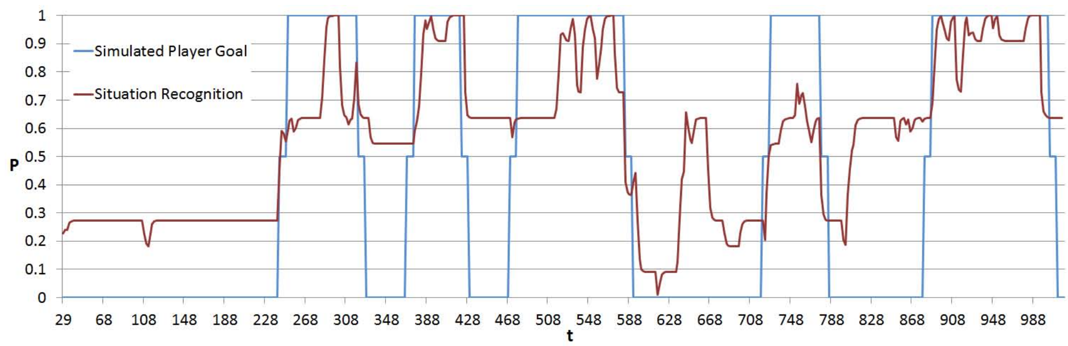

Figure 43: Sample situation recognition of the *Searching For Berries* situation (with simulated players)

The *SearchingForBottle* situation was recognized correctly only during 34% of the time it was active (metric 2) with only 4% of false positive (metric 3) in the setting with simulated players. It was recognized correctly only during 57% of the time it was active (metric 2) with only 2% of false positive (metric 3) in the setting with real players.

The *HuntingHeron* situation is recognized over 52% (metric 2) of its active time with a false positive of 3% (metric 3).

SITUATION		No. Occ.	RECOGNIZED (METRIC 1)	RATIO (METRIC 2)	FALSE POS. (METRIC 3)
Searching For Berries	Simulated	98	1.00 ± 0.00	0.92 ± 0.07	0.28 ± 0.20
	Real players	7	1.00 ± 0.00	0.84 ± 0.19	0.20 ± 0.11
	Overall	105	1.00 ± 0.00	0.90 ± 0.12	0.25 ± 0.17
Searching For Bottle	Simulated	8	1.00 ± 0.00	0.34 ± 0.12	0.02 ± 0.02
	Real players	9	1.00 ± 0.00	0.57 ± 0.30	0.02 ± 0.01
	Overall	17	1.00 ± 0.00	0.48 ± 0.27	0.01 ± 0.02
Hunting Heron	Simulated	31	1.00 ± 0.00	0.44 ± 0.13	0.06 ± 0.09
	Real players	15	1.00 ± 0.00	0.64 ± 0.13	0.02 ± 0.00
	Overall	46	1.00 ± 0.00	0.52 ± 0.16	0.03 ± 0.07
Building Hut	Simulated	42	1.00 ± 0.00	0.52 ± 0.14	0.02 ± 0.02
	Real players	15	1.00 ± 0.00	0.63 ± 0.10	0.04 ± 0.03
	Overall	57	1.00 ± 0.00	0.57 ± 0.13	0.03 ± 0.02
Talking to Hank	Simulated	48	1.00 ± 0.00	0.38 ± 0.20	0.01 ± 0.01
	Real players	11	1.00 ± 0.00	0.90 ± 0.08	0.02 ± 0.02
	Overall	59	1.00 ± 0.00	0.58 ± 0.31	0.01 ± 0.01

Table 29: Situation recognition accuracy. For the five situations the number of occurences of each situation, the recognition, the recognition ratio, and the ratio of false positives are shown.

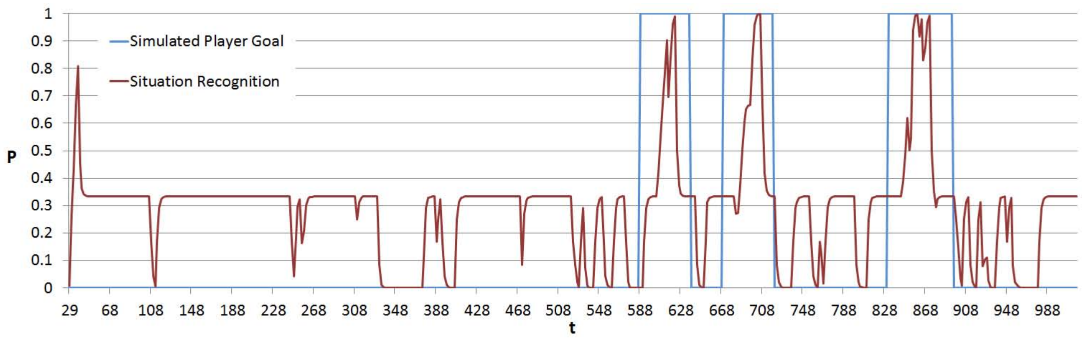

Figure 44: Sample situation recognition of *Searching Bottle* situation (with simulated players)

Building the hut is recognized correctly over 57% (metric 2) of the situation's active time with 3% of false positive recognitions (metric 3).

Talking to Hank is recognized during 58% of the situation's duration on average (metric 2) with only 1% of false positive recognition (metric 3). All talking situations are recognized. Yet only 38% of the time simulated players were about to talk to the NPC, it could be recognized. For real players, 90% of the time was recognized correctly.

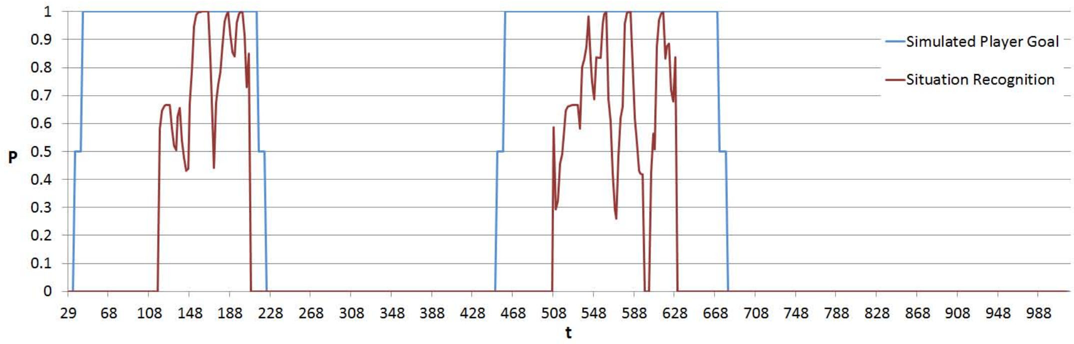

Figure 45: Sample situation recognition of *Hunting Heron* situation (with simulated players)

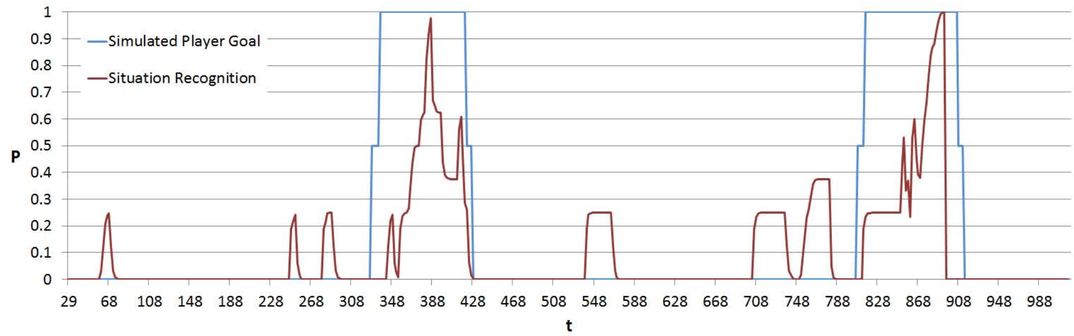

Figure 46: Sample situation recognition of *Build Hut* situation (with simulated players)

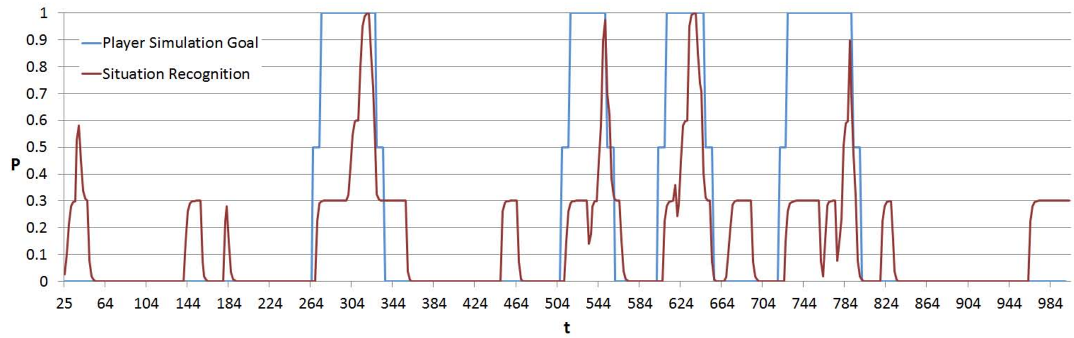

Figure 47: Sample situation recognition of *Talk To Hank* situation (with simulated players)

INTERPRETATION AND DISCUSSION

The false positive recognition of the *Searching For Berries* situation usually results from players staying near berry bushes for longer periods without actually searching for berries (see Figure 43).

Regarding the *Searching For Bottle* situation, the false positives can be explained with the importance of the *RegionCriterion:Beach* which results in recognition of this situation whenever players walk along the beach without actually searching for the bottle. Similar it exacerbates recognition of the situation while players have not yet reached the beach but are already searching for the bottle (compare Figure 44 for a typical recognition pattern of the *Searching For Bottle* situation). The real players always started searching for the bottle from the beginning of the game, i.e., directly from the beach. Hence, the accuracy is higher in that scenario.

Summarizing, it can be stated that the situation recognition is very well able to detect what players are doing in EFWI and why. However, the quality of the situation recognition strongly depends on the type of related criteria used to describe the situation. There is an inherent delay in almost all of those situations, which in most cases is caused by the fact that players first need to walk towards a certain spot in the level. While they are moving there, it is often unclear to what purpose they are doing it. Hence the recognition is inaccurate at that time.

Looking at Figure 45, one can see that about half of the time players spent on hunting the heron is not recognized correctly. The biggest problem here is that players often need to walk a big distance from where they are to near the heron. Only after some time it is recognized where they are heading and why. Also, the time span after the hunt to chopping meat from the heron is not recognized well.

The recognition of the *Building Hut* situation happens mainly when players are carrying a palm or are walking towards one to carry it. However, the situation recognition does not recognize when a player is felling a palm to build the hut. As the player might fell it for various other reasons, this can be considered as correct. Yet, it explains the delay in recognizing the situation for the player felling the palm (see Figure 46).

Looking at the *Talking to Hank* situation, the problem is that a major part of a player's intention to talk to the NPC ($> 95\%$) is the process of walking towards him. This part is recognized with a low accuracy, as walking could imply a lot of other intentions. Only if the player is moving towards Hank consistently (e.g. for more than 5 seconds), or is close to Hank, the situation recognition considers it to be on purpose and not on chance. Hence, the situation is recognized mostly during the second half of its active time (see Figure 47). In the scenarios with real players, on average they moved shorter distances towards the NPC to talk to it.

In summary, the situation recognition is able to recognize all situations at least once each time they occur. For all situations, the recognition ratio (the percentage of time the situation is recognized while it is active) is $\geqslant 48\%$. For four of the five situations, the ratio of false positive recognition is $\leqslant 3\%$, only the *Searching for Berries* situation is falsely recognized in 25% of the game time.

9.3 EVALUATION OF GAME ADAPTATION SYSTEM PARAMETERS

In this section the evaluation of the system parameters of the adaptation engine is described. The main purpose of this evaluation is to understand the designed adaptation mechanism and the influence of changing systems parameters on the adaptation effectiveness.

9.3.1 *Experiment Design and Setup*

The changeable system parameters are the weighting parameters α, β, γ, and δ for the adaptation selection (see Section 5.7). As EFWI does not offer enough ways of adapting the game in terms of gameplay, there were no adaptations designed for adapting the game in terms of gaming preferences- Hence, all adaptations and adaptation objects as defined in Appendix A focus on adapting challenge, learning skills, or interaction skills. Consequently, the parameter α cannot influence the selection of adaptations. Therefore, α is not considered within this experiment. A $2^k \cdot r$ factorial design is performed to evaluate the impact of the three parameters β, γ, and δ on the system with $k = 3$ and $r = 5$.

For the conduction of the evaluation, a series of measurements was performed. For a test instance, a server was started in the simulation mode. Four player instances were started and joined the game (as simulated players). Standardized simulated players were used for all the simulation runs. The simulated players had all skills set to 0 and a 'neutral' player model (i.e., $(0.5, 0.5, 0.5, 0.5, 0.5)$).

The adaptation engine was configured using the following settings: The adaptation objects were used as defined in Chapter 8 (Tables 54, 55, and 56) using the game elements, information objects, game actions, shapes, tasks, situations, and situation objects as defined in Appendix A. α is always set to 0. For the case $\alpha = \beta = \gamma = \delta = 0$, no adaptations are selected and executed.

The response values are the

- learning skills progress,
- teamwork and collaboration skills progress,
- average challenge values throughout the game,
- player performance values,

after a game session of 2730 seconds (45 minutes play time + 30 seconds start-up).

The player learning skills, as well as the teamwork and collaboration are taken from the player agent models. Those are logged in a 1-second interval. (For the process of 'learning' of virtual player agents see Section 8.7). It is of interest, at what point during the game, the skills are learnt, i.e., at what point their values become greater than 0.8. The average challenge is taken from the adaptation engine. For each challenge value, the average over the 45 minutes is calculated from logged data whereas the challenge values are logged in a 1-second interval.

9.3.2 *Results and Discussion*

A PRIORI

The effect on the player performance, the average challenge values over the course of the game, and the average game time for a skill to be learnt, are shown in Appendix B. Table 63 contains the influence of the three parameters β, γ, and δ on the variance of the mean response values of the player performance values and Table 64 shows the absolute mean responses of the player performance values.

The influence of the three parameters β, γ, and δ on the variance of the mean response values of the average challenges is shown Table 65. Table 66 contains the absolute mean responses of the average challenges.

The influence of the three parameters β, γ, and δ on the variance of the mean response values of the average time to learn a skill is shown in Table 67. Table 68 contains the absolute mean responses of the of the average time to learn a skill.

The influence of β on the proportion of variation of the system response of the player performance is notable in the number of berries gathered (9%), lowering the mean number of berries gathered (89.60) about 8.45. For the number of herons hunted it is (18%, 4.68 on average, 5.40 with $\beta = 0$, 3.95 with $\beta = 1$). β further has an impact on the challenges *KeepSatietyHigh* (30%, lowering the average of -0.25 about 0.12), as well as a minor impact on *BuildLogHut* (15%), lowering the average challenge of 0.58 about 0.08, and on *FindBottle* (16%, -0.93 for $\beta = 0$ and -0.91 for $\beta = 1$). β has a visible effect on the skills *BuildHut* (27%, 895s for $\beta = 0$ and 344s for $\beta = 1$), and a negative effect on *Communication* (18%, 1630s for $\beta = 0$ and 2423s for $\beta = 1$). For all other skills, the effect is $\leqslant 7\%$.

It can be seen that γ has a major influence ($0.28 - 0.78$) on all player performance values, except for the number of herons hunted and the average time to fill the bottle. γ also has a major influence on the average challenges for *KeepFitnessHigh* (53%, 0.76 for $\gamma = 0$ and 0.20 for $\gamma = 1$), *KeepHealthHigh* (66%, -0.05 for $\gamma = 0$ and -0.48 for $\gamma = 1$), *BuildLogHut* (58%, 0.74 for $\gamma = 0$ and 0.42 for $\gamma = 1$), *BuildRaft* (93%, 0.23 for $\gamma = 0$ and -0.78 for $\gamma = 1$), and *FindBottle* (46%, -0.93 for $\gamma = 0$ and -0.90 for $\gamma = 1$). Further, γ has a major positive influence ($0.32 - 0.89$) on the average time to learn a skill for all skills except *KnowledgeBottle* (0.00), *SleepInHut* (0.01), and *Communication* (0.00).

δ does not have an visible (i.e., $\geqslant 0.05$) effect on player performance values. It also does not notably influence the average challenge values. δ does have an effect on the average time to learn the *Teamwork* (21%, 1490s for $\delta = 0$ and 647s for $\delta = 1$) and *Communication* (27%, 2510s for $\delta = 0$ and 1543s for $\delta = 1$) skills, but not on any other skills ($\leqslant 0.05$).

INTERPRETATION AND DISCUSSION

Looking at the absolute changes of the mean values, it can be seen that a change of β from 0 to 1 results in a change of the mean value towards the overall mean value for the player performance, challenge value, or time to acquire a skill. This indicates that β reacts to deviations of player performance (and resulting over- or under challenge and time to acquire skills). Hence, it can be concluded that β mainly influences how high and low challenge is addressed during adaptation selection. However, it also prioritizes adaptation of skills, if they are directly influencing challenges (e.g., a high *BuildFire* skills lets players roast meat, directly improving their satiety).

Looking at the absolute influence on the responses, one can see that a change of γ from 0 to 1 results in a strong improvement of the resulting player performance values, as well as a strong tendency to lower overall challenge, which in some cases results in tasks to become too easy, i.e., the related challenge becomes negative (e.g., BuildRaft $\gamma = 0 : 0.23$; $\gamma = 1 : -0.78$). Similar, γ strongly lowers the mean time to acquire a skill throughout all skills observed. Summarizing, it can be stated that γ has a major influence on both the average challenges and the average time to learn a skill, resulting in a major influence on most of the player performance values.

Looking at the mean values, it can be seen that a focus on interaction ($\delta = 1$) does actually have a negative impact on the time to learn various skills, as well as on some challenge values, and on player performance values.

Concluding, it can be stated that the learning part (γ) of the adaptation selection metric has the greatest influence on the average challenge, on the average time to learn skills, and on the resulting player performance. The challenge part (β) does have a visible influence on some challenges and on some learning skills, as well as on the *Communication* skill. Its impact on the resulting player performance, however, is only minor. The interaction part (δ) does only positively effect the average time to acquire the interaction skills *Teamwork* and *Communication* and does in fact impair the learning times of other skills, the average challenges and the player performance.

Hence, it is decided to to set $\alpha = 0$, and $\beta = \gamma = \delta = 0.33$ for further evaluations. This ensures the positive effects on learning and challenge through β and γ, as well as the positive effect on the interaction skills caused by δ.

9.4 AUTOMATIC GAME ADAPTATION EFFECTIVENESS WITH SIMULATED PLAYERS

Using the simulated players, the effectiveness of the automatic game adaptation will be evaluated. The goal of this evaluation is to find out to which extend the automatic game adaptation can (positively) influence the performance of the (simulated) players. Simulated players are used for two reasons. Firstly, it enables to have players with controlled player, learner, and interaction models. This would not be possible with real players. Although it is possible to find out a player's player model through the use of questionnaires, there is only a limited accurateness. Moreover, knowing a player's player model does not help when certain player models are intended to be used. For this, it would be necessary to have a huge amount of players with known player models if which players with a desired player model could be chosen. Secondly, using simulated players, it is possible to run a multitude of repeatable simulation runs with controlled input parameters without the need of a large amount of human players.

9.4.1 *Experiment Design and Setup*

The purpose of this evaluation is to show the impact of the use of the game adaptation mechanism on the performance of the (simulated) players. Therefore, a set of player agents with varying player, learner, and interaction models are generated and used in the simulation runs. System parameters are fixed using the results from Section 9.4. In the reference group, the same sets of player agents play the game with disabled adaptation engine. Observed performance parameters are the ones described in the first evaluation in Section 9.1.

For the conduction of the evaluation, a series of measurements was performed. For a test instance, a server was started in the simulation mode. Four player instances were started and joined the game (as simulated players). The player model, learner model, and interaction model was set as shown in Table 30. The adaptation engine was configured using the following settings: The adaptation objects (see Tables 54, 55, and 56 in Appendix A) were used as defined in Chapter 8 using the game elements, information objects, game actions, shapes, tasks, situations, and situation objects as defined in Appendix A.

Table 30 shows the 6 configurations of the simulated players. Sets 1 to 3 do not use any adaptations, whereas sets 4 to 6 do use the adaptation engine. In sets 1 and 4, the skills are set to 0.0, in sets 2 and 5 the skills are set to 0.5 and in sets 3 and 6 the skills are set to 1.0. The adaptation engine is used with the following adaptation selection parameters: $\alpha = 0$, $\beta = \gamma = \delta = 0.33$. A run was started and ran until players built the raft. At this point, the simulation was stopped. Game data was logged (compare Section 9.1).

Set	Adaptation	Player Model	Learner Model	Interaction Model
1	not activated	$< 0.5, 0.5, 0.5, 0.5, 0.5 >$	all 0.0	<0.0,0.0>
2	not activated	$< 0.5, 0.5, 0.5, 0.5, 0.5 >$	all 0.5	<0.5,0.5>
3	not activated	$< 0.5, 0.5, 0.5, 0.5, 0.5 >$	all 1.0	<1.0,1.0>
4	activated	$< 0.5, 0.5, 0.5, 0.5, 0.5 >$	all 0.0	<0.0,0.0>
5	activated	$< 0.5, 0.5, 0.5, 0.5, 0.5 >$	all 0.5	<0.5,0.5>
6	activated	$< 0.5, 0.5, 0.5, 0.5, 0.5 >$	all 1.0	<1.0,1.0>

Table 30: Setup configuration for the automatic game adaptation.

9.4.2 *Results and Discussion*

A PRIORI

The effect on the player performance, the average challenge values over the course of the game, and the average game time for a skill to be learnt are shown in Appendix B. In Table 69, the mean values and standard deviation of the player performance values are shown for the groups with adaptation and the reference groups without adaptation. Table 70 contains the mean values and standard deviation of the challenge values for the groups with adaptation and the reference groups without adaptation and Table 71 shows the mean values and standard deviation of the average time needed to learn a skill for the groups with adaptation and the reference groups without adaptation. Following, the most significant results are pointed out and discussed.

Figure 48 shows the comparison of challenge for the three skill configurations 0.0, 0.5, and 1.0 between the groups with adaptation and the groups without adaptation. In the groups without adaptation, the average *KeepFitnessHigh* challenge among the groups with no knowledge/skills (skills $= 0.0$) was 0.82 ± 0.06 and among the groups with skills $= 0.5$, it was 0.54 ± 0.25. The respective values in the groups with adaptation were 0.17 ± 0.10 and 0.22 ± 0.15. There is no big difference between the groups with and without adaptation for the sets with skills set to 1.0. For the *BuildLogHut* challenge, a similar observation can be made: 0.81 ± 0.01 / 0.79 ± 0.01 for the groups without adaptation and 0.30 ± 0.07 / 0.26 ± 0.12 for the groups with adaptation. Here, the challenge of the groups with initial high knowledge (skills set to 1.0) could be improved from 0.32 ± 0.30 to 0.03 ± 0.11.

Figure 49 shows the comparison of skill development for the three skill configurations 0.0, 0.5, and 1.0 between the groups with adaptation and the groups without adaptation. Looking at the skill development in Table 71, one can see that for all of the 18 skills, the adaptation mechanism did greatly reduce the average time to acquire the skills both for the settings with skills $= 0.0$ and skills $= 0.5$. For the settings where players already knew everything about the game (skills = 1.0), there was of course no change, as they trivially had acquired all skills already.

Figure 50 shows the comparison of player performance values for the three skill configurations 0.0, 0.5, and 1.0 between the groups with adaptation and the groups without adaptation. The average fitness is improved from 24.50 ± 14.01 to 43.38 ± 7.95

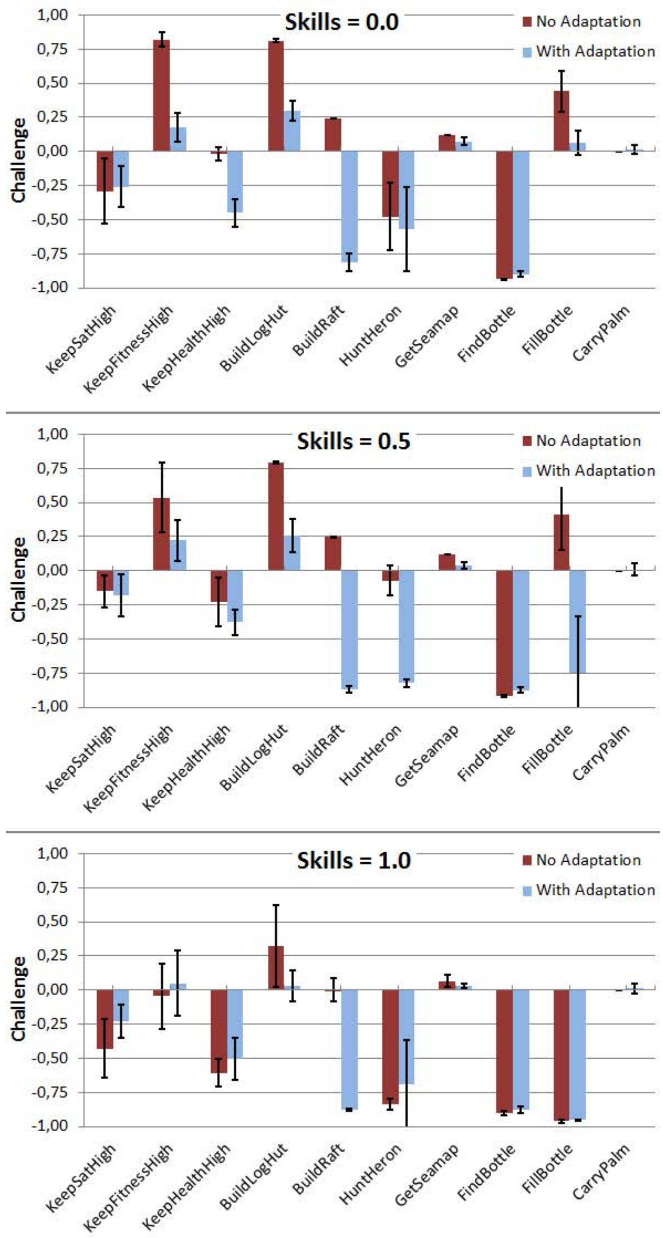

Figure 48: Average challenge comparison for the three skill configurations 0.0, 0.5, and 1.0 between the groups with adaptation (blue) and the groups without adaptation (red).

and the number of berries gathered is reduced from 118.60 ± 21.73 to 49.20 ± 6.83. The number of meat roasted is almost doubled (from 5.80 ± 5.59 to 10.60 ± 1.34) and the time required to build the hut (from 1336 ± 849s to 743 ± 153s), to find the bottle (from 1300 ± 398s to 138 ± 93s), to fill the bottle (from 1505 ± 330s to 616 ± 303s) were greatly reduced. Also, the adaptation engine helped to enable players to build the raft in all cases whereas this was not possible for players with initials skills of 0.5. The effects are even greater when comparing the groups with skills set to 0.0. Moreover, here an improvement in the overall health (from 44.13 ± 0.89 to 80.79 ± 13.19) and satiety (from 45.82 ± 5.37 to 54.57 ± 3.34) could be observed.

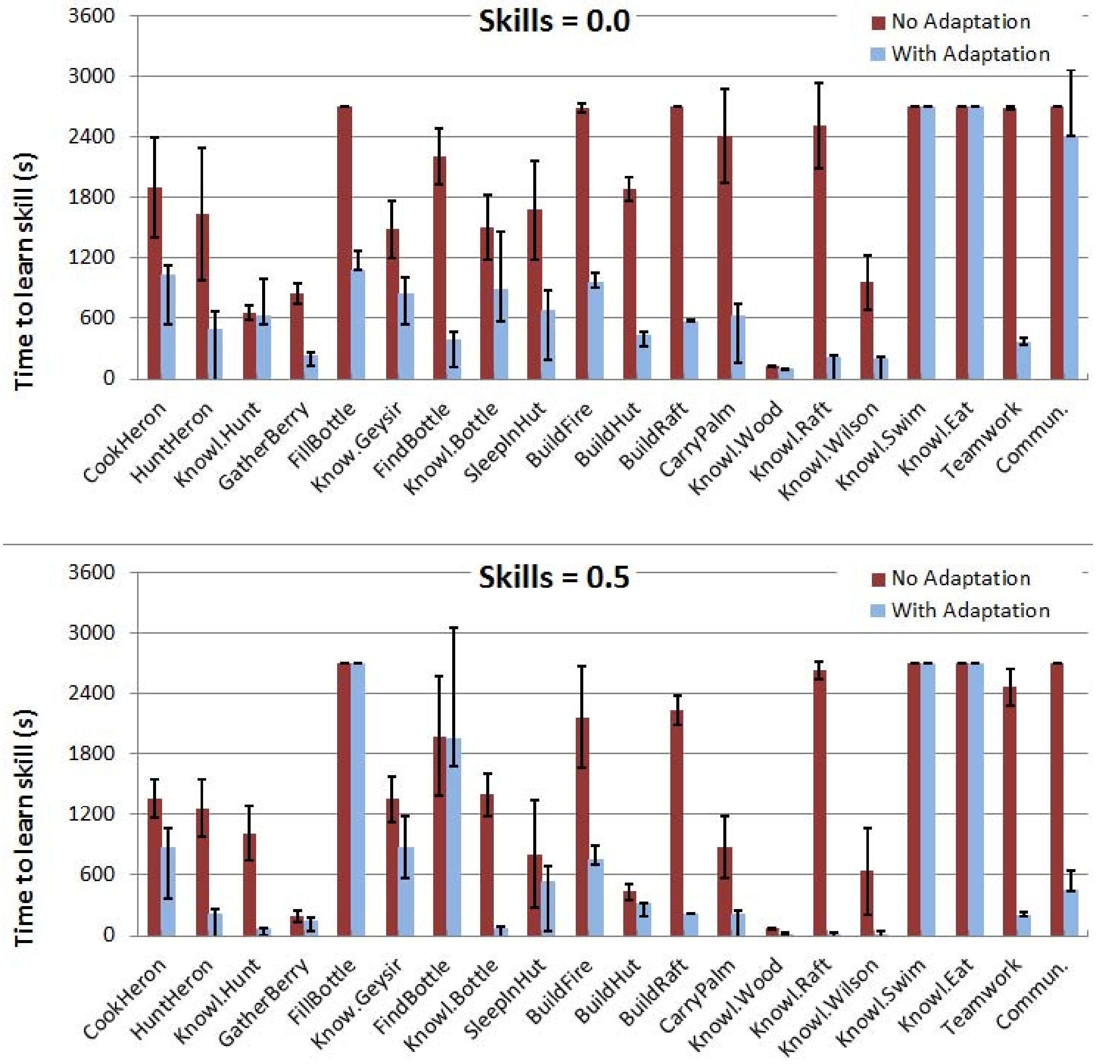

Figure 49: Average time to learn skills for the skill configurations 0.0 and 0.5 between the groups with adaptation (blue) and the groups without adaptation (red).

interpretation and discussion

Comparing the average challenges throughout the six sets, one can see that the adaptation mechanism was able to eliminate high over-challenge. Hence, challenge was much better for those groups. However, not all challenges could be improved and in some cases the adaptation mechanism reduced the challenge too strongly, e.g., the *BuildRaft* challenge varies between -0.81 and -0.88 in the settings with adaptation. This is also reflected in a shortened time needed to build the raft for those groups. Altogether, it can be stated that the adaptation mechanism was able to reduce the challenge in some cases where it was necessary, but made it too easy on some other cases. Here, the respective adaptation objects need to be improved.

The skill development time was greatly improved through the adaptation mechanism for all 18 skills for the settings with skills set to 0.0 and 0.5.

Looking at the resulting player performance values, the simulated players playing with adaptation performed overall a lot better than the simulated players without adaptation. Whereas there is almost no difference between the two groups when the

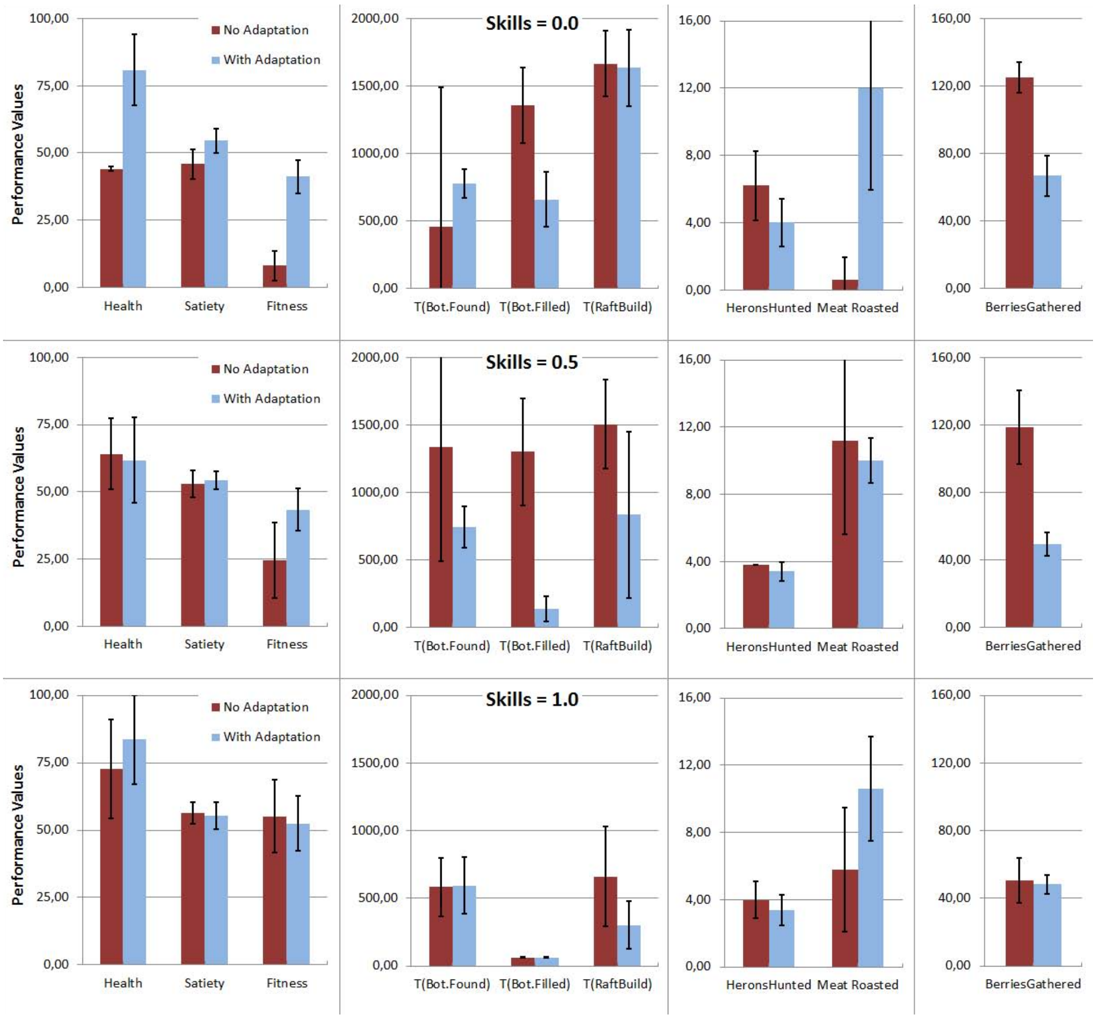

Figure 50: Performance values for the skill configurations 0.0 and 0.5 between the settings with adaptation (blue) and the settings without adaptation (red).

skills were set to 1.0 from the beginning, already in the groups with skills set to 0.5, there are visible improvements when the adaptation mechanism is used.

Altogether, it can be concluded that the adaptation mechanism was able to improve the player performance indicators for the simulated players. Especially, when the simulated players do not start with the required skills (i.e., skills set to < 1.0), it was possible to improve their performance. Moreover, the adaptation mechanism was able to trigger required adaptations such that the simulated players did receive necessary information to improve their performance, which can be seen when looking at the average times to learn skills. Finally, the adaptation mechanism was able to improve the overall challenge of some of the challenge values. However, for others it simply made the game too easy. Here, the underlying adaptation objects need to be revised.

9.5 AUTOMATIC GAME ADAPTATION EFFECTIVENESS WITH REAL PLAYERS

For the sake of an improved validity, the evaluation of the influence of the adaptation mechanism on players was repeated using real players. The goal of this evaluation was to evaluate the effectiveness of GameAdapt.KOM under realistic conditions. Hence, this evaluation directly contributes to Hypothesis II (see Section 1.2.3).

9.5.1 *Experiment Design and Setup*

Referring to Hypothesis II, the dependent variables are learning success, gaming success, and game experience. Learning success can be measured directly from the game. It is the development of the players' skill set during the game. Gained knowledge is hence defined as the difference in skills comparing a player's learner model before the game session with his/her learner model after the game session. Gaming success can be measured directly from game data using player performance.

As collaboration and teamwork is the learning goal of *EFWI*, it is measured using the questionnaire testing for the players' perception of teamwork and communication, as described in Table 80 in Appendix C. For measuring game experience, the game experience questionnaire explained in Table 78 in Appendix C is used. Gaming success is measured based on the group's overall performance.

To determine the player model of a real player, player actions are initialized as described in Section 8.4. Further, player actions are continuously monitored and attributed using the situation recognition shown in Section 5.5. Players then play the game and the adaptation engine evaluates player behavior and performance and selects adaptations to be performed when necessary. Player performance data is gathered as described in Section 9.1. From this data, the player performance is calculated. Hence, the treatment group played the game with a Game Master present. A reference group consisting of a set of player groups plays the game without the adaptation engine to be active. The independent variable (stimulus) is the presence of the adaptation mechanism during the game session. The dependent variable (response) is the group's performance which is calculated from the collected data.

The evaluation was performed between October 2014 and January 2015. 40 participants were selected and randomly assigned into groups of four players. Thus, ten groups played the game. Of those ten groups, five were randomly selected for the treatment group, and five for the reference group. Only players which had not played the game before were selected.

All players received an instruction about the game and their goal. This included an instruction about game controls and graphical user interface.

A server was started and four clients were started and connected to the server. Once all players were ready, the game session was started and played either until the game was won or 45 minutes of play-time were reached, whichever happened first. After the game, the players were asked to fill out the two questionnaires.

9.5.2 *Results and Discussion*

A score of the *quality of the subjective experience* was built from the data of the game experience questionnaire. Therefore the mean of the items of the questionnaire were calculated for each of the categories negative emotion, positive emotion, cognitive load, motivation, immersion, flow, and arousal. For statistical testing of the hypothesis, a two-sided ANOVA with (all) 40 participants was used, testing the effect of the presence of the adaptation mechanism on the perceived teamwork.

Further, a score of the *teamwork and interaction* was built from the data of the teamwork questionnaire. Therefore the mean of the items of the questionnaire were calculated for each of the single tasks and the whole playtime. For statistical testing of the hypothesis a two-sided ANOVA with (all) 40 participants was used, testing the effect of the presence of the adaptation mechanism on the game experience.

Performance data was aggregated as sum for discrete elements (e.g., gathered berries, herons hunted) or as mean for continuous measurements (health, satiety, fitness) over the 45 minutes playtime.

A PRIORI

Table 73 in Appendix B shows the perceived game experience among the two groups. Figure 51 illustrates Table 73.

One can see that players perceived slightly less *negative emotion* in the setting with adaptation ($M = 7.27, SD = 1.72$) than in the setting without adaptation ($M = 6.22, SD = 1.96$). (Note: as the questions on this category were coded inversely, a high value in this category actually means that less negative experience was perceived). In all of the other six categories, the players in the setting with adaptation rated significantly better than the players which played without adaptation. The *cognitive load* was perceived with ($M = 7.12, SD = 1.39$) by the groups with adaptation compared to ($M = 4.30, SD = 2.21$) at the groups without adaptation. *Positive emotion* was rated ($M = 7.12, SD = 1.52$) by groups with adaptation and ($M = 4.92, SD = 1.78$) by groups without adaptation. Groups that played with adaptation showed a stronger *motivation* ($M = 6.95, SD = 1.76$) than groups without adaptation ($M = 4.15, SD = 1.34$). Likewise, groups with adaptation ($M = 5.92, SD = 1.47$) had a better *immersion* than by groups without adaptation ($M = 3.50, SD = 1.77$). *Flow* was rated strongest of all categories both by groups with adaptation ($M = 7.53, SD = 1.64$) compared to ($M = 5.10, SD = 1.82$) by groups without adaptation. Arousal was also rated higher by groups that played with adaptation ($M = 6.47, SD = 1.81$) compared to groups that played without adaptation ($M = 4.38, SD = 1.53$). Hence, 7 out of 7 User Experience categories were rated significantly better by the players with adaptation compared to the players without adaptation.

Table 74 in Appendix B contains the results for the teamwork and communication questionnaire. Figure 55 illustrates Table 74.

Looking at the *Hunting Heron* task, the *group collaboration* was rated better by groups with adaptation ($M = 4.54, SD = 0.70$) than by groups in the setting without adaptation ($M = 3.64, SD = 0.74$). Likewise the *task solving* rated better when adaptation was used ($M = 4.08, SD = 0.87$) than when adaptation was not used ($M = 3.23, SD = 0.76$). Also, the *perception of the group performance* was better in the groups with adaptation ($M = 4.38, SD = 0.67$) compared to the groups without adaptation ($M = 3.26, SD = 0.70$). *Flow experience* was perceived slightly better in the groups with adaptation ($M = 4.17, SD = 0.95$) than in the groups without adaptation ($M = 3.68, SF = 0.95$). Similar ratings can be observed for the tasks *Carrying Palm*, *Steering Raft*, and for the overall rating. Altogether, for all three tasks and the game itself, the *group collaboration*, the *task solving*, the *perception of the group performance*, and *flow experience* were rated clearly better by teams where adaptation was enabled than by teams which played without adaptation.

Table 72 contains the performance data of the groups. Figure 53 illustrates Table 72.

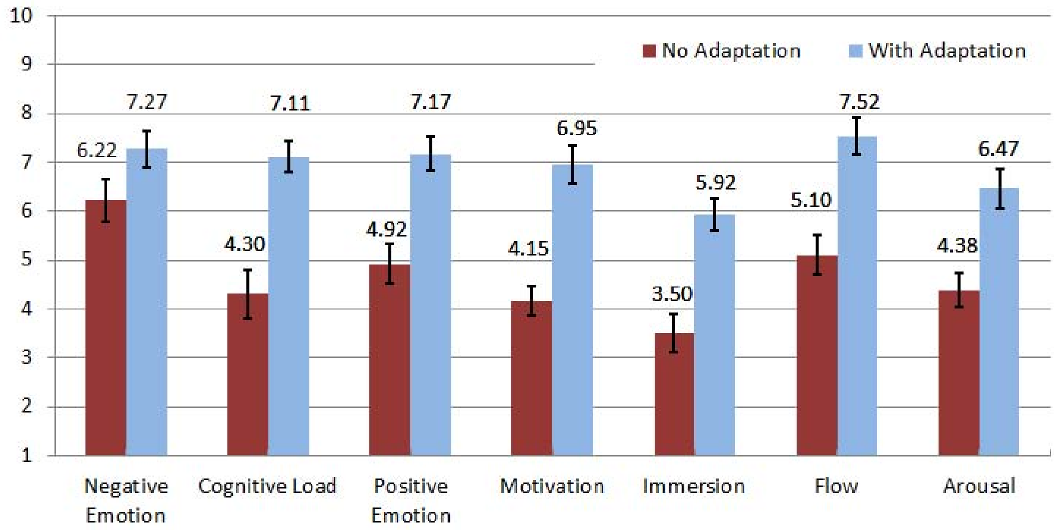

Figure 51: Game experience values for the 7 dimensions of the questionnaire with 95% confidence intervals

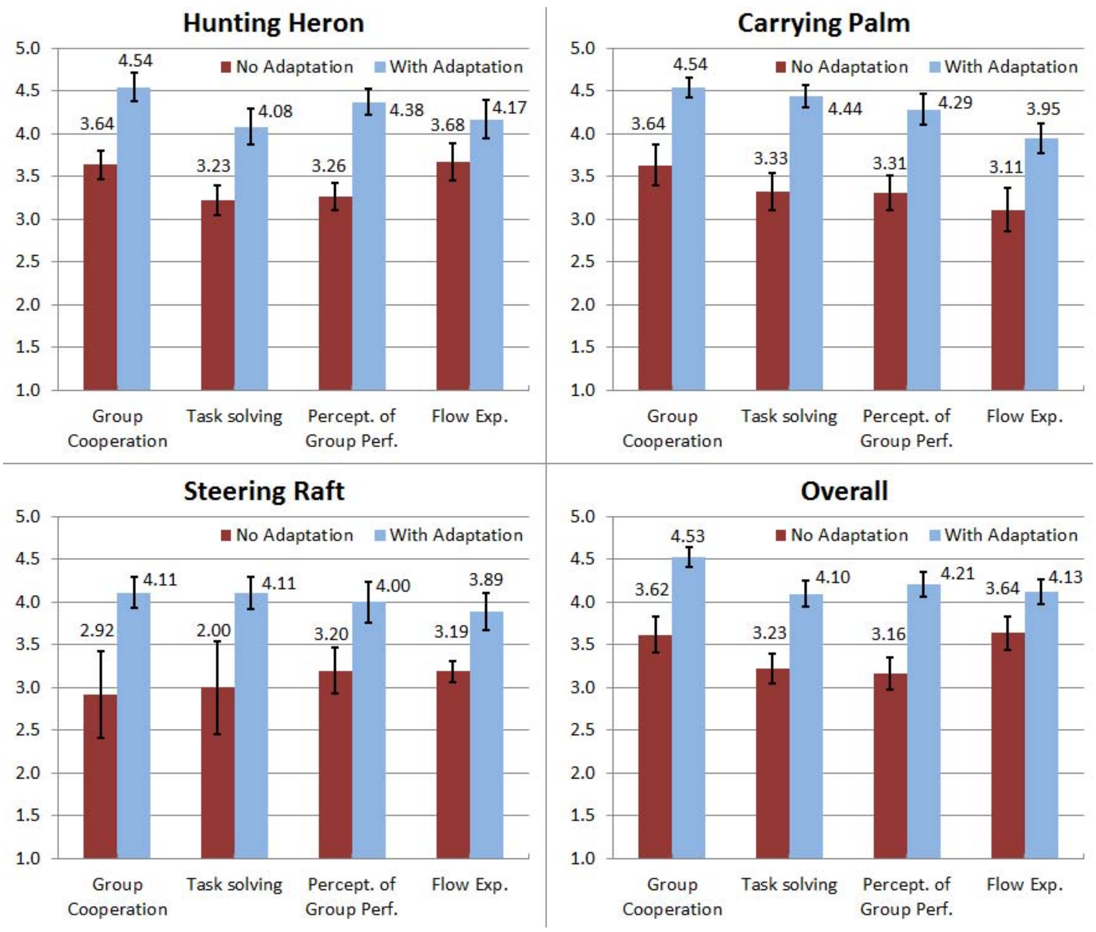

Figure 52: Teamwork and collaboration values for the 3 collaborative tasks and the overall game of the questionnaire with 95% confidence intervals

Looking at the player performance values, one can see from the setting with enabled adaptation, the average health ($M = 68.83, SD = 13.07$ vs. $M = 44.93, SD =$

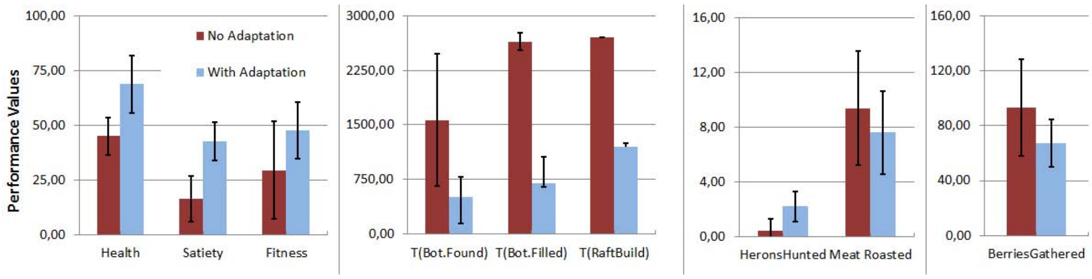

Figure 53: Performance values comparison between the groups with adaptation (blue) and the groups without adaptation (red).

8.75), satiety ($M = 42.74, SD = 8.75$ vs. $M = 16.36, SD = 10.48$), and fitness ($M = 47.54, SD = 12.91$ vs. $M = 29.37, SD = 22.34$) is significantly higher than in the setting without adaptation. Also, on average, players gathered less berries ($M = 67.20, SD = 17.43$ vs. $M = 93.20, SD = 35.26$), hunted more herons ($M = 2.20, SD = 1.10$ vs. $M = 0.40, SD = 0.89$), and roasted more meat ($M = 9.40, SD = 4.16$ vs. $M = 7.60, SD = 3.05$). The times to build the hut ($M = 508.80, SD = 268.54$ vs. $M = 1566.20, SD = 912.60$) and to find ($M = 694, SD = 362.62$ vs. $M = 2648.20, SD = 115.83$) and fill ($M = 1196.80, SD = 49.18$ vs. $M = 2700, SD = 0.00$) the bottle are also significantly lower when automatic adaptation was enabled as compared to when it was not enabled. In the settings with adaptation, all groups were able to build the raft within the 45 minutes of play time, whereas none of the teams where able when no adaptation was performed.

INTERPRETATION AND DISCUSSION

The use of the adaptation mechanism appears to have a positive influence on the game experience of the players, their perception of their own and the group's teamwork, as well as on their actual performance in the game. From the improved game experience, it can be concluded that players did indeed have more fun and less frustration when GameAdapt.KOM adapts the game. From the improved perception of teamwork and collaboration, it can be concluded that either teamwork and collaboration was better among the teams where adaptation was enabled, or that the adapted challenge and support positively influenced the players' perception of the collaborative tasks. The measured performance data shows that players in the settings with adaptation perform better that players without adaptation, which results in a more efficient use of resources and a more goal-oriented play-style resulting in a quicker achievement of goals. Thus, it can be concluded that players did indeed play better in terms of the game's goals, solving the presented tasks better or quicker.

Hence, it can be concluded that GameAdapt.KOM was able to successfully adapt the game in the dimensions learning, gaming, interaction, and challenge, resulting in an improvement of learning success, game experience, and the players' performance with regard to the goal of the game.

9.6 GAME MASTERING EFFECTIVENESS

The evaluation of the effectiveness of Game Mastering targets how well an instructor can make a positive impact on player performance, game experience, and collaboration using the GameAdapt.KOM Game Mastering environment to orchestrate a game session. This evaluation aims at evaluating the usefulness of the Game Mastering concept and framework. It does not aim at evaluating the abilities of the Game Master. Hence, this evaluation directly contributes to Hypothesis I (see Section 1.2.3).

9.6.1 *Experiment Design and Setup*

The experiment design is analogue to the evaluation of the automatic game adaptation effectiveness with real players described above.

Referring to Hypothesis I, the dependent variables (response) are learning success, gaming experience, and performance. As collaboration and teamwork is the learning goal of *EFWI*, it is measured using the questionnaire testing for the players' perception of teamwork and communication, as described above. For measuring game experience, the game experience questionnaire is used Gaming success is again measured through the group's overall performance. The independent variable (stimulus) is the presence of the Game Master. Hence, the treatment group played the game with a Game Master present. In the reference group the set of players played the game without the Game Master.

For the Game Masters' orchestration of the game sessions, it was necessary to be able to interpret GM actions. Hence, Game Masters were instructed to pursue the following goal: Ensure that players were able to win the game by

- helping them understand their tasks,
- helping them improve their teamwork and communication,
- keeping motivation up by keeping challenge at an optimal level.

Game Masters were further introduced into the game mechanics and especially into the Game Master frontend to ensure that they know what GM interface options they can use to achieve a desired effect in the game. This ensures that Game Master actions are standardized preventing Game Masters from distorting the players' performance by e.g. making the game easier than necessary.

The evaluation was performed between November 2014 and January 2015. A total of 40 (32 male, 8 female) participants aged between 16 and 27 ($M = 20.90; SD = 3.05$) were selected and randomly assigned into ten groups of four players. Thus, five groups played the game with a GM and five without a GM. None of the players played the game before.

All players received an instruction about the game and their goal. This included an instruction about game controls and graphical user interface. A server was started with the adaptation mechanism turned off and four clients were started and connected to the server. The Game Master was instructed about his/her tasks and seated in front of the server instance having available the Game Master version of the game using the Game Mastering frontend. Once all players and the GM were ready, the game session was started and played either until the game was won or 45 minutes of play-time were reached, whichever happened first. The Game Master orchestrated the game according to his/her professional opinion based on the instructions beforehand. After the game, the players were asked to fill out the game experience questionnaire and the teamwork questionnaire.

9.6.2 *Results and Discussion*

Analogue to the evaluation of the automatic adaptation mechanism with real players, the data aggregation for the Game Mastering effectiveness is performed. A score of the *quality of the subjective experience* was built from the data of the game experience questionnaire. The mean of the items of the questionnaire were calculated for each of the seven categories *negative emotion*, *positive emotion*, *cognitive load*, *motivation*, *immersion*, *flow*, and *arousal*. For statistical testing of the hypothesis, a two-sided ANOVA with (all) 40 participants was used, testing the effect of the presence of a GM on the game experience.

Further, a score of the *teamwork and interaction* was built from the data of the teamwork questionnaire. For this purpose, the mean values of the items of the questionnaire were calculated for each of the single tasks and the whole playtime. For statistical testing of the hypothesis, a two-sided ANOVA with (all) 40 participants was used, testing the effect of the presence of a GM on the perceived teamwork.

Performance data was aggregated as sum for discrete elements (e.g., gathered berries, herons hunted) or as mean for continuous measurements (health, satiety, fitness) over the 45 minutes of playtime.

A PRIORI

Table 76 in Appendix B shows the perceived game experience among the two groups. Figure 54 illustrates Table 76.

Players perceived slightly less negative emotion in the setting with the GM ($M = 7.22, SD = 1.83$) as compared to the setting without the GM ($M = 6.22, SD = 1.96$). (Note: as the questions on this category were coded inversely, a high value in this category actually means that less negative experience was perceived (see Section C.1). Players perceived a lot more positive emotion in the setting with the GM ($M = 7.83, SD = 2.11$) as compared to the setting without the GM ($M = 4.92, SD = 1.78$). Similar, motivation was rated much better by players in the GM scenario ($M = 7.42, SD = 2.03$) as compared to the setting without the GM ($M = 4.15, SD = 1.34$). Smaller effects can be observed for flow, where players with a GM ($M = 7.08, SD = 2.17$) still report better immersion than players without a GM ($M = 5.10, SD = 1.82$) and for arousal with ($M = 6.40, SD = 2.22$ with GM) and ($M = 4.38, SD = 1.53$ without GM). Immersion is also perceived better by players in the GM setting ($M = 5.27, SD = 2.72$) compared to players where no GM was present ($M = 3.50, SD = 1.77$). The perceived cognitive load does not differ much between the two groups ($M = 4.60, SD = 2.00$ with GM) and ($M = 4.30, SD = 2.21$ without GM).

Hence, 6 out of 7 game experience categories were rated significantly better by the players where a Game Master was present compared to the players where no Game Master was present.

Table 74 in Appendix B contains the results for the teamwork and communication questionnaire. Figure 55 illustrates Table 74.

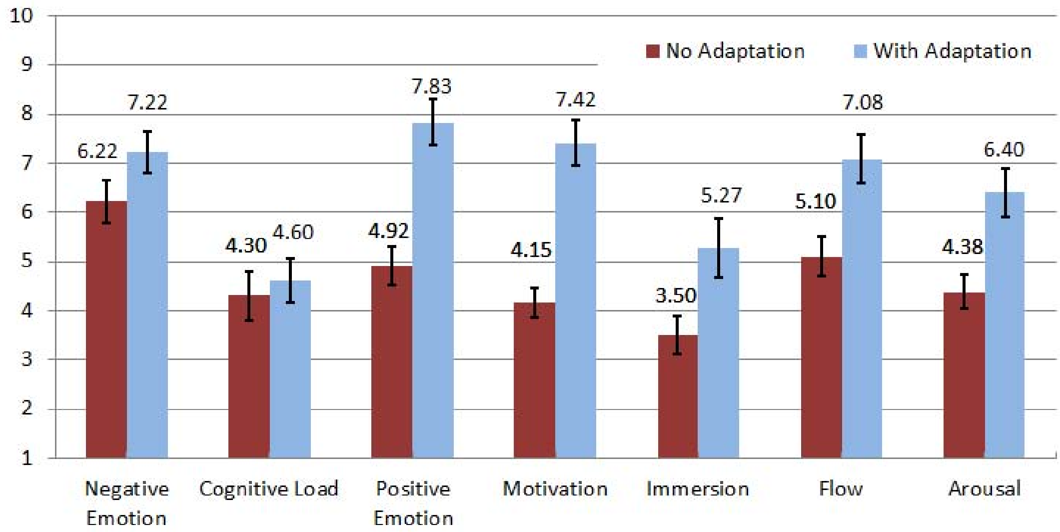

Figure 54: Game experience values for the 7 dimensions of the questionnaire with 95% confidence intervals.

For the *Hunting Heron* task, players rated the collaboration on average higher in the setting with the GM ($M = 4.43, SD = 0.61$) compared to the setting without a GM ($M = 3.64, SD = 0.74$). Similarly, the perception of the own competence was rated higher with GM ($M = 4.16, SD = 0.85$) compared to the setting without the GM ($M = 3.23, SD = 0.76$). Also, the group competence was perceived better in the scenario with GM ($M = 4.43, SD = 0.61$) than in the scenario without the GM ($M = 3.26, SD = 0.70$). Flow is rated slightly better on average in the scenario with GM ($M = 4.04, SD = 0.94$) compared to the scenario without the GM ($M = 3.68, SD = 0.95$).

Regarding the *Carrying Palm* task, players rated the collaboration on average higher in the setting with the GM ($M = 4.60, SD = 0.64$) compared to the setting without a GM ($M = 3.64, SD = 1.08$). Similarly, the perception of the own competence was rated higher with GM ($M = 4.64, SD = 0.45$) compared to the setting without the GM ($M = 3.33, SD = 0.98$). Also, the group competence was perceived better in the scenario with GM ($M = 4.64, SD = 0.47$) than in the scenario without the GM ($M = 3.31, SD = 0.91$). Flow is rated better on average in the scenario with GM ($M = 4.12, SD = 0.80$) compared to the scenario without the GM ($M = 3.11, SD = 1.15$).

Regarding the *Steering Raft* task, players rated the collaboration on average higher in the setting with the GM ($M = 4.13, SD = 0.45$) compared to the setting without a GM ($M = 2.65, SD = 1.11$). Similarly, the perception of the own competence was rated higher with GM ($M = 4.21, SD = 0.73$) compared to the setting without the GM ($M = 2.69, SD = 1.14$). Also, the group competence was perceived better in the scenario with GM ($M = 4.21, SD = 0.53$) than in the scenario without the GM ($M = 2.94, SD = 0.13$). Flow is rated better on average in the scenario with GM ($M = 4.25, SD = 0.75$) compared to the scenario without the GM ($M = 3.19, SD = 0.24$).

Overall, players rated the collaboration on average higher in the setting with the GM ($M = 4.45, SD = 0.69$) compared to the setting without a GM ($M = 3.62, SD = 0.93$). Similarly, the perception of the own competence was rated higher with GM ($M = 4.23, SD = 0.86$) compared to the setting without the GM ($M = 3.23, SD =$

0.79). Also, the group competence was perceived better in the scenario with GM ($M = 4.32, SD = 0.70$) than in the scenario without the GM ($M = 3.16, SD = 0.83$). Flow is rated better on average in the scenario with GM ($M = 4.18, SD = 0.76$) compared to the scenario without the GM ($M = 3.64, SD = 0.88$).

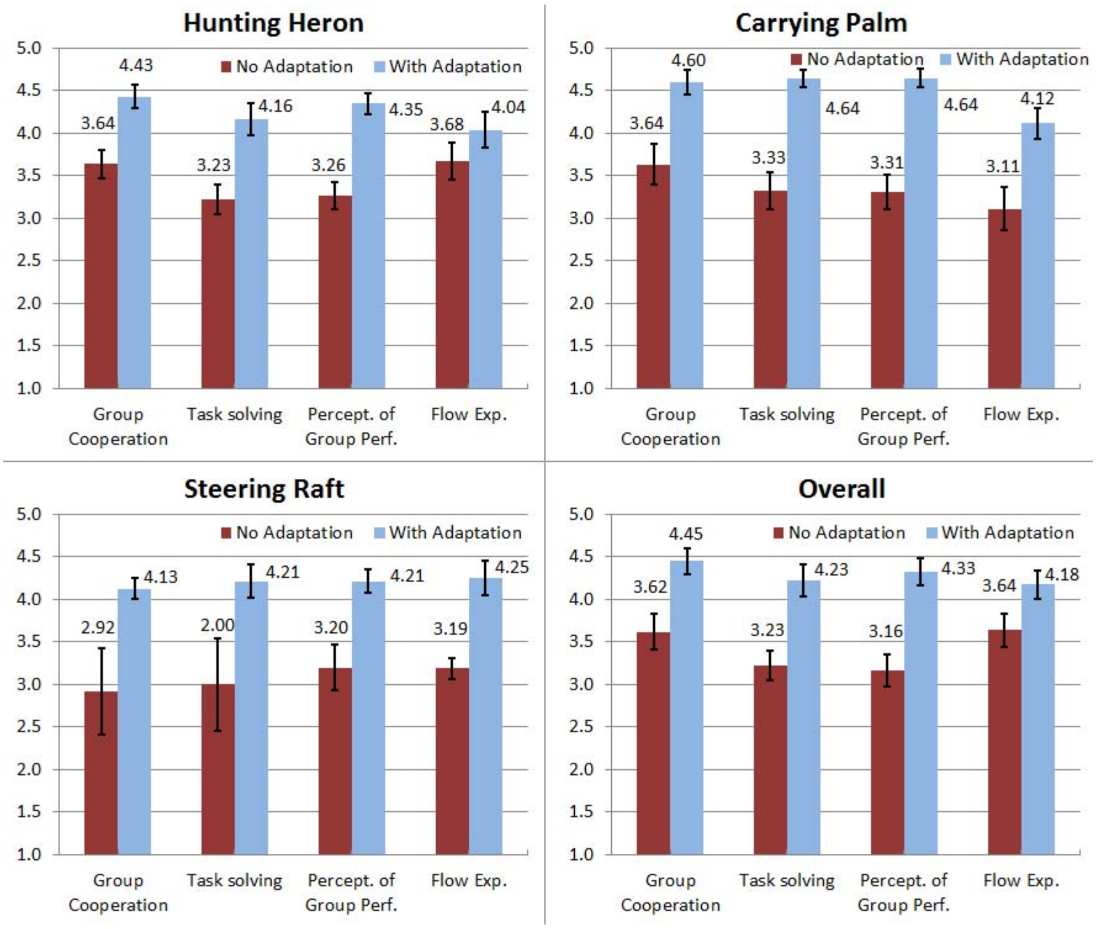

Figure 55: Teamwork and collaboration values for the 3 collaborative tasks and the overall game of the questionnaire with 95% confidence intervals

Table 75 in Section C.1 contains the performance data of the groups. Figure 56 illustrates Table 75. The presentation in form of a table was chosen, as a presentation

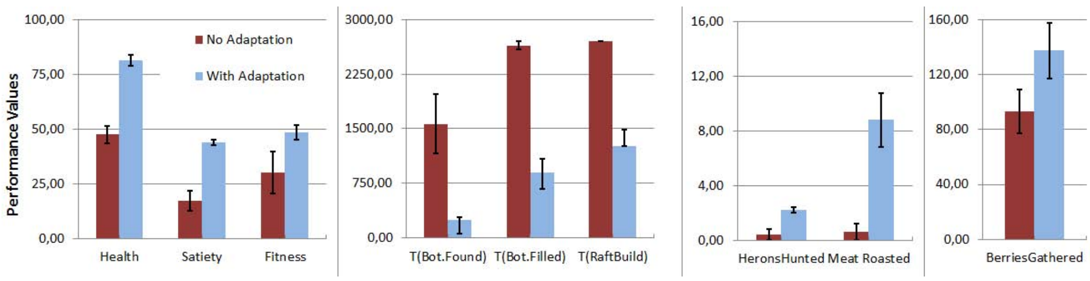

Figure 56: Performance values comparison between the groups with adaptation (blue) and the groups without adaptation (red).

of the data in form of plots was considered of limited usefulness due to the highly varying domains of the parameters observed. Hence, comparing the average values with 95% confidence interval would be very difficult in a graphical presentation. Instead, each response value will be regarded on its own. Looking at Table 75, one can see that in the setting with a GM the average health, satiety, and fitness values are significantly higher than in the setting without a GM. Also, on average players gathered more berries, hunted more herons, and roasted more meat. The number of berries gathered, herons hunted, and meat roasted on average is higher in the setting with a GM than without a GM. The times to build the hut and to find and fill the bottle are also significantly lower when a GM was adapting the game as compared to when no GM was present. In the settings with a GM, all groups were able to build the raft within the 45 minutes of play time, whereas none of the teams where no GM was adapting the game was able to build the raft.

INTERPRETATION AND DISCUSSION

The presence of the GM using the Game Master frontend appears to have a positive influence on the game experience of the players, their perception of their own and the group's teamwork, as well as on their actual performance in the game. From the improved game experience, it can be concluded that players did indeed have more fun, less frustration, etc. when a GM adapts the game, balancing challenge and helping with problems. From the improved perception of teamwork and collaboration, it can be concluded that either teamwork and collaboration was better among the teams where a GM adapted the game, or that the adapted challenge and support through the GM positively influenced the players' perception of the collaborative tasks. The measured performance data shows that players in the settings with a GM perform better that players where no GM is present, which results in a more efficient use of resources and a more goal-oriented play-style resulting in a quicker achievement of goals. Thus, it can be concluded that players did indeed play better in terms of the game's goals, solving the presented tasks better or quicker.

Hence, it can be concluded that the GM was able to successfully adapt the game in the dimensions learning, gaming, interaction, and challenge, resulting in an improvement of learning success, game experience, and the players' performance with regard to the goal of the game.

9.7 CHAPTER SUMMARY

This chapter contains the evaluation of the developed methods and concepts. The simulated player model was evaluated for its soundness comparing the expected behavior of the player agents with the observed behavior (Section 9.1). The situation recognition was evaluated measuring the accuracy of recognized simulations compared to the actually present situations - both with simulated players and with real players (Section 9.2). The system parameters for the game adaptation evaluation metric were examined and their influence on the adaptation mechanism's behavior determined using simulated players (Section 9.3). The actual impact of the adaptation mechanism on the skill development of players, their interaction, and their resulting performance was evaluated with a set of simulated players (Section 9.4) and with 5 groups of real players (Section 9.5). Finally, the impact of a Game Master adapting a game was evaluated with another 5 groups of real players (Section 9.6). The results were compared to a control group consisting of 5 player groups. All groups consisted of four players, totaling at 60 participants throughout the three studies with real players (20 participants playing the GM version, 20 playing the adaptation version, and 20 participants playing the non-adapted version as a control group for the two other groups).

The evaluation of the simulated player model concept (see Section 9.1) shows that using the designed model, it is now possible to configure virtual player agents for a collaborative learning game, so that they behave analogue to real players with similar traits and knowledge as simulated. This means that the designed model does enable the user to configure virtual players in terms of player, learner, and interaction models resulting in comprehensible player agent behavior. The evaluation was targeted at measuring player actions and player performance (in terms of game success) in relation to the player, learner, and interaction model configuration. It also revealed the impact of the defined skillset on the simulated players' performance. Moreover, it could be shown that a variation of player models did alter the way in which the simulated players 'played' the game. This was reflected in the order and emphasis of tackled game targets. This makes it possible to perform the evaluation in Section 9.4 with freely configurable player, learner, and interaction models enabling a sound evaluation of the adaptation engine instead of with a limited number of real players whose player, learner, and interaction model are only inaccurately assessable and immutable.

The results of the situation recognition evaluation (Section 9.2) show that the situation recognition is able to recognize all situations at least once each time they occur. Further the recognition ratio (the percentage of time the situation is recognized while it is active) is about 50% for all situations and for four of the five situations, the ratio of false positive recognition is $\leqslant 3\%$. From the visualization of the comparison of the recognized situation with actually present situations, it could further be explained why the situation recognition could recognize some situations better than others and based on that it was reflected on possible improvements.

In Section 9.3 the evaluation of the system parameters of the adaptation mechanism is described. The goal of this evaluation was to investigate the influence of the system parameters on the adaptation engine performance. A series of tests was performed using different configurations of virtual players. The results show that the learner model has the greatest impact on the average challenge values, the average time to learn the various skills, and the resulting player performance values. It further showed the influence of the challenge component an of the interaction component of the adaptation selection metric. Considering the results from this evaluation, it was decided to set the adaptation selection parameters $\beta = \gamma = \delta = 0.33$ and $\alpha = 0$ for the following evaluations.

The results and findings of the effectiveness evaluation of the automatic game adaptation (Section 9.4) reveals that the adaptation engine is able to improve the players' performance (i.e., that the players learned the skills to be taught in the game better) compared to the setting without adaptation engine. Moreover, it showed that the collaborative tasks could be solved better when the game was being adapted than without adaptation engine which is shown by the improvements of average time needed to solve those tasks. The impact on a part of the challenge values was very positive, keeping the respective challenges around 0. However, other challenges were made too easy for the players, which implies the need of a revision of the adaptation objects. Although the results are considered satisfactory and show that the adaptation mechanism works as intended, the validity of the results for real players cannot be guaranteed. To be able to make valid statements about learning effectiveness, user experience, and quality of collaboration, simulated players are not sufficient.

Hence, in Section 9.5, the effectiveness of the automatic adaptation engine es evaluated using real players. A two-factorial design is used with one group playing with an activated adaptation engine and a reference group which does not use the adaptation engine. The results show that players were able to perform better when the adaptation engine was used. This is reflected in an improved overall game performance and a better user experience which is mainly accounted to a reduced amount of frustration or boredom which occur when the game is either too difficult or too easy. The results of the game experience evaluation showed a significant improvement in all 7 game experience categories. The effects on collaboration were visible within the score of the collaborative tasks, but also within the perceived collaboration and teamwork of the players. For all of the three collaborative tasks and for the overall teamwork experience, players rated all four categories *group collaboration*, *task solving*, *perception of the group performance*, and *flow experience* significantly higher when the adaptation mechanism was used compared to when it was not used.

The findings from the Game Master evaluation (Section 9.6) show that the use of the Game Master frontend positively influences player performance, collaboration, and game experience in the games played. This could be shown throughout all groups playing with a GM who used the Game Master frontend to manually control and adapt a game session at run-time. The improved player performance was shown by an improvement of all observed player score values. The results of the game experience evaluation showed a significant improvement in 6 of the 7 user experience categories. The effects on collaboration were visible within the score of the collaborative tasks, but also within the perceived collaboration and teamwork of the players. For all of the three collaborative tasks and for the overall teamwork experience, players rated all four categories *group collaboration*, *task solving*, *perception of the group performance*, and *flow experience* significantly higher when a GM was overseeing the game compared to when no GM was present. The evaluation moreover revealed that the impact of the GM depends on the players' experience and previous knowledge. Game Masters appear to be able to impact the game stronger when the team itself is inexperienced than compared to an experienced team. The findings also revealed that the composition of the team and the presence of leader personalities do have a strong impact on the gameplay, the interaction between players and subsequently the success of the group.

10 CONCLUSIONS AND OUTLOOK

»We don't stop playing because we grow old; we grow old because we stop playing.«

— George Bernard Shaw

The following chapter concludes this thesis. Section 10.1 summarizes the contents of this thesis and presents its major findings, followed by a discussion of the contributions in Section 10.2. Section 10.3 critically reflects on the goals and hypotheses of this thesis. The final Section 10.4 concludes with an outlook and final remarks.

10.1 THESIS SUMMARY

This thesis is motivated by the challenges that have arisen in recent years concerning the fields of learning and gaming and their superposition in the area of Serious Games. The combination of collaborative learning principles and Serious Games concepts creates challenges for both the design of collaborative multiplayer Serious Games and their orchestration, which might be performed by a human instructor or by an algorithmic adaptation of the underlying game.

Hence, in Chapter 1, the research challenges are outlined. Following, the research goal of this thesis is stated, which is the conceptualization and development of a generic approach to human or automatic adaptation of collaborative multiplayer Serious Games, including a thoroughly designed and comprehensive model for this type of game and the players/learners. Two hypotheses are formulated based on this goal concerning the impact of (a) an instructor and (b) an automatic adaptation mechanism on learning and gaming success as well as on the game experience of the group of players/learners. Based on this, four research questions are identified and addressed throughout this thesis.

As a foundation of this work, the underlying scenario is described in Chapter 2. It explains the features of small learning groups and motivates the role of the instructor. Further, the research field of Serious Games is outlined, with a focus on Serious Games application areas, the use of Serious Games (especially in the classroom), and the Serious Games research group at the Multimedia Communications Lab at the Technische Universität Darmstadt.

The related fields of research are thoroughly discussed in Chapter 3. The field of Serious Games design is covered in Section 3.1. State-of-the-art Serious Games design concepts and frameworks are examined with a focus on the Serious Games-related new design challenges, which go beyond the known challenges of game design. With collaboration being a focus of this thesis, the concept of collaborative gaming is investigated. A major part of the research in this field covers the analysis of collaborative (board) games, with the goal of transferring the identified principles and design guidelines to digital games. As there are no standardized procedures for evaluating Serious Games, different concepts and criteria to evaluate Serious Games are considered, with a focus on flow and user experience. As the thesis is focused on collaborative learning scenarios, the foundations and concepts of collaborative learning are highlighted in Section 3.2, including the computer-supported learning concept, the importance of communication in collaborative learning scenarios, the role of the instructor, and concepts to model and assess teamwork. The field of adaptation in games is investigated in Section 3.3, which covers the concept of Flow, player modeling and various known player models, interaction between players, learner modeling (focusing on the Competency-based Knowledge Space Theory), concepts for modeling whole groups in games, and existing adaptation algorithms. The identified gap is motivated afterward based on the analysis of the state of the art in the related fields of this thesis, which are the adaptation of multiplayer Serious Games in collaborative multiplayer scenarios and the question of how to enable an instructor to adapt a game in a meaningful way at run-time. Because it became necessary to design and implement simulated players, concepts for AI and player simulation are explored in Section 3.4. The concept of the utility-based agent was identified as the most promising concept, considering the requirements of the simulated players. With Game Mastering being one of the two core focuses of this thesis, in Section 3.5, the term *Game Mastering* is elaborated on, starting with the origin in the field of pen-and-paper role-play games, continuing with ideas and approaches to transferring the concept into digital games, and finishing by examining state-of-the-art Game Mastering approaches in games.

10.2 CONTRIBUTIONS OF THIS THESIS

The main contributions of this thesis are as follows:

Generic model for collaborative multiplayer Serious Games

The first main contribution, presented in Chapter 4 of this thesis, is the development of a generic model to represent collaborative multiplayer Serious Games. The model is able to describe relevant game elements and their interconnections. Based on this, the model allows the semantics of game elements and game information to be defined and game information and modifiable game elements to be accessed in order to adapt the game. The model further enables the representation of a group of players in terms of their gaming style, knowledge and skills, and perceived challenge and fun (flow). The four parts of the group model are derived from existing models found in the literature and combined to develop an initial integrated player model, which represents a player/learner of a collaborative multiplayer Serious Game with all relevant features. The developed model makes it possible to describe similar learning games formally, so that meaningful situations and adaptations for the respective game can be defined formally without the necessity of integration into the core game. This enables a transferability of the adaptation and Game Mastering concepts developed in this thesis.

Adaptation selection concept for collaborative multiplayer Serious Games

The second major contribution, which is presented in Chapter 5 of this thesis, is the conceptualization of an adaptation selection mechanism - GameAdapt.KOM. GameAdapt.KOM defines an interface to access relevant game elements based on the model developed in Chapter 4 to assess information and to adapt the game. Thus, it allows game situations to be defined based on information taken from the game description of the related game if the game is represented using the generic model. It contains an algorithm to decide which game situations are present based on the current game state and the integrated group model. It further allows adaptations based on the described accessible game elements to be formally described. GameAdapt.KOM operates in three steps. In the first step, present game situations are recognized algorithmically and interpreted. Based on that, the integrated group model is updated. During the second step, it uses the integrated group model and the observed game state to decide which adaptations to execute. Therefore, a specifically designed adaptation selection algorithm filters available adaptations for validity based on the current game state, rates valid adaptations for their suitability regarding the current game state and group model, and selects the most suitable adaptation. Hence, with GameAdapt.KOM it is possible to automatically adapt a collaborative multiplayer Serious Game at run-time based on player performance and behavior, given the game is formally described using the generic model.

Novel concept for Game Mastering in multiplayer Games

The third major contribution is a novel concept for orchestrating digital educational games (presented in Chapter 6). This concept focused on, but is not limited to collaborative multiplayer games. Given the game is formally described using the generic model, it enables the information extract by GameAdapt.KOM to be accessed and presented in a meaningful way to an instructor (Game Master). In the same way, it makes the defined adaptations available to the instructor. The Game Mastering interface as an extension of GameAdapt.KOM is presented in Chapter 6. It enhances the GameAdapt.KOM functionality with the concept of providing an instructor (GM) with the necessary information and access to adapt a game at run-time. Therefore, methods to extract information and to present that information to the GM, as well as methods to adapt the game by the GM are developed using the underlying collaborative game model. Thus, the GM concept enables an instructor to orchestrate a game at run-time, assessing the game state and the players (as well as their actions and performance) and adapting it according to his/her professional opinion.

Agent-based player and learner simulation

In order to be able to perform an exhaustive feasibility study with a high number of game sessions and a great variety of players (referring to their play style, knowledge and skills, and collaboration and teamwork abilities), an agent-based player and learner simulation was developed (Chapter 7). The agent model uses the game model defined above. This makes it possible to perceive game-world changes based on the defined game elements. The perceived data is used in the planning module to decide what goals to pursue and to select plans to achieve those goals. Therefore, the agent uses a model to represent the simulated player based on the integrated player model presented above. Using the defined interface, the control module transfers plans into actions that exist in the game using the defined interface. This is possible because the required actions are accessible via GameAdapt.KOM as the game and its actions are described in the developed game model. Hence, using the agent-based player and learner simulation, it is possible to configure virtual players with a desirable player model, learner model, and interaction model to be used in simulation runs or to replace a missing human player.

Simulation and prototyping environment in the form of a collaborative multiplayer Serious Game

To evaluate the concepts developed in this thesis, a simulation environment was designed and implemented in the form of the collaborative multiplayer Serious Game *Escape From Wilson Island* (EFWI) (Chapter 8). The game was implemented following design principles for collaborative games identified in the literature (Section 8.2). The concepts developed in Chapter 5 were implemented as an extension for this game, including situations, adaptations, and definitions of relevant information of EFWI. A Game Master front-end was realized, implementing the Game Mastering concepts developed in Chapter 6. Finally, the player simulation was implemented as an AI component extending EFWI. This enables the methods and concepts developed to be evaluated using a real game. Moreover, the evaluations can be performed using real players or simulated player agents with configurable knowledge and behavior. Apart from this, *Escape From Wilson Island*, including the Game Master frontend and the adaptation mechanism module, is ready to be used in real scenarios at school or in higher education. A first deployment was performed between December 2014 and January 2015 by AVM Rüsselsheim where it will be further used within the curriculum.

Feasibility study for the developed concepts

To show that the developed concepts do fulfill the requirements, a feasibility study was conducted. For evaluation of GameAdapt.KOM (Chapter 9), as a first step the simulated player agent model was evaluated for soundness (Section 9.1). The goal of this evaluation was to assess the correct and expected behavior of the simulated players based on the configurations of their integrated player model. The results showed that the simulated players behaved according to their configurations in a reasonable and understandable way. To assess the impact of the game adaptation system parameters, a $2^k \cdot r$ factorial design approach was chosen. Based on the results of this evaluation, the automatic game adaptation mechanism was evaluated using the simulated player agents (Section 9.4). The same evaluation was repeated with a smaller set of real players to confirm the results. Finally, using the same design approach, the Game Mastering concept was evaluated in a comparative evaluation with a group of players playing in the presence of a GM and a group of players playing without a GM (Section 9.6). It was shown that both the automatic adaptation mechanism and the Game Mastering positively influenced player performance, learning success, and user experience.

10.3 CRITICAL REFLECTION ON THESIS GOALS

In Section 1.2.2, the goal of this thesis was stated based on the three identified challenges *heterogeneity of players and learners*, *high cognitive load on the instructor*, and *reluctance towards the use of Serious Games*. The goal was formulated in the form of two hypotheses:

> Providing an instructor in a collaborative multiplayer Serious Game with technology to assess the game process and player information and to adapt the game according to the instructor's professional opinion improves learning success, the game experience, and the players' performance with regard to the goal of the game.

> Automatic adaptation of a collaborative multiplayer game to the needs of a heterogeneous group considering gaming, learning, and interaction improves learning success, the game experience, and the players' performance with regard to the goal of the game.

From these hypotheses, the following research questions were formulated:

Research Question 1: *How can the most well suited adaptation of a multiplayer (Serious) game be determined depending on a given game situation with regard to players' traits, levels of knowledge, learning styles, and interaction?*

Research Question 2: *Can a positive impact on players' learning, gaming, and interaction performance be measured when automatic adaptation is used compared to a session without automatic adaptation?*

Research Question 3: *How can a Game Master get the required information from a collaborative multiplayer (Serious) game and adaptation mechanisms to manipulate the game, considering the players and the current state of the game?*

Research Question 4: *Can a positive impact on players' learning, gaming, and interaction performance be measured when a Game Master using appropriate technological support is orchestrating a game compared to a session where an instructor oversees the learning/gaming process without Game Mastering support?*

The concept of the *automatic adaptation mechanism* was designed to answer Research Question 1. GameAdapt.KOM addresses the problem of automatic optimal adaptation selection based on recognized game situations taking into account player traits, knowledge, and interaction. The implementation of GameAdapt.KOM was able shown in various game sessions to be able to positively influence player performance, learning success, and interaction between players (collaboration and teamwork) using the automatic adaptation selection. This represents a positive answer to Research Question 2.

Regarding Hypothesis I, it can be stated that it was possible to improve learning success, gaming success, and game experience using an automatic adaptation. However, it should be noted that game experience can be measured only with real players as game experience is a psychological construct and is perceived by humans. Using the simulated agents, it was possible to show that the game could be adapted to work in more harmony with the players' player model. That this would result in a better perceived user experience is an assumption made in the literature. However, this implication could not be shown within this thesis.

Regarding Research Question 3, in Chapter 6, resulting from the literature, typical instructor tasks were identified, and based on those, requirements for Game Mastering were derived. Although personal differences impact which information a GM considers important and which adaptations he/she would like to use, from a technical perspective it was possible to identify information provision methods independent of personal preferences and game related methods to implement them.

To answer Research Question 4, the Game Master interface was developed and implemented. The resulting software solution, the Game Master front-end, which was implemented as an extension to *Escape From Wilson Island*, showed that a GM overseeing and managing a game session by adapting it according to his/her professional opinion using the methods provided by the Game Master frontend, was able to positively influence the game in terms of learning and interaction. The influence on the players' gaming behavior (i.e., player model) was, however, only marginal. This is to say, a positive effect on game experience was measured. However, this effect is rather attributed to the optimized challenge, and hence the resulting flow, than to the adaptation of the game in terms of player traits.

Considering Hypothesis II, it can be concluded that it was possible to show that an instructor using the Game Master front-end was able to improve players'/learners' learning success, gaming success, and gaming experience. It should be mentioned, however, that the actual effects strongly depend on the game, the players (their previous knowledge about the game, motivation, etc.), and the GM itself. The pedagogical expertise of the instructor does have a big influence on the impact a GM can make (with or without Game Mastering). Still, it was possible to show that a Game Mastering concept can be designed and implemented to enable an instructor from a technical perspective to orchestrate a game-based collaborative learning session from within the game at run-time, thus being able to positively influence the players.

10.4 OUTLOOK

The contributions of this thesis are considered a first step toward a broader acceptance of Serious Games in collaborative learning scenarios - be it in school classes, institutes of higher education, or in corporate training. Moreover, the author hopes it is another step in the transition of Serious Games to the multiplayer sector, to a combination of collaborative learning principles and gaming technology, and to a better integration of the instructor in game-based learning scenarios. The concepts of Game Mastering might help to improve control over the complexity that a game-based multiplayer learning environment represents to the instructor, hence reducing reluctance toward using it. Further, being able to automatically adapt games and game-based learning scenarios to the needs of a usually heterogeneous group of players/learners might help to make game-based learning approaches easier to use and open up new fields of application for Serious Games. The findings in this thesis show that it is possible to make complex Serious Games manageable and thus broaden their field of application. It is considered necessary to deploy the Game Mastering concept to a broader set of players using more and different games.

The contributions and findings in this thesis might open up new, interesting research topics in the interdisciplinary field of Serious Games research. In the following, three interesting fields of research are outlined, resulting partially from new research questions based on the findings in this thesis.

Enhanced player and learner models and instructor analysis

The research questions formulated within this thesis were mainly covered from a technical/engineering perspective. Although this is necessary, it appears essential to tackle those questions in a more interdisciplinary way. The role of the instructor (Game Master) is of vital importance. Hence, it should be examined more closely within the context of game-based learning. This might help to better understand the needs of instructors orchestrating game-based learning scenarios, resulting in a better understanding of requirements for the assessment, control, and adaptation mechanisms of Serious Games. Apart from the instructor, there appears to be a need for more accurate player and learner models to better understand players and learners in the process of learning in game-based scenarios. This is accompanied by the need to assess those models within games. As of now, there are no well-engineered methods known to the author to soundly assess a player or learner model within a game without breaking the flow of the game (in form of tests, mini-games, etc.).

Serious Games Design

With *Escape From Wilson Island*, a Serious Game was designed as a test environment to evaluate the developed concepts. The game was designed following Serious Game design patterns, guidelines for collaborative gaming, and general game design mechanics. Game design is a field that is widely considered to be characterized by its artistic focus. Although plenty of formal approaches, developed patterns, and guidelines based on the analysis of games exist, there is an undeniably large amount of artistic, intuitive, and explorative work included. Focusing on the development of Serious Games, an additional challenge appears: the meaningful inclusion of a serious content, whether it is learning content, information for opinion making, or the inclusion of physical exercises (exergames). Moreover, traditional game design does not consider the presence of a GM who might influence the pace, difficulty, or shape of a game. Likewise, adaptability needs to be considered from the very beginning of the game design process. Hence, there appears to be a need for deeper research in the field of Serious Games design, especially when it comes to designing games with a certain goal (e.g., learning, opinion forming, exercise), or when the game focuses on collaboration or teamwork aspects, is intended to be adaptive, or should support Game Mastering. New concepts and formal approaches to Serious Games design are required to be able to cope with those challenges.

Game Mastering and adaptation in other fields of games research

Currently, the focus on the adaptation and Game Mastering approach proposed in this thesis lies within the field of Serious Games. However, it might easily be transferred to the field of entertainment (traditional games). The global games market has become a huge growth sector during the last decade and appears to have the potential to become even bigger. A huge part of this sector is the field of MMORPGs, such as *World of Warcraft (WoW)* and others. The role-play game sector attracts millions of players (about 10 million active WoW players in December 2014). Yet, role-play games still fail to provide 'living' and developing worlds. Worlds are designed and created before they are released, and each player - although in a world with thousands of other players at the same time - experiences the same narration, story hooks, and quests. So far, there are no large-scale games on the market offering a changing world that reacts to players' actions with an active and developing narration that takes the players into account. The concept of Game Mastering and adaptation within such a game might offer a whole new way in which these games are played. However, the development of concepts to transfer Game Mastering to large-scale games with possibly thousands of simultaneous players is required. Moreover, existing sandbox games, such as *Minecraft* offer the possibility to transfer the concepts into an open world with a smaller-scale game (5-100 simultaneous players). These games are already established on the market, and initial developments toward educational applications (e.g., *MinecraftEdu*) show that they can be used for educational purposes and that there is user interest. Again, there is a need for meaningful Serious Game design and the transfer of Game Mastering and adaptation principles to this form of game. Research might provide novel concepts and mechanisms to address those needs.

10.5 FINAL REMARKS

Education is a key issue in today's society. Serious Games have been shown to offer new ways of providing educational content within a game-based environment. However, there are obstacles that hinder the use of Serious Games in areas of application like schools, higher education, or corporate training. Poor design resulting in either a lack of fun or insufficiently integrated serious content, general reluctance due to a lack of trust in the technology or an instructor's computer skills, and the lack of proof of effects of the use of Serious Games are reasons for the reluctance to use Serious Games. This thesis has identified and motivated a need to address these problems by developing a way to improve the inclusion of instructors, thus reducing an existing reluctance toward this technology. Moreover, it addressed the problem of the heterogeneous group of learners through an automatic adaptation mechanism for collaborative learning games. The findings in thesis showed that it is possible to support instructors in game-based learning scenarios by developing in-game methods for the assessment and orchestration of games at run-time and that it is possible to adapt games at run-time to a heterogeneous group of players with different skills, abilities, and preferences, thus increasing their game performance and learning success. This can form the basis for the establishment of broader fields of application for Serious Games with more ease of use for both players and instructors.

BIBLIOGRAPHY

[1] Results in Focus: What 15-year-olds Know and What They Can Do With What They Know: Key Results From PISA 2012. *Organization for Economic Cooperation and Development*, 2012. URL http://www.oecd.org/pisa/keyfindings/pisa-2012-results-overview.pdf. (Retrieved: 06/02/15).

[2] Spieler in Deutschland , 2014. URL http://www.biu-online.de/de/fakten/reichweiten/spieler-in-deutschland.html. Retrieved: 15/01/15.

[3] Altersverteilung , 2014. URL http://www.biu-online.de/de/fakten/reichweiten/altersverteilung.html. Retrieved: 15/01/15.

[4] Age Breakdown of Video Game Players in the United States in 2014 , 2014. URL http://www.statista.com/statistics/189582/age-of-us-video-game-players-since-2010/. Retrieved: 15/01/15.

[5] Steven L. Abrantes and Luis B. Gouveia. Evaluating Adoption of Innovations of Mobile Devices and Desktops within Collaborative Environments in a Higher Education Context. *Communications of the IBIMA*, Vol. 2011, 2011. doi: 10.5171/2011.341897.

[6] Ernest Adams. *Fundamentals of Game Design*. New Riders, Berkeley, CA, 2010. ISBN 0321643372.

[7] Alan Amory. Game Object Model Version II: a Theoretical Framework for Educational Game Development. *Educational Technology Research and Development*, 55(1):51–77, 2007.

[8] Alan Amory, Kevin Naicker, Jacky Vincent, and Claudia Adams. The Use of Computer Games as an Educational Tool: Identification of Appropriate Game Types and Game Elements. *British Journal of Educational Technology*, 30(4):311–321, 1999.

[9] Erik Andersen, Yun-En Liu, Ethan Apter, François Boucher-Genesse, and Zoran Popović. Gameplay Analysis Through State Projection. In *Proceedings of the fifth International Conference on the Foundations of Digital Games*, pages 1–8. ACM, 2010.

[10] Christina Anger, Vera Demary, Oliver Koppel, and Axel Plünnecke. MINT-Frühjahrsreport 2013 - Innovationskraft, Aufstiegschance und demografische Herausforderung . Online, 2013. URL http://www.iwkoeln.de/_storage/asset/111717/storage/master/file/2790404/download/MINT-Frühjahrsbericht-2013.pdf. (Retrieved: 06/02/15).

[11] Thomas Augustin, Cord Hockemeyer, Michael Kickmeier-Rust, and Dietrich Albert. Individualized Skill Assessment in Digital Learning Games: Basic Definitions and Mathematical Formalism. *IEEE Transactions On Learning Technologies*, 4(2):138–148, 2011.

[12] Ruth Aylett. Emergent Narrative, Social Immersion and "Storyfication". In *Proceedings of the Narrative and Learning Environments Conference NILE00*, Edinburgh, Scotland, 2000.

[13] Ruth Aylett, Sandy Louchart, Anders Tychsen, Michael Hitchens, Rui Figueiredo, and Carlos Delgado Mata. Managing Emergent Character-based Narrative. In *INTETAIN '08: Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment*, pages 1–8, ICST, Brussels, Belgium, 2008. ISBN 978-963-9799-13-4.

[14] Michael Baker and Kristine Lund. Promoting Reflective Interactions in a Computer-supported Collaborative Learning Environment. *Journal of Computer Assisted Learning*, 13:175–193, 1997.

[15] Richard A. Bartle. Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs. *Journal of MUD Research*, 1(1):19, 1996.

[16] Francesco Bellotti, Riccardo Berta, Alessandro De Gloria, and Ludovica Primavera. Adaptive Experience Engine for Serious Games. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4):264–280, 2009. ISSN 1943-068X.

[17] Bryan Bergeron. *Developing Serious Games (Game Development Series)*. Charles River Media, 2006. ISBN 1584504447.

[18] Sonja Bergsträßer. *Virtuell Kontextbasierte Dienste - Definition, Beschreibung und Verwendung von Kontext in Virtuellen Welten*. PhD thesis, TU Darmstadt, May 2010.

[19] Elizabeth Bonsignore, Kari Kraus, Ahn June, Visconti Amanda, Ann Fraistat, and Allison Druin. Alternate Reality Games: Platforms for Collaborative Learning. In *Proceedings of the Tenth International Conference of the Learning Sciences*, Sydney: International Society of the Learning Sciences, 2012.

[20] Walter R. Boot, Arthur F. Kramer, Daniel J. Simons, Monica Fabiani, and Gabriele Gratton. The Effects of Video Game Playing on Attention, Memory, and Executive Control. *Acta Psychologica*, 129(3):387–398, 2008.

[21] Clint Bowers, Eduardo Salas, Carolyn Prince, and Michael Brannick. Games Teams Play: A method for Investigating Team Coordination and Performance. *Behavior Research Methods, Instruments, & Computers*, 24(4):503–506, 1992.

[22] Joseph Campbell. *The Hero with a Thousand Faces*, volume 17. New World Library, USA, 3rd edition, 2008. ISBN 978-1-57731-593-3.

[23] Thibault Carron and Jean-Charles Marty. Enhancement of Adaptation and Monitoring in Game-Based Learning Environments. *Student Usability in Educational Software and Games: Improving Experiences*, page 201, 2012.

[24] Darryl Charles and Michaela Black. Dynamic Player Modelling: A Framework for Player-Centered Digital Games. In *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, pages 29–35, 2004.

[25] Darryl Charles, Michael McNeill, Moira McAlister, Michaela Black, Adrian Moore, Karl Stringer, Julian Kücklich, and Aphra Kerr. Player-centred Game Design: Player Modelling and Adaptive Digital Games. In *Proceedings of the Digital Games Research Conference*, volume 285. Citeseer, 2005.

[26] Jenova Chen. Flow in Games (and Everything Else). *Communications of the ACM*, 50(4):31–34, 2007.

[27] Marcus D. Childress and Ray Braswell. Using Massively Multiplayer Online Roleplaying Games for Online Learning. *Distance Education*, 27(2):187–196, 2006.

[28] Chien Chou and Meng-Jung Tsai. Gender Differences in Taiwan High School Students' Computer Game Playing. *Computers in Human Behavior*, 23(1):812–824, 2007.

[29] César A. Collazos, Luis A. Guerrero, José A. Pino, and Sergio F. Ochoa. Collaborative Scenarios to Promote Positive Interdependence Among Group Members. In *Groupware: Design, Implementation, and Use*, pages 356–370. Springer, 2003.

[30] Chris Crawford. *Chris Crawford on Game Design*. New Riders, US, 2003. ISBN 0-13-146099-4.

[31] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper and Row, New York, NY, 1990. ISBN 0-06-092043-2.

[32] Larry Cuban. *The Classroom Use of Technology since 1920*, chapter Teachers and machines. Teachers College Press, 1986.

[33] Aaron Delwiche. Massively Multiplayer Online Games (MMOs) in the New Media Classroom. *Educational Technology & Society*, 9(3):160–172, 2006.

[34] Pierre Dillenbourg. What Do You Mean by Collaborative Learning? In Pierre Dillenbourg, editor, *Collaborative-learning: Cognitive and Computational Approaches*, pages 1–19. Elsevier, Oxford, 1999.

[35] Pierre Dillenbourg and Patrick Jermann. Technology for Classroom Orchestration. In *New science of learning*, pages 525–552. Springer, 2010.

[36] Pierre Dillenbourg, Michael Baker, Agnes Blaye, and Claire O'Malley. The Evolution of Research on Collaborative Learning. In E. Spada & P. Reiman (Eds), editor, *Learning in Humans and Machine: Towards an interdisciplinary learning science.*, pages 189–211. Oxford: Elsevier, 1995.

[37] Dan Dixon. Player Types and Gamification. In *Proceedings of the CHI 2011 Workshop on Gamification*, 2011.

[38] Jean-Paul Doignon and Jean-Claude Falmagne. *Knowledge Spaces*. Springer, Germany, 1999. ISBN 3540645012.

[39] Nicola Döring. *Sozialpsychologie des Internet. Die Bedeutung des Internet für Kommunikationsprozesse, Identitäten, soziale Beziehungen und Gruppen.* Hogrefe Verlag, 2003. ISBN 3801714667.

[40] Matthew W.G. Dye, C. Shawn Green, and Daphne Bavelier. Increasing Speed of Processing with Action Video Games. *Current Directions in Psychological Science*, 18(6):321–326, 2009.

[41] MWG Dye, CS Green, and D Bavelier. The Development of Attention Skills in Action Video Game Players. *Neuropsychologia*, 47(8):1780–1789, 2009.

[42] Magy Seif El-Nasr, Bardia Aghabeigi, David Milam, Mona Erfani, Beth Lameman, Hamid Maygoli, and Sang Mah. Understanding and Evaluating Cooperative Games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 253–262. ACM, 2010.

[43] Miguel Encarnação. On the Future of Serious Games in Science and Industry. In *Proceedings of CGames*, pages 9 – 16, 2009.

[44] Eustace, K. and Lee, M. and Fellows, G. and Bytheway, A. and Irving, L. The Application of Massively Multiplayer Online Role Playing Games to Collaborative Learning and Teaching Practice in Schools. In R. Atkinson, C. McBeath, D. Jonas-Dwyer, and R. Phillips, editors, *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference*, 2004.

[45] Carlo Fabricatore. Learning and Videogames: An Unexploited Synergy. *Unpublished manuscript*, 2000.

[46] Melissa A. Federoff. *Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games*. PhD thesis, Citeseer, 2002.

[47] Daniel Fu and Ryan Houlette. Putting AI in Entertainment: An AI Authoring Tool for Simulation and Games. *Intelligent Systems, IEEE*, 17(4):81–84, 2002.

[48] Tracy Fullerton, Chris Swain, and Steve Hoffman. Improving Player Choices. Gamasutra (March 2004), March 2004. URL http://www.gamasutra.com/view/feature/2039/improving_player_choices.php. (Retrieved: 06/02/15).

[49] Lisa Galarneau. 2014 Global Gaming Stats: Who's Playing What, and Why? Online, January 2014. URL http://www.bigfishgames.com/blog/2014-global-gaming-stats-whos-playing-what-and-why/. (Retrieved: 06/02/15).

[50] James Paul Gee. What Video Games Have to Teach Us About Learning and Literacy. *Computers in Entertainment (CIE)*, 1:20, 2003.

[51] James Paul Gee. Learning by Design: Good Video Games as Learning Machines. *E-Learning and Digital Media*, 2(1):5–16, 2005.

[52] Stefan Göbel, André de Carvalho Rodrigues, Florian Mehm, and Ralf Steinmetz. Narrative Game-based Learning Objects for Story-based Digital Educational Games. In Michael D. Kickmeier-Rust, editor, *Proceedings of the 1st International Open Workshop on Intelligent Personalization and Adaptation in Digital Educational Games*, pages 43–53, Oct 2009.

[53] Stefan Göbel, Viktor Wendel, Christopher Ritter, and Ralf Steinmetz. Personalized, Adaptive Digital Educational Games using Narrative Game-based Learning Objects. pages 438–445. Springer, 2010.

[54] Stefan Göbel, Michael Gutjahr, and Ralf Steinmetz. What Makes a Good Serious Game - Conceptual Approach Towards a Metadata Format for the Description and Evaluation of Serious Games. In Dimitris Gouscous and Michalis Meimaris, editors, *Proceedings of the 5th European Conference on Games Based Learning*, pages 202–210. Academic Conferences Limited, Reading, UK, Oct 2011. ISBN 978-1-908272-19-5 CD.

[55] Stefan Göbel, Michael Gutjahr, and Sandro Hardy. Evaluation of Serious Games. In Klaus Bredl and Wolfgang Bösche, editors, *Serious Games and Virtual Worlds in Education, Professional Development, and Healthcare*, chapter 7, pages 105–115. IGI Global, Hershey, USA, 1 edition, 2013. ISBN 9781466636736.

[56] Shawn Graham. Vespasian, Civ IV, and Intro to Roman Culture, 2007. URL http://planetcivilization.gamespy.com/View80c7.html?view=Articles.Detail&id=33. (Retrieved: 06/02/15).

[57] C.S. Green and D. Bavelier. Learning, Attentional Control, and Action Video Games. *Current Biology*, 22(6):R197–R206, 2012.

[58] C.S. Green, A. Pouget, and D. Bavelier. A General Mechanism for Learning with Action Video Games: Improved Probabilistic Inference. *Current Biology*, 20:1573–15792, 2010.

[59] Bradley S. Greenberg, John Sherry, Kenneth Lachlan, Kristen Lucas, and Amanda Holmstrom. Orientations to Video Games Among Gender and Age Groups. *Simulation & Gaming*, 41(2):238–259, 2010.

[60] Jennifer S. Groff and Chrystalla Mouza. A Framework for Addressing Challenges to Classroom Technology Use. *Association for the Advancement of Computing in Education (AACE) Journal*, 16(1):21–46, 2008.

[61] Jörg Haake, Gerhard Schwabe, and Martin Wessner. *CSCL-Kompendium - Lehr- und Handbuch zum Computerunterstützten Kooperativen Lernen*. Oldenbourg Wissenschaftsverlag, 2004. ISBN 3486274368.

[62] Johan Hagelbäck and Stefan J. Johansson. A Multi-Agent Potential Field-based Bot for a Full RTS Game Scenario. *International Journal of Computer Games Technology*, Volume 2009:10, 2009. doi: http://dx.doi.org/10.1155/2009/910819. Article ID 910819.

[63] Raija Hämäläinen. Using a Game Environment to Foster Collaborative Learning: a Design-based Study. *Technology, Pedagogy and Education*, 20(1):61–78, 2011.

[64] Raija Hämäläinen and Kimmo Oksanen. Challenge of Supporting Vocational Learning: Empowering Collaboration in a Scripted 3D Game - How Does Teachers' Realtime Orchestration Make a Difference? *Computers and Education*, 59:281–293, 2012.

[65] Raija Hämäläinen, Tony Manninen, Sanna Järvelä, and Päivi Häkkinen. Learning to Collaborate: Designing Collaboration in a 3-D Game Environment. *The Internet and Higher Education*, 9(1):47 – 61, 2006. ISSN 1096-7516. doi: DOI:10.1016/j.iheduc.2005.12.004.

[66] Sandro Hardy, Stefan Göbel, Michael Gutjahr, Josef Wiemeyer, and Ralf Steinmetz. Adaptation Model for Indoor Exergames. *International Journal of Computer Science in Sport*, 11(Special Edition: Serious Games - Theory, Technology & Practice):73–85, 2012.

[67] Casper Harteveld. *Triadic Game Design*. Springer-Verlag New York Inc, 2011. ISBN 978-1-84996-157-8.

[68] Jürgen Heller, Christina Steiner, Cord Hockemeyer, and Dietrich Albert. Competence-based Knowledge Structures for Personalised Learning. *International Journal on E-Learning*, 5(1):75–88, 2006.

[69] J.C. Herz. Gaming the System: What Higher Education Can Learn From Multiplayer Online Worlds. In *The Internet and the University: Forum*, pages 169–291, 2001.

[70] Kathrin Hoeckel and R. Schwartz. OECD-Studie zur Berufsbildung: Lernen für die Arbeitswelt, 2010. URL http://www.oecd.org/berlin/45924455.pdf. (Retrieved: 06/02/15).

[71] H. Ulrich Hoppe and Rolf Ploetzner. Can Analytic Models Support Learning in Groups. *Collaborative-learning: Cognitive and computational approaches*, pages 147–168, 1999.

[72] Ryan Houlette. Player Modelling for Adaptive Games. *AI Game Programming Wisdom II*, pages 557–566, 2004.

[73] Wijnand IJsselsteijn, Yvonne de Kort, Karolien Poels, Audrius Jurgelionis, and Francesco Bellotti. Characterising and Measuring User Experiences in Digital Games. In *International Conference on Advances in Computer Entertainment Technology*, 2007.

[74] Akiko Inaba, Ryoji Ohkubo, Mitsuru Ikeda, and Riichiro Mizoguchi. Modeling Learner-to-Learner Interaction Process in Collaborative Learning - an Ontological Approach to Interaction Analysis. In *Supplementary Proceedings of the International Conference on Computer Support for Collaborative Learning (CSCL2003)*, pages 4–6, Bergen, Norway, 2003.

[75] Troy Innocent and Stewart Haines. Nonverbal Communication in Multiplayer Game Worlds. In *Proceedings of the 4th Australasian conference on Interactive entertainment*, page 11. RMIT University, 2007.

[76] David W. Johnson and Roger T. Johnson. *Learning Together and Alone, Cooperative, Competitive, and Individualistic Learning.* Needham Heights, MA: Prentice-Hall., 1994.

[77] David W. Johnson and Roger T. Johnson. Making Cooperative Learning Work. *Theory into practice*, 38(2):67–73, 1999.

[78] David W. Johnson and Roger T. Johnson. An Educational Psychology Success Story: Social Interdependence Theory and Cooperative Learning. *Educational researcher*, 38(5):365–379, 2009.

[79] Timo Kaukoranta, Jouni Smed, and Harri Hakonen. Understanding Pattern Recognition Methods. *AI Game Programming Wisdom*, 2:579–589, 2003.

[80] Henry Kelly, Kay Howell, Eitan Glinert, Loring Holding, Chris Swain, Adam Burrowbridge, and Michelle Roper. How to Build Serious Games. *Communications of the ACM*, 50(7):44–49, 2007.

[81] Harri Ketamo and Kristian Kiili. New Teachership in Game Worlds. In *Proceedings of ECGBL 2009*, 2009.

[82] Michael D. Kickmeier-Rust and Dietrich Albert. Educationally Adaptive: Balancing Serious Games. *Int. J. Comp. Sci. Sport*, 11(1), 2012.

[83] Michael D. Kickmeier-Rust and Dietrich Albert. Using Hasse Diagrams for Competence-Oriented Learning Analytics. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, pages 59–64. Springer, 2013.

[84] K. Kiili. Digital Game-based Learning: Towards an Experiential Gaming Model. *The Internet and higher education*, 8(1):13–24, 2005.

[85] John Kirriemuir and Angela McFarlane. Use of Computer and Video Games in the Classroom. In *Level Up: The digital games research conference, Utrecht University, The Netherlands*, 2003.

[86] Eric Klopfer, Scot Osterweil, Jennifer Groff, and Jason Haas. Using the Technology of Today, in the Classroom Today: The Instructional Power of Digital Games, Social Networking, Simulations and How Teachers can Leverage Them. *The Education Arcade*, pages 1–21, 2009.

[87] Johannes Konert. *Interactive Multimedia Learning: Using Social Media for Peer Education in Single-Player Educational Games*. Springer, Darmstadt, Germany, 2014. ISBN 978-3-319-10256-6.

[88] Johannes Konert, Kristina Richter, Florian Mehm, Stefan Göbel, Regina Bruder, and Ralf Steinmetz. PEDALE - A Peer Education Diagnostic and Learning Environment. *Journal of Educational Technology & Society*, 15(4):27–38, Sep 2012. ISSN 1176-3647.

[89] Johannes Konert, Dmitrij Burlak, Stefan Göbel, and Ralf Steinmetz. GroupAL: ein Algorithmus zur Formation und Qualitätsbewertung von Lergruppen in E-Learning-Szenarien mittels n-dimensionaler Gütekriterien. In Andreas Breitner and Christoph Rensing, editors, *Proceedings der Deutschen E-Learning Fachtagung (DeLFI)*, pages 71–82, Bremen, Germany, Sep 2013. Gesellschaft für Informatik e.V. (gi), Köllen. ISBN 9783885796121.

[90] Johannes Konert, Viktor Wendel, Kristina Richter, and Stefan Göbel. Collaborative Learning and Game Mastering in Multiplayer Games. In Klaus Bredl and Wolfgang Bösche, editors, *Serious Games and Virtual Worlds in Education, Professional Development, and Healthcare*, chapter 6, pages 85–104. IGI Global, Hershey, USA, 1 edition, 2013. ISBN 9781466636736.

[91] Johannes Konert, Michael Gutjahr, Stefan Göbel, and Ralf Steinmetz. Modeling the Player: Predictability of the Models of Bartle and Kolb Based on NEO-FFI (Big5) and the Implications for Game Based Learning. *International Journal of Game-Based Learning*, 4(2):36–50, January 2014. ISSN 2155-6849.

[92] Klaus Korossy. Modeling Knowledge as Competence and Performance. *Knowledge spaces: Theories, empirical research, and applications*, pages 103–132, 1999.

[93] Johann A. Larusson and Richard Alterman. Wikis to Support the Collaborative Part of Collaborative Learning. *International Journal of Computer-Supported Collaborative Learning*, 4(4):371–402, 2009.

[94] Robin D. Laws. *Robin's Laws of Good Game Mastering*. Steve Jackson Games, 2002. ISBN 1-55634-629-8.

[95] Nicole Lazzaro. Why we Play Games: Four Keys to More Emotion Without Story, 2004. URL http://www.xeodesign.com/whyweplaygames/xeodesign_whyweplaygames.pdf. (Retrieved: 06/02/15).

[96] Ronan Le Hy, Anthony Arrigoni, Pierre Bessiere, and Olivier Lebeltel. Teaching Bayesian Behaviours to Video Game Characters. *Robotics and Autonomous Systems*, 47(2):177–185, 2004.

[97] Craig A. Lindley. The Semiotics of Time Structure in Ludic Space as a Foundation for Analysis and Design. *Game Studies*, 5(1):2005, 2005.

[98] Tom Lowrie and Robyn Jorgensen. Gender Differences in Students' Mathematics Game Playing. *Computers & Education*, 57(4):2244–2248, 2011.

[99] Regan L. Mandryk, M. Stella Atkins, and Kori M. Inkpen. A Continuous and Objective Evaluation of Emotional Experience with Interactive Play Environments. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1027–1036. ACM, 2006.

[100] Tony Manninen. Towards Communicative, Collaborative and Constructive Multi-player Games. In *In: Mäyrä F. (ed) Computer Games and Digital Cultures Conference*, pages 7–8. University press, 2002.

[101] Tony Manninen. Interaction Forms and Communicative Actions in Multiplayer Games. *Game studies*, 3(1):2003, 2003.

[102] Tony Manninen and Tuomo Korva. Designing Puzzles for Collaborative Gaming Experience–CASE: eScape. In S. Castell and J. Jennifer, editors, *Selected papers of the Digital Interactive Games Research Associationś second international-conference (DiGRA 205)*, pages 233–247, Vancouver, Canada, 2005. Digital Interactive Games Research Association.

[103] Samah Mansour and Mostafa El-Said. Multi-Players Role- Playing Educational Serious Games: A Link between Fun and Learning. *The International Journal of Learning*, 15(11):229–240, 2008.

[104] Michael Mateas and Andrew Stern. A Behavior Language for Story-Based Believable Agents. *IEEE Intelligent Systems*, 17(4):39–47, 2002.

[105] Igor Mayer, Geertje Bekebrede, Casper Harteveld, Harald Warmelink, Qiqi Zhou, Theo Ruijven, Julia Lo, Rens Kortmann, and Ivo Wenzler. The Research and Evaluation of Serious Games: Toward a Comprehensive Methodology. In *British Journal of Educational Technology*. Wiley Online Library, 2013.

[106] Angela McFarlane, Anne Sparrowhawk, and Ysanne Heald. Report on the Educational Use of Games: an Exploration by TEEM of the Contribution which Games Can Make to the Education Process. Technical report, TEEM, St. Ives, Cambridgeshire, UK, 2002.

[107] Katelyn McKenna and Sangchul Lee. A Love Affair with MUDs: Flow and Social Interaction in Multi-user-dungeons, 2003. URL http://www.websm.org/uploadi/editor/McKenna_Sangchul_2004_MUDs_love_affair.doc. (Retrieved: 06/02/15).

[108] Florian Mehm. *Authoring of Adaptive Single-Player Educational Games*. Ph.d. thesis, Technische Universität Darmstadt, 2013.

[109] Florian Mehm, Stefan Göbel, and Ralf Steinmetz. An Authoring Tool for Educational Adventure Games: Concept, Game Models and Authoring Processes. *International Journal on Game-Based Learning*, 3(1):63–79, 2013.

[110] Florian Mehm, Christian Reuter, and Stefan Göbel. Authoring of Serious Games for Education. In Klaus Bredl and Wolfgang Bösche, editors, *Serious Games and Virtual Worlds in Education, Professional Development, and Healthcare*, chapter 4, pages 60–73. IGI Global, Hershey, USA, 1 edition, 2013. ISBN 9781466636736.

[111] Ronald Metoyer, Simone Stumpf, Christoph Neumann, Jonathan Dodge, Jill Cao, and Aaron Schnabel. Explaining how to play real-time strategy games. *Knowledge-Based Systems*, 23(4):295–301, 2010.

[112] David Michael and Sande Chen. *Serious games: Games that Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005. ISBN 1592006221.

[113] Lutz Michel, Hayp Werner, Lutz Goertz, and Axel Wolpert. Digitale Schule–wie Lehrer Angebote im Internet nutzen. Eine Bestandsaufnahme im Auftrag des Bundesministeriums für Bildung und Forschung (BMBF). *Institut fuer Medien und Kompetenzforschung*, 2008:26, 2008.

[114] Philip Mildner, Christopher Campbell, Mark Himmelsbach, Christoph Malassa, Marco Miczka, and Wolfgang Effelsberg. A Serious Game for Architectural Knowledge in the Classroom. In *E-Learning and Games for Training, Education, Health and Sports*, pages 72–77. Springer, 2012.

[115] Alice Mitchell and Carol Savill-Smith. *The Use of Computer and Video Games for Learning: A Review of the Literature*. Learning and Skills Development Agency London, Great Britain, 2004. ISBN 1853389048.

[116] Ben Morgan Jr., Albert Glickman, Elizabeth Woodard, Arthur Blaiwes, and Eduardo Salas. Measurement of Team Behaviors in a Navy Environment. Technical report, DTIC Document, 1986.

[117] Lennart Nacke. *Affective Ludology: Scientific Measurement of User Experience in Interactive Entertainment*. PhD thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 2009. http://phd.acagamic.com, (ISBN) 978-91-7295-169-3.

[118] Lennart Nacke, Anders Drachen, and Stefan Göbel. Methods for Evaluating Gameplay Experience in a Serious Gaming Context. *International Journal of Computer Science in Sport*, 9(2):1–12, 2010.

[119] Maaz Nasir, Kelly Lyons, Rock Leung, and Ali Moradian. Cooperative Games and Their Effect on Group Collaboration. In *Design Science at the Intersection of Physical and Virtual Design*, pages 502–510. Springer, 2013.

[120] OECD. Bildung auf einen Blick 2013: OECD-Indikatoren. Online, 2013. URL http://www.oecd-ilibrary.org/education/bildung-auf-einen-blick-2014-oecd-indikatoren_eag-2014-de. (Retrieved: 06/02/15).

[121] Orlando J. Olivares. Collaborative vs. Cooperative Learning: The Instructor's Role in Computer Supported Collaborative Learning. In K.L. Orvis and A.L.R. Lassiter, editors, *Computer-supported collaborative learning: Best practices and principles for instrucors*, pages 20–30. Hershey, PA: Information Science Publishing, 2007.

[122] Antonio Padilla-Meléndez, Ana Rosa del Aguila-Obra, and Aurora Garrido-Moreno. Perceived Playfulness, Gender Differences and Technology Acceptance Model in a Blended Learning Scenario. *Computers & Education*, 63:306–317, 2013.

[123] Carol R. Paris, Eduardo Salas, and Janis A. Cannon-Bowers. Teamwork in Multi-person Systems: A Review and Analysis. *Ergonomics*, 43(8):1052–1075, 2000.

[124] Federico Peinado and Pablo Gervás. Transferring Game Mastering Laws to Interactive Digital Storytelling. *Lecture Notes in Computer Science.*, 3105:48–54, 2004.

[125] Marc Prensky. Digital Game-based Learning. *Computes in Entertainment*, 1(1): 21, 2003.

[126] Ashwin Ram, Santiago Ontañón, and Manish Mehta. Artificial Intelligence for Adaptive Computer Games. In *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society (FLAIRS) Conference on Artificial Intelligence*, 2007.

[127] Matthias Rauterberg. Determinantes for Collaboration in Networked Multi-User Games. In Ryohei Nakatsu and Junichi Hoshino, editors, *Entertainment Computing*, volume 112 of *IFIP - The International Federation for Information Processing*, pages 313–321. Springer US, 2003. ISBN 978-1-4757-5153-6. doi: 10.1007/978-0-387-35660-0_38. URL http://dx.doi.org/10.1007/978-0-387-35660-0_38.

[128] Christian Reuter. Authoring Multiplayer Serious Games. In Georgios N. Yannakakis and Espen Aarseth, editors, *Proceedings of the 8th International Conference on the Foundations of Digital Games*, pages 490–492, Chania, Greece, 2013.

[129] Christian Reuter, Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. Multiplayer Adventures for Collaborative Learning With Serious Games. In Patrick Felicia, editor, *6th European Conference on Games Based Learning*, pages 416–423, Reading, UK, 2012. Academic Conferences Limited.

[130] Christian Reuter, Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. Game Design Patterns for Collaborative Player Interactions. In Staffan Björk and Annika Waern, editors, *«Verb that ends in 'ing'> the <noun> of Game <plural noun> - Proceedings of DiGRA 2014*, page 16, Aug 2014.

[131] Mark O. Riedl and R. Michael Young. An Intent-Driven Planner for Multi-Agent Story Generation. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 186–193, Washington DC, USA, 2004. IEEE Computer Society. ISBN 1-58113-864-4.

[132] José B. Rocha, Samuel Mascarenhas, and Rui Prada. Game Mechanics for Cooperative Games. In *ZON Digital Games 2008*, pages 72–80, 2008.

[133] Pieter R. Roelfsema, Arjen van Ooyen, and Takeo Watanabe. Perceptual Learning Rules Based on Reinforcers and Attention. *Trends in cognitive sciences*, 14(2): 64–71, 2010.

[134] Jeremy Roschelle and Stephanie D. Teasley. The Construction of Shared Knowledge in Collaborative Problem Solving. In C. O'Malley, editor, *Computer-supported Collaborative Learning*, pages 69–97, Berlin, 1995. Springer-Verlag.

[135] James C. Rosser, Paul J. Lynch, Laurie Cuddihy, Douglas A. Gentile, Jonathan Klonsky, and Ronald Merrell. The Impact of Video Games on Training Surgeons in the 21st Century. *Archives of Surgery*, 142(2):181–186, 2007.

[136] James C. Rosser, Paul J. Lynch, Laurie Cuddihy, Douglas A. Gentile, Jonathan Klonsky, and Ronald Merrell. The Impact of Video Games in Surgical Training. *Archives of surgery*, 142(2):181–186, 2007.

[137] Richard Rouse III. *Game Design: Theory and Practice*. Jones & Bartlett Learning, 2010. ISBN 1556229127.

[138] Stuart Russell. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 2009. ISBN 0-13-103805-2.

[139] Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. The MIT Press, 2004. ISBN 0262240459.

[140] Pilar Sancho, Javier Torrente, and Baltasar Fernández-Manjón. Do Multi-User Virtual Environments Really Enhance Student's Motivation in Engineering Education? In *Proceedings of the 39th IEEE international conference on Frontiers in education conference*, pages 744–749. IEEE Press, 2009.

[141] Richard Sandford and Ben Williamson. Games and Learning - a Handbook from Futurelab, 2005. URL http://www2.futurelab.org.uk/resources/documents/handbooks/games_and_learning2.pdf. (Retrieved: 06/02/15).

[142] Ben Sawyer and Peter Smith. Serious Games Taxonomy. In *Slides from the Serious Games Summit at the Game Developers Conference*, 2008.

[143] Jesse Schell. *The Art of Game Design: A Book of Lenses*. Taylor & Francis US, 2008. ISBN 0123694965.

[144] Marc J. Shapiro, Roxane Gardner, Steven A. Godwin, Gregory D. Jay, David G. Lindquist, Mary L. Salisbury, and Eduardo Salas. Defining Team Performance for Simulation-based Training: Methodology, Metrics, and Opportunities for Emergency Medicine. *Academic Emergency Medicine*, 15(11):1088–1097, 2008.

[145] Laila Shoukry, Stefan Göbel, Galal H. Galal-Edeen, and Christian Sturm. Conducting Evaluation Studies of Mobile Games with Preschoolers. In Christoph Rensing and Stephan Trahasch, editors, *Proceedings of DeLFI Workshops 2014*, volume 1227, pages 262–269. CEUR Workshop Proceedings, Sep 2014.

[146] Laila Shoukry, Stefan Göbel, and Ralf Steinmetz. Towards Mobile Multimodal Learning Analytics. In *Learning Analytics for and in Serious Games - Workshop, EC-TEL 2014*, page 16, Sep 2014.

[147] Adam M. Smith, Chris Lewis, Kenneth Hullet, and Anne Sullivan. An Inclusive View of Player Modeling. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 301–303. ACM, 2011.

[148] Adam M. Smith, Chris Lewis, Kenneth Hullett, Gillian Smith, and Anne Sullivan. An Inclusive Taxonomy of Player Modeling. Technical Report UCSC-SOE-11-13, University of California, Santa Cruz, 2011.

[149] Pieter Spronck, Marc Ponsen, Ida Sprinkhuizen-Kuyper, and Eric Postma. Adaptive Game AI with Dynamic Scripting. *Machine Learning*, 63(3):217–248, 2006.

[150] Kurt Squire. Video Games in Education. *International journal of intelligent simulations and gaming*, 2(1):49–62, 2003.

[151] Kurt Squire. Changing the Game: What Happens When Video Games Enter the Classroom. *Innovate: Journal of online education*, 1(6), 2005.

[152] Kurt Squire and Sasha Barab. Replaying History: Engaging Urban Underserved Students in Learning World History Through Computer Simulation Games. In *Proceedings of the 6th international conference on Learning sciences*, pages 505–512. International Society of the Learning Sciences, 2004.

[153] Gerry Stahl, Timothy Koschmann, and Dan Suthers. *Cambridge Handbook of the Learning Sciences*, chapter Computer-supported Collaborative Learning: An Historical Perspective, pages 409–426. Cambridge University Press, 2006.

[154] Christina M. Steiner, Alexander Nussbaumer, and Dietrich Albert. Supporting Self-Regulated Personalised Learning through Competence-Based Knowledge Space Theory. *Policy Futures in Education*, 7(6):645–661, 2009.

[155] Constance A. Steinkuehler. Learning in Massively Multiplayer Online Games. In *ICLS '04: Proceedings of the 6th international conference on Learning sciences*, pages 521–528. International Society of the Learning Sciences, 2004.

[156] Ralf Steinmetz and Klara Nahrstedt. *Multimedia Applications*. Springer, 2004. ISBN 3540408495.

[157] Penelope Sweetser and Peta Wyeth. GameFlow : A Model for Evaluating Player Enjoyment in Games. *Computers in Entertainment (CIE)*, 3(3):1–24, 2005.

[158] Chek Tien Tan and Ho-lun Cheng. A Combined Tactical and Strategic Hierarchical Learning Framework in Multi-agent Games. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, pages 115–122. ACM, 2008.

[159] Phit-Huan Tan, Siew-Woei Ling, and Choo-Yee Ting. Adaptive Digital Game-based Learning Framework. In *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*, page 146. ACM, 2007.

[160] Mariët Theune, Sander Faas, Anton Nijholt, and Dirk Heylen. The Virtual Storyteller: Story Creation by Intelligent Agents. In *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, pages 204–215, 2002.

[161] Ann Marie Thomson, James L. Perry, and Theodore K. Miller. Conceptualizing and Measuring Collaboration. *Journal of Public Administration Research and Theory*, 19(1):23–56, 2009.

[162] David Thue, Vadim Bulitko, and Marcia Spetch. Player Modeling for Interactive Storytelling: A Practical Approach. *AI Game Programming Wisdom*, 4: 633–646, 2008.

[163] Javier Torrente, Pablo Moreno-Ger, and Baltasar Fernandez-Manjon. Learning Models for the Integration of Adaptive Educational Games in Virtual Learning Environments. In Zhigeng Pan, Xiaopeng Zhang, Abdennour El Rhalibi, Woontack Woo, and Yi Li, editors, *Technologies for E-Learning and Digital Entertainment*, volume 5093 of *Lecture Notes in Computer Science*, pages 463–474. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-69734-3.

[164] Linda Klebe Trevino and Jane Webster. Flow in Computer-mediated Communication Electronic Mail and Voice Mail Evaluation and Impacts. *Communication research*, 19(5):539–573, 1992.

[165] Tonio Triebel, Benjamin Guthier, Thomas Plotkowiak, and Wolfgang Effelsberg. Peer-to-peer Voice Communication for Massively Multiplayer Online Games. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1–5. IEEE, 2009.

[166] Anders Tychsen. Role Playing Games: Comparative Analysis Across Two Media Platforms. In *Proceedings of the 3rd Australasian conference on Interactive entertainment*, pages 75–82. Murdoch University, 2006.

[167] Anders Tychsen. Tales for the Many: Process and Authorial Control in Multiplayer Role-Playing Games. In *ICIDS '08: Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling*, pages 309–320, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-89424-7. doi: http://dx.doi.org/10.1007/978-3-540-89454-4_38.

[168] Anders Tychsen, Michael Hitchens, Thea Brolund, and Manolya Kavakli. The Game Master. In *Proceedings of the second Australasian conference on Interactive entertainment*, pages 215–222. Creativity & Cognition Studios Press, 2005.

[169] Richard Van Eck. Digital Game-based Learning: It's Not Just the Digital Natives Who Are Restless. *Educause Review*, 41(2):16, 2006.

[170] Dessislava Vassileva and Boyan Bontchev. Adaptation Engine Construction Based on Formal Rules. In *Proceedings of the International Conference on Computer Supported Education (CSEDU)*, 2009.

[171] Peter Vorderer, Tilo Hartmann, and Christoph Klimmt. Explaining the Enjoyment of Playing Video Games: The Role of Competition. In *Proceedings of the second international conference on Entertainment computing*, pages 1–9. Carnegie Mellon University, 2003.

[172] Jeroen Warmerdam, Maxim Knepflé, Rafael Bidarra, Geertje Bekebrede, and Igor Mayer. SimPort: a Multiplayer Management Game Framework. Technical report, Delft University of Technology, 2006.

[173] Noreen M. Webb. Testing a Theoretical Model of Student Interaction and Learning in Small Groups. *Interaction in cooperative groups: The theoretical anatomy of group learning*, 102:119, 1992.

[174] Ulrich Wechselberger. The Eduventure II. An Approach to Educational Game Design. In *CW '08: Proceedings of the 2008 International Conference on Cyberworlds*, pages 397–404, Washington, DC, USA, 2008. IEEE Computer Society.

[175] Viktor Wendel, Maxim Babarinow, Tobias Hörl, Sergej Kolmogorov, Stefan Göbel, and Ralf Steinmetz. *Woodment: Web-Based Collaborative Multiplayer Serious Game*, volume 6250 of *Lecture Notes in Computer Science*, pages 68–78. Springer, 1st edition, 2010.

[176] Viktor Wendel, Annika Kliem, Stefan Göbel, Josef Wiemeyer, and Ralf Steinmetz. Virtual Sports Instructor : A 3D Serious Game for Physical Education with Game Master Support. In *Ed-Media 2011-World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2011.

[177] Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. Game Mastering in Collaborative Multiplayer Serious Games. In Stefan Göbel, Wolfgang Müller, Bodo Urban, and Josef Wiemeyer, editors, *E-Learning and Games for Training, Education, Health and Sports - LNCS*, volume 7516, pages 23–34, Darmstadt, Germany, Sep 2012. Springer. ISBN 978-3-642-33465-8.

[178] Viktor Wendel, Michael Gutjahr, Stefan Göbel, and Ralf Steinmetz. Designing Collaborative Multiplayer Serious Games. *Education and Information Technologies*, 18(2):287–308, 2013.

[179] Joost Westra, Hado van Hasselt, Frank Dignum, and Virginia Dignum. Adaptive Serious Games Using Agent Organizations. 5920:206–220, 2009.

[180] Wendelin Wiedeking. "Wiedeking: "Education is the Strongest Profit-yielding Form of Investment.", November 3 2009. URL http://www.porsche.com/international/aboutporsche/pressreleases/archive2009/quarter1/?pool=international-de&id=2009-03-11. (Retrieved: 06/02/15).

[181] Georgios N. Yannakakis and Manolis Maragoudakis. Player Modeling Impact on Player's Entertainment in Computer Games. 3538:74–78, 2005.

[182] Nick Yee. Motivations for Play in Online Games. *CyberPsychology & Behavior*, 9(6):772–775, 2006.

[183] José P. Zagal, Jochen Rick, and Idris Hsi. Collaborative Games: Lessons Learned From Board Games. *Simulation and Gaming*, 37(1):24–40, 2006. ISSN 1046-8781. doi: http://dx.doi.org/10.1177/1046878105282279.

[184] Natalia P. Zea, Francisco L. Gutiérrez, Nuria M. Medina, and José L.G. Sánchez. Adapting Contents and Procedures in Educational Video Games with Collaborative Activities. In *Proceedings of the 1st International Open Workshop on Intelligent Personalization and Adaptation in Digital Educational Games*, 2009.

[185] Natalia P. Zea, José L. G. Sánchez, and Francisco L. Gutiérrez. Collaborative Learning by Means of Video Games: An Entertainment System in the Learning Processes. In *ICALT '09: Proceedings of the 2009 Ninth IEEE International Conference on Advanced Learning Technologies*, pages 215–217, Washington, DC, USA, 2009. IEEE Computer Society.

[186] Natalia P. Zea, José L. G. Sánchez, Francisco L. Gutiérrez, Marcelino J. Cabrera, and P. Paderewski. Design of Educational Multiplayer Videogames: A Vision From Collaborative Learning. *Advances in Engineering Software*, 40(12):1251–1260, 2009.

[187] Natalia P. Zea, R. Lopez-Arcos, Francisco L. Gutiérrez-Vela, P. Paderewski, and Nuria M. Medina. Architecture for Monitoring Learning Processes using Video Games. In *Learning and Intelligent Optimization*, volume 7997, 2011.

[188] Birgit Zimmermann. *Pattern-basierte Prozessbeschreibung und -unterstützung: Ein Werkzeug zur Unterstützung von Prozessen zur Anpassung von E-Learning-Materialien*. PhD thesis, TU Darmstadt - Elektrotechnik und Informationstechnik - Multimedia Kommunikation, 2008.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

GM	Game Master
STEM	Science, Technology, Engineering, and Mathematics
MINT	Mathematics, Information Sciences, Natural Sciences, and Technology
OECD	Organisation for Economic Co-operation and Development
CEO	Chief Executive Officer
PISA	Programme for International Student Assessment
EFWI	Escape From Wilson Island
MMOG	Massively Multiplayer Online Game
MMO(RP)G	Massively Multiplayer Online (Role-play) Game
MMORPG	Massively Multiplayer Online Role-play Game
BMBF	Bundesministerium für Bildung und Forschung
RPG	Role-playing Game
CSCL	Computer-supported Collaborative Learning
P2P	Peer-to-Peer
ADHS	Attention Deficit Hyperactivity Disorder
NPC	Non-Player Character
AI	Artificial Intelligence
FSM	Finite State Machine
CbKST	Competency-based Knowledge Space Theory
NGLOB	Narrative Game-based Learning Object
CMSG	Collaborative Multiplayer Serious Game
UML	Unified Modeling Language
AO	Adaptation Object
HCI	Human-Computer-Interaction
GUI	Graphical User Interface

A EFWI ADAPTATION DETAILS

»The opposite of play is not work. It's depression.«

— Brian Sutton-Smith

THIS APPENDIX contains details about the defined game variables, actions, events, tasks, regions, situations, situation objects, adaptations, and adaptation objects for *Escape From Wilson Island*.

A.1 EFWI GAME VARIABLES

In the following Tables 31, 32, and 33, a complete list of the accessed game variables of EFWI is given. These variables are used to define the actions, events, tasks, regions, situations, and adaptations. The tables contain the variables' names, data type, value domain, and an explanation, where required.

Variable Name	Type	Value Range	Description
GameTime	Float	$[0, \infty]$	total game time
PalmsFelled	Integer	$[0, 1000]$	total number of felled palms
BerriesGathered	Integer	$[0, 1000]$	total number of gathered berries
FelledPalmsInWorld	Integer	$[0, 100]$	felled palms currently available
HalfPalmsInWorld	Integer	$[0, 100]$	half palms currently available
SmallPalmsInWorld	Integer	$[0, 100]$	quarter palms currently available
PalmsCarried	Integer	$[0, 1000]$	total number of palms lifted
PalmsDropped	Integer	$[0, 1000]$	total number of palms dropped
HutBuilt	Boolean		
FirewoodChopped	Integer	$[0, 100]$	total number of firewood chopped
FirewoodInWorld	Integer	$[0, 100]$	total number of firewood available
BottleFound	Boolean		
BottleFilled	Boolean		
FlashlightState	Boolean		true = on, false = off
IsNight	Boolean		true = night, false = day
NPCTalkState	Short	$[0, 24]$	information about which conversations with the NPC are already done
HeronsHunted	Integer	$[0, 100]$	total number of herons hunted
MeatGathered	Integer	$[0, 100]$	total meat gathered
MeatCooked	Integer	$[0, 100]$	total meat cooked
RawMeatAvailable	Integer	$[0, 100]$	raw meat in world
CookedMeatAvailable	Integer	$[0, 100]$	cooked meat in world
RaftBuild	Boolean		
PlayersOnRaft	Short	$[0, 4]$	total number of players on raft
SignalFire	Boolean		Game win condition

Table 31: EFWI game variables. For reasons of clarity and comprehensibility, the value range is left empty when the type is 'boolean' as in this case it is always $\{\mathtt{true}, \mathtt{false}\}$.

Parameter Name	Type	Value Range	Description
Health	Float	$[0, 100]$	current health
Satiety	Float	$[0, 100]$	current satiety
Fitness	Float	$[0, 100]$	current fitness
HasAxe	Boolean		
HasFlashlight	Boolean		
HasRope	Boolean		
HasPaper	Boolean		
HasMap	Boolean		
HasWilson	Boolean		
HasEmpty	Boolean		
HasFilled	Boolean		
Berries	Boolean		number of berries in inventory
Firewood	Boolean		number of firewood in inventory
RawMeat	Boolean		number of raw meat in inventory
CookedMeat	Boolean		number of cooked meat in inventory
IsMoving	Boolean		
IsGatheringBerries	Boolean		
IsPickingUpBottle	Boolean		
IsFellingPalm	Boolean		
IsCarryingPalm	Boolean		
TotalMoveDistance	Float	$[0, \infty]$	total distance moved by this player

Table 32: EFWI player parameters. For reasons of clarity and comprehensibility, the value range is left empty when the type is 'boolean' as in this case it is always $\{true, false\}$.

Parameter Name	Type	Value Range	Description
HealthLoss	Float	$[0, 1]$	health loss multiplicator
HealthGain	Float	$[0, 1]$	helth gain multiplicator
FitnessLoss	Float	$[0, 1]$	fitness loss multiplicator
FitnessGain	Float	$[0, 1]$	fitness gain multiplicator
SatietyLoss	Float	$[0, 1]$	sturation loss multiplicator
SatietyGain	Float	$[0, 1]$	saturation gain multiplicator
BerryProb	Float	$[0, 1]$	gather berry success probability
CarryPalmtolerance	Float	$[0, 1]$	tolerance radius for carrying a palm
HeronFleeRadius	Float	$[0, 1]$	tolerance radius for the heron to flee
ScatterPalmProb	Float	$[0, 1]$	probability for a palm to scatter when it is dropped

Table 33: EFWI game adaptation variables - configuration variables used to adapt the game.

A.2 EFWI ACTIONS AND EVENTS

The following Table 34 contains a complete list of EFWI game actions and events which are accessed by GameAdapt.KOM.

Action Name	Type	Description
GatherBerry	Player Action	Player adds berries from bush to inventory
EatBerry	Player Action	Player eats berry from inventory
GetHeronMeat	Player Action	Player picks meat up
EatRawMeat	Player Action	Player eats raw meat
CookRawMeat	Player Action	Player cooks meat at fire
EatCookedMeat	Player Action	Player eats cooked meat
FellPalm	Player Action	Player fells palm
ChopPalm	Player Action	Player splits felled palm
PickupFirewood	Player Action	Player adds firewood to inventory
PickUpPalm	Player Action	Player starts carrying
DropPalm	Player Action	Player stops carrying
AddPalmToHut	Player Action	Player adds palm to hut
AddFireWood	Player Action	Player adds firewood to fire
PickupBottle	Player Action	Player adds bottle to inventory
FillBottle	Player Action	Player removes bottle from and adds filled bottle to inventory
SwitchFlashlight	Player Action	Player toggles flashlight on or off
TalkToHank	Player Action	Player initiates dialog with NPC
Walk	Player Action	Walking action
OpenBox	Player Action	Player opens shared inventory
PutItemIntoBox	Player Action	Player adds item from inventory to shared inventory
RemoveItemFromBox	Player Action	Player adds item from shared inventory to inventory
StepOnRaft	Player Action	Player steps on the raft
GoToSleep	Player Action	Player starts sleeping phase for all players
UpdatePlayerStats	Game Event	Applies hunger, fitness, and health changes
NewDayStarted	Game Event	Apply regeneration effects
HeronDied	Game Event	Removes heron and spawns meat
PalmDropped	Game Event	Checks if palm breaks
BottleFound	Game Event	Notifies that a player found the bottle

Table 34: EFWI actions and events.

A.3 EFWI TASKS

Table 35 contains a complete list of EFWI tasks. For each task, a list of its states, the related actions or events, the related shapes and the successor state is denoted.

State Name	Action/Event a	Shape s	successor state σ_{new}
KeepSaturationHigh			
σ_0	UpdatePlayerStats	$\text{Saturation} < 25$	σ_{Hunger}
σ_0	UpdatePlayerStats	$\text{Saturation} \geqslant 25 \wedge$ $\text{Saturation} \leqslant 75$	σ_0
σ_0	UpdatePlayerStats	$\text{Saturation} > 75$	$\sigma_{saturated}$
σ_{Hunger}	UpdatePlayerStats	$\text{Saturation} < 25$	σ_{Hunger}
σ_{Hunger}	UpdatePlayerStats	$\text{Saturation} \geqslant 25$	σ_0
$\sigma_{saturated}$	UpdatePlayerStats	$\text{Saturation} \leqslant 75$	σ_0
$\sigma_{saturated}$	UpdatePlayerStats	$\text{Saturation} > 75$	$\sigma_{saturated}$
BuildLogHut			
σ_0	AddPalm	$\text{HutBuild} = \text{false}$	$\sigma_{1stPalmAdded}$
$\sigma_{1stPalmAdded}$	AddPalm	$\text{HutBuild} = \text{false}$	$\sigma_{2ndPalmAdded}$
$\sigma_{2ndPalmAdded}$	AddPalm	$\text{HutBuild} = \text{false}$	$\sigma_{HutFinished}$
BuildRaft			
σ_0	AddHalfPalm	$\text{RaftBuild} = \text{false}$	$\sigma_{1stPalmAdded}$
$\sigma_{1stPalmAdded}$	AddHalfPalm	$\text{RaftBuild} = \text{false}$	$\sigma_{2ndPalmAdded}$
$\sigma_{2ndPalmAdded}$	AddHalfPalm	$\text{RaftBuild} = \text{false}$	$\sigma_{3rdPalmAdded}$
$\sigma_{3rdPalmAdded}$	AddHalfPalm	$\text{RaftBuild} = \text{false}$	$\sigma_{RaftFinished}$
GetSeamap			
σ_0	TalkToHank	$\text{NPCTalkState} = 20$ $\wedge \text{HasPaper} = \text{true}$	$\sigma_{GotSeamap}$
HuntHeron			
σ_0	HeronDied	–	σ_1
σ_1	HeronDied	–	σ_2
...	...	–	...
σ_4	HeronDied	–	$\sigma_{5ormore}$
$\sigma_{5ormore}$	HeronDied	–	$\sigma_{5ormore}$
FillGasBottle			
σ_0	PickupBottle	–	$\sigma_{BottleFound}$
$\sigma_{BottleFound}$	FillBottle	–	$\sigma_{BottleFilled}$
EscapeFast			
σ_0	StepOnRaft	$\text{PlayersOnRaft} = 4$	$\sigma_{Rafting}$
$\sigma_{Rafting}$	LeaveRaft	$\text{PlayersOnRaft} = 0 \wedge$ IsOn2ndIsland	$\sigma_{On2ndISland}$
$\sigma_{On2ndIsland}$	AddPalm	$\text{SignalFire} = \text{true}$	$\sigma_{GameWon}$

Table 35: EFWI tasks. The Tasks *KeepFitnessHigh* and *KeepHealthHigh* are analogue to *KeepSaturationHigh*. For reasons of clarity they are omitted here.

A.4 EFWI REGIONS

The defined regions which are used in situation criterias are shown in Table 36. For each region, the name, the type of region, and the defining parameters are provided.

Region Name	Type	Coordinates
CloseHutArea	Circle	x:420;z:500;radius:10
WideHutArea	Circle	x:420;z:500;radius:20
HeronHideArea	Circle	x:650;z:220;radius:30
HeronCliffArea	Rectangle	x0:600;z0:120;x1:750;z1:270
BushArea1	Rectangle	x0:350;z0:550;x1:450;z1:650
BushArea2	Rectangle	x0:690;z0:180;x1:7600;z1:220
BottleArea	Circle	x:330;z:380;radius:10
GeyserArea	Circle	x:440;z:370;radius:100
RaftBuildArea	Circle	x:560;z:590;radius: 10
Beach	ComplexArea	distance to water < 10

Table 36: EFWI regions.

A.5 EFWI SITUATIONS

Following the definitions of situations and situation objects in Chapter 5, in Tables 37, 38, 39, 40, 41, and 42, all situations are provided, that were used in EFWI to recognize player intentions. Each situation is described by a unique identifier, its name, followed by a 'P' to indicate that the situation applies to a single player or a 'G' to indicate that the situation applies to the whole group. Subsequently, a description text is provided. Finally, the list of criteria describing the situation is provided.

Based on the list of situation, in Tables 43, 44, and 45, a complete list of resulting situation objects is provided. Each situation object refers to one situation, defines a cooldown, the effect on the challenge vector, and the effect on the related skills.

Name (P/G)	Idle(P)
Description	A player is idle
Criteria	[2]AtomicCriterion : IsMoving = false [1]AtomicCriterion : HasMovedIn(10) = 0
Name (P/G)	Exploring(P)
Description	A player is exploring the island
Criteria	[2]AtomicCriterion : IsMoving = true [2]AtomicCriterion : HasMovedIn(60) > 50 [2]AtomicCriterion : HasMovedIn(60) > 100 [2]AtomicCriterion : HasMovedIn(60) > 200
Name (P/G)	SearchingForBerries(P)
Description	A player is searching for berries
Criteria	[2]RegionCriterion : BushArea1 or BushArea2 [2]TaskCriterion : KeepSatHigh : state = lightHunger [4]TaskCriterion : KeepSatHigh : state = strongHunger [2]IntervalCriterion : GatherBerries(30)
Name (P/G)	GatheringBerries(P)
Description	A player is gathering berries
Criteria	AtomicCriterion : IsGatheringBerries = true
Name (P/G)	FellingPalm(P)
Description	A player is felling a palm
Criteria	[0]AtomicCriterion : HasAxe = true [3]RegionCriterion : Beach [1]AtomicCriterion : IsMoving = false [4]AtomicCriterion : IsFellingPalm = true
Name (P/G)	SearchingForBottle(P)
Description	A player is searching for the bottle
Criteria	[0]TaskCriterion : FillGasBottle : state = σ_0 [1]AtomicCriterion : IsMoving = true [2]RegionCriterion : Beach
Name (P/G)	SearchingForGeysir(P)
Description	A player is searching for the geysir
Criteria	[0]TaskCriterion : FillGasBottle : state = $\sigma_{\text{BottleFound}}$ [1]AtomicCriterion : IsNight = true [2]RegionCriterion : Geysir

Table 37: EFWI situations 1/6.

Name (P/G)	PickingUpBottle(P)
Description	A player found the bottle
Criteria	$[0]AtomicCriterion : IsPickingUpBottle = true$ $[1]AtomicCriterion : IsNight = true$ $[2]RegionCriterion : Geysir$ $DistanceCriterion : (FlashlightPlayer, \leqslant 10)$
Name (P/G)	FillingBottle(P)
Description	A player is searching for the bottle
Criteria	$[0]TaskCriterion : FillGasBottle : state = \sigma_{BottleFound}$ $[1]AtomicCriterion : IsNight = true$ $[2]RegionCriterion : Geysir$ $DistanceCriterion : (FlashlightPlayer, \leqslant 10)$
Name (P/G)	CarryingPalm(G)
Description	The player is carrying a palm
Criteria	$[1]AtomicCriterion : IsCarryingPalm = true$
Name (P/G)	BuildingHut(G)
Description	The players are building the hut
Criteria	$[0]TaskCriterion : BuildLogHut : state = !\sigma_{HutFinished}$ $[2]AtomicCriterion : CarryingPalm = true$ $[2]RegionCriterion : CloseHutArea$ $[1]RegionCriterion : WideHutArea$
Name (P/G)	BuildingRaft(G)
Description	The players are building the raft
Criteria	$[0]TaskCriterion : BuildRaft : state = !\sigma_{RaftFinished}$ $[2]AtomicCriterion : CarryingHalfPalm = true$ $[2]RegionCriterion : RaftBuildArea$
Name (P/G)	TalkingToHank(P)
Description	The player is talking to Hank
Criteria	$[2]AtomicCriterion : IsCarryingPalm = true$ $[1]AtomicCriterion : IsMoving = false$ $[1]DistanceCriterion : (Hank, \leqslant 5)$

Table 38: EFWI situations 2/6.

Name (P/G)	NoPalmsFelled(P)
Description	No palm felled for too long
Criteria	[0]AtomicCriterion : HasAxe = true [0]AtomicCriterion : GameTime > 120 AtomicCriterion : PalmsFelled = 0
Name (P/G)	PalmDropped(P)
Description	The player dropped the palm
Criteria	[1]ActionCriterion : PalmDropped
Name (P/G)	HutBuildingNotStarted(G)
Description	Failed starting to build the hut
Criteria	[1]TaskCriterion : BuildLoghut : state = σ_0 [0]AtomicCriterion : GameTime > 300
Name (P/G)	HutBuildingNotFinished(G)
Description	Building the hut takes too long
Criteria	[1]TaskCriterion : BuildLoghut : state! = σ_0 [1]TaskCriterion : BuildLoghut : state! = HutFinished [0]AtomicCriterion : GameTime > 600
Name (P/G)	PlayerIsHungry(P)
Description	The player's saturation is low
Criteria	[1]TaskCriterion : KeepSaturationHigh : state = $\sigma_{lightHunger}$
Name (P/G)	PlayerIsTired(P)
Description	The player's fitness is low
Criteria	[1]TaskCriterion : KeepFitnessHigh : state = $\sigma_{lightTiredness}$
Name (P/G)	PlayerIsHurt(P)
Description	The player's health is low
Criteria	[1]TaskCriterion : KeepHealthHigh : state = $\sigma_{lightHurt}$
Name (P/G)	FirstHeronTooLate(G)
Description	The group takes too long to hunt the first heron
Criteria	[1]TaskCriterion : HuntHeron : state = σ_0 [0]AtomicCriterion : Gametime > 1200

Table 39: EFWI situations 3/6.

NAME (P/G)	OtherHeronTooLate(G)
DESCRIPTION	The group takes too long to hunt further heron
CRITERIA	$[1]\mathrm{TaskCriterion} : \mathrm{HuntHeron} : \mathrm{state} = \sigma_1$ $[0]\mathrm{AtomicCriterion} : \mathrm{Gametime} > 1800$
NAME (P/G)	FindBottleTooLate(G)
DESCRIPTION	The group takes too long to find the bottle
CRITERIA	$[1]\mathrm{TaskCriterion} : \mathrm{FillGasBottle} : \mathrm{state} = \sigma_1$ $[0]\mathrm{AtomicCriterion} : \mathrm{Gametime} > 900$
NAME (P/G)	FillBottleTooLate(G)
DESCRIPTION	The group takes too long to fill the bottle
CRITERIA	$[1]\mathrm{TaskCriterion} : \mathrm{FillGasBottle} : \mathrm{state} = \sigma_{\mathrm{BottleFound}}$ $[0]\mathrm{AtomicCriterion} : \mathrm{Gametime} > 1800$
NAME (P/G)	TalkToHankTooLate(P)
DESCRIPTION	The player takes too long to talk to Hank
CRITERIA	$[1]\mathrm{AtomicCriterion} : \mathrm{NPCTalkState} = 0$ $[0]\mathrm{AtomicCriterion} : \mathrm{Gametime} > 900$
NAME (P/G)	GetSeamapTooLate(P)
DESCRIPTION	The player takes too long to get the seamap
CRITERIA	$[1]\mathrm{TaskCriterion} : \mathrm{GetSeamap} : \mathrm{state}! = \sigma_{\mathrm{GotSeamap}}$ $[0]\mathrm{AtomicCriterion} : \mathrm{Gametime} > 1200$
NAME (P/G)	BuildRaftTooLate(G)
DESCRIPTION	The group takes too long to build the raft
CRITERIA	$[1]\mathrm{TaskCriterion} : \mathrm{BuildRaft} : \mathrm{state}! = \sigma_{\mathrm{RaftFinished}}$ $[0]\mathrm{AtomicCriterion} : \mathrm{Gametime} > 1800$

Table 40: EFWI situations 4/6.

Name (P/G)	PlayerIsSaturated(P)
Description	The player's saturation is high
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{KeepSaturationHigh} : \mathtt{state} = \sigma_{\mathtt{saturated}}$
Name (P/G)	PlayerIsFit(P)
Description	The player's fitness is high
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{KeepFitnessHigh} : \mathtt{state} = \sigma_{\mathtt{fit}}$
Name (P/G)	PlayerIsHealthy(P)
Description	The player's health is high
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{KeepHealtHigh} : \mathtt{state} = \sigma_{\mathtt{healthy}}$
Name (P/G)	BottleFoundQuick(G)
Description	The group found the bottle quickly
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{FillGasBottle} : \mathtt{state} = \sigma_{\mathtt{BottleFound}}$ [0]$\mathtt{AtomicCriterion} : \mathtt{Gametime} < 150$
Name (P/G)	BottleFilledQuick(G)
Description	The group filled the bottle quickly
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{FillGasBottle} : \mathtt{state} = \sigma_{\mathtt{BottleFilled}}$ [0]$\mathtt{AtomicCriterion} : \mathtt{Gametime} < 900$
Name (P/G)	HutBuild(G)
Description	The group built the log hut
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{BuildLogHut} : \mathtt{state} = \sigma_{\mathtt{HutBuild}}$
Name (P/G)	HutBuildQuick(G)
Description	The group built the log hut quickly
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{BuildLogHut} : \mathtt{state} = \sigma_{\mathtt{HutBuild}}$ [0]$\mathtt{AtomicCriterion} : \mathtt{Gametime} < 600$
Name (P/G)	RaftBuild(G)
Description	The group built the raft
Criteria	[1]$\mathtt{TaskCriterion} : \mathtt{BuildRaft} : \mathtt{state} = \sigma_{\mathtt{RaftBuild}}$

Table 41: EFWI situations 5/6.

NAME (P/G)	RaftBuildQuick(G)
DESCRIPTION	The group built the raft quickly
CRITERIA	[1]$\mathtt{TaskCriterion} : \mathtt{BuildRaft} : \mathtt{state} = \sigma_{\mathtt{RaftBuild}}$ [0]$\mathtt{AtomicCriterion} : \mathtt{Gametime} < 1200$
NAME (P/G)	TalkToHankQuick(P)
DESCRIPTION	The player talked to Hank quickly
CRITERIA	[1]$\mathtt{AtomicCriterion} : \mathtt{NPCTalkState} > 0$ [0]$\mathtt{AtomicCriterion} : \mathtt{Gametime} < 30$
NAME (P/G)	FirstHeronQuick(G)
DESCRIPTION	The group managed to hunt the first heron quickly
CRITERIA	[1]$\mathtt{TaskCriterion} : \mathtt{HuntHeron} : \mathtt{state!} = \sigma_0$ [0]$\mathtt{AtomicCriterion} : \mathtt{Gametime} < 600$
NAME (P/G)	LotsOfBerries(P)
DESCRIPTION	The player has lots of berries in the inventory
CRITERIA	[1]$\mathtt{InventoryCriterion} : \mathtt{Berries} \geqslant 10$
NAME (P/G)	LotsOfHerons(G)
DESCRIPTION	The group managed to hunt many herons
CRITERIA	[1]$\mathtt{TaskCriterion} : \mathtt{HuntHeron} : \mathtt{state!} = \sigma_{\mathtt{5ormore}}$
NAME (P/G)	LotsOfPalms(P)
DESCRIPTION	The player fell lots of palms
CRITERIA	[0]$\mathtt{AtomicCriterion} : \mathtt{HasAxe} = \mathtt{true}$ [1]$\mathtt{AtomicCriterion} : \mathtt{PalmsFelled} \geqslant 6$
NAME (P/G)	PalmCarryEasy(P)
DESCRIPTION	The player did not drop palms
CRITERIA	[0]$\mathtt{AtomicCriterion} : \mathtt{PalmsLifted} > 0$ [1]$\mathtt{AtomicCriterion} : \mathtt{PalmsDropped} = 0$ [0]$\mathtt{AtomicCriterion} : \mathtt{GameTime} > 600$

Table 42: EFWI situations 6/6.

Situation	CD(s)	Effect on player model	Effect on challenge	Effect on associated skills
FellingPalm	0	$(0.05, 0.05, -0.02, -0.04, -0.04)$	$\begin{pmatrix} \mathtt{BuildLogHut} & -0.1 \end{pmatrix}$	$\begin{pmatrix} \mathtt{KnowledgeWood} & +0.1 \end{pmatrix}$
GatheringBerries	0	$(-0.02, 0.05, 0.05, -0.04, -0.04)$		$\begin{pmatrix} \mathtt{KnowledgeBerries} & +0.25 \end{pmatrix}$
PickingUpBottle	0			$\begin{pmatrix} \mathtt{KnowledgeBottle} & +1.0 \\ \mathtt{FindBottle} & +1.0 \end{pmatrix}$
FillingBottle	0		$\begin{pmatrix} \mathtt{FillBottle} & = 0.0 \end{pmatrix}$	$\begin{pmatrix} \mathtt{FillBottle} & +1.0 \\ \mathtt{Teamwork} & +0.3 \end{pmatrix}$
CarryingPalm	1			$\begin{pmatrix} \mathtt{CarryPalm} & +0.01 \\ \mathtt{Teamwork} & +0.01 \\ \mathtt{Communication} & +0.01 \end{pmatrix}$
HuntingHeron	1			$\begin{pmatrix} \mathtt{HuntHeron} & +0.01 \\ \mathtt{Teamwork} & +0.01 \\ \mathtt{Communication} & +0.01 \end{pmatrix}$
HutBuild	0		$\begin{pmatrix} \mathtt{BuildLogHut} & = 0.0 \end{pmatrix}$	$\begin{pmatrix} \mathtt{KnowledgeWood} & +1.0 \\ \mathtt{CarryPalm} & +1.0 \\ \mathtt{BuildHut} & +1.0 \end{pmatrix}$
RaftBuild	0		$\begin{pmatrix} \mathtt{BuildRaft} & = 0.0 \end{pmatrix}$	$\begin{pmatrix} \mathtt{KnowledgeRaft} & +1.0 \end{pmatrix}$
TalkToHank	0	$(-0.01, -0.01, -0.07, 0.10, -0.01)$		$\begin{pmatrix} \mathtt{KnowledgeWilson} & +0.1 \\ \mathtt{KnowledgeRaft} & +0.1 \\ \mathtt{KnowledgeWood} & +0.1 \\ \mathtt{KnowledgeHunt} & +0.1 \\ \mathtt{KnowledgeBottle} & +0.1 \end{pmatrix}$
Exploring	10	$(0.10, -0.06, -0.07, -0.07, 0.10)$	$\begin{pmatrix} \mathtt{FindBottle} & -0.05 \end{pmatrix}$	

Table 43: EFWI situation objects 1/3.

Situation	CD(s)	Effect on player model	Effect on challenge	Effect on associated skills
SearchingForBerries	5	$(-0.02, 0.05, 0.05, -0.04, -0.04)$		$\begin{pmatrix} \mathit{KnowledgeBerries} & +0.1 \end{pmatrix}$
SearchingForBottle	10	$(-0.02, 0.05, 0.05, -0.04, -0.04))$		$\begin{pmatrix} \mathit{KnowledgeBottle} & +0.1 \end{pmatrix}$
SearchingForGeysir	10	$(-0.02, 0.05, +0.05, -0.04, -0.04))$		$\begin{pmatrix} \mathit{KnowledgeGeysir} & +0.1 \end{pmatrix}$
NoPalmsFelled	60		$\begin{pmatrix} \mathit{BuildLogHut} & +0.1 \\ \mathit{EscapeFast} & +0.05 \end{pmatrix}$	$\begin{pmatrix} \mathit{KnowledgeWood} & -0.1 \end{pmatrix}$
PalmDropped	0		$\begin{pmatrix} \mathit{CarryPalm} & +0.2 \end{pmatrix}$	$\begin{pmatrix} \mathit{CarryPalm} & -0.2 \end{pmatrix}$
HutBuildingNotStarted	60		$\begin{pmatrix} \mathit{BuildLogHut} & +0.1 \end{pmatrix}$	$\begin{pmatrix} \mathit{BuildHut} & -0.1 \end{pmatrix}$
HutBuildingNotFinished	60		$\begin{pmatrix} \mathit{BuildLogHut} & +0.1 \end{pmatrix}$	$\begin{pmatrix} \mathit{BuildHut} & -0.1 \end{pmatrix}$
PlayerIsHungry	30		$\begin{pmatrix} \mathit{KeepSatHigh} & +0.1 \end{pmatrix}$	
PlayerIsTired	30		$\begin{pmatrix} \mathit{KeepFitnessHigh} & +0.1 \end{pmatrix}$	
PlayerIsHurt	30		$\begin{pmatrix} \mathit{KeepHealthHigh} & +0.1 \end{pmatrix}$	
FirstHeronTooLate	60		$\begin{pmatrix} \mathit{HuntHeron} & +0.1 \end{pmatrix}$	$\begin{pmatrix} \mathit{HuntHeron} & -0.1 \end{pmatrix}$
OtherHeronTooLate	60		$\begin{pmatrix} \mathit{HuntHeron} & +0.1 \end{pmatrix}$	$\begin{pmatrix} \mathit{HuntHeron} & -0.1 \end{pmatrix}$
FindBottleTooLate	60		$\begin{pmatrix} \mathit{FillBottle} & +0.1 \end{pmatrix}$	$\begin{pmatrix} \mathit{FindBottle} & -0.1 \end{pmatrix}$
FillBottleTooLate	60		$\begin{pmatrix} \mathit{FillBottle} & +0.1 \end{pmatrix}$	$\begin{pmatrix} \mathit{FillBottle} & -0.1 \end{pmatrix}$
TalkToHankTooLate	60	$(+0.02, +0.02, +0.04, -0.10, +0.02)$		$\begin{pmatrix} \mathit{KnowledgeWilson} & -0.1 \\ \mathit{KnowledgeRaft} & -0.1 \\ \mathit{KnowledgeWood} & -0.1 \\ \mathit{KnowledgeHunt} & -0.1 \\ \mathit{KnowledgeBottle} & -0.1 \end{pmatrix}$
GetSeamapTooLate	60	$(+0.00, +0.00, +0.00, +0.00, +0.00)$	$\begin{pmatrix} \mathit{GetSeamap} & +0.1 \end{pmatrix}$	

Table 44: EFWI situation objects 2/3.

Situation	CD(s)	Effect on player model	Effect on challenge	Effect on associated skills
BuildRaftTooLate	60		$\begin{pmatrix}\mathtt{BuildRaft} & +0.1\end{pmatrix}$	$\begin{pmatrix}\mathtt{BuildRaft} & -0.1\end{pmatrix}$
PlayerIsSaturated	30		$\begin{pmatrix}\mathtt{KeepSatHigh} & -0.1\end{pmatrix}$	
PlayerIsFit	30		$\begin{pmatrix}\mathtt{KeepFitnessHigh} & -0.1\end{pmatrix}$	
PlayerIsHealthy	30		$\begin{pmatrix}\mathtt{KeepHealthHigh} & -0.1\end{pmatrix}$	
BottleFoundQuick	0		$\begin{pmatrix}\mathtt{FillBottle} & -0.5\end{pmatrix}$	$\begin{pmatrix}\mathtt{FindBottle} & +1.0\end{pmatrix}$
BottleFilledQuick	0		$\begin{pmatrix}\mathtt{FillBottle} & -0.5\end{pmatrix}$	$\begin{pmatrix}\mathtt{FillBottle} & +1.0\end{pmatrix}$
FillBottleTooLate	60		$\begin{pmatrix}\mathtt{FillBottle} & +0.1\end{pmatrix}$	$\begin{pmatrix}\mathtt{FillBottle} & -0.1\end{pmatrix}$
HutBuildQuick	0		$\begin{pmatrix}\mathtt{BuildHut} & -0.5\end{pmatrix}$	$\begin{pmatrix}\mathtt{KnowledgeWood} & +1.0\\ \mathtt{CarryPalm} & +1.0\\ \mathtt{BuildHut} & +1.0\end{pmatrix}$
RaftBuildQuick	0		$\begin{pmatrix}\mathtt{BuildRaft} & -0.5\end{pmatrix}$	$\begin{pmatrix}\mathtt{KnowledgeRaft} & +1.0\end{pmatrix}$
TalkToHankQuick	60	$(-0.02, -0.02, -0.04, 0.10, -0.02)$		$\begin{pmatrix}\mathtt{KnowledgeWilson} & -0.1\\ \mathtt{KnowledgeRaft} & -0.1\\ \mathtt{KnowledgeWood} & -0.1\\ \mathtt{KnowledgeHunt} & -0.1\\ \mathtt{KnowledgeBottle} & -0.1\end{pmatrix}$
FirstHeronQuick	0		$\begin{pmatrix}\mathtt{HuntHeron} & -0.5\end{pmatrix}$	$\begin{pmatrix}\mathtt{HuntHeron} & +0.5\end{pmatrix}$
LotsOfBerries	0		$\begin{pmatrix}\mathtt{KeepSatHigh} & -0.25\end{pmatrix}$	$\begin{pmatrix}\mathtt{KnowledgeBerries} & +0.5\end{pmatrix}$
LotsOfHerons	0		$\begin{pmatrix}\mathtt{KeepSatHigh} & -0.25\end{pmatrix}$	$\begin{pmatrix}\mathtt{KnowledgeHunt} & +1.0\\ \mathtt{HuntHeron} & +1.0\end{pmatrix}$
LotsOfPalms	0		$\begin{pmatrix}\mathtt{BuildLogHut} & -0.25\end{pmatrix}$	$\begin{pmatrix}\mathtt{KnowledgeWood} & +1.0\end{pmatrix}$
PalmCarryEasy	0		$\begin{pmatrix}\mathtt{CarryPalms} & -0.5\end{pmatrix}$	$\begin{pmatrix}\mathtt{CarryPalm} & +0.5\end{pmatrix}$

Table 45: EFWI situation objects 3/3.

A.6 EFWI ADAPTIONS

Following the definition of adaptations and adaptation objects in Chapter 5, Tables 46, 47, 48, 49, 50, 51, 52, and 53 contain a complete list of all adaptations that were defined to adapt EFWI. Each adaptation contains a unique identifier (name), a description, a specification of the game element it adapts, and parameters describing the effect.

Based on the defined adaptations, a complete list of adaptation objects is provided in Tables 54, 55, and 56. The adaptation objects refer to an adaptation, define possible prerequisites, an effect on the challenge vector, and an effect on related skills.

Name	DecrFitnessLoss
Description	decreases the amount of fitness loss per minute
Game Element	Game Variable: FitnessLoss
Parameter(s)	-0.1
Name	IncrFitnessLoss
Description	increases the amount of fitness loss per minute
Game Element	Game Variable: FitnessLoss
Parameter(s)	$+0.1$
Name	DecrFitnessGain
Description	decreases the amount of saturation gained by sleeping
Game Element	Game Variable: FitnessGain
Parameter(s)	-0.1
Name	IncrFitnessGain
Description	increases the amount of saturation gained by sleeping
Game Element	Game Variable: FitnessGain
Parameter(s)	$+0.1$
Name	Message_TipSleeping
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"Sleeping restores fitness! Click your loghut to go sleeping."
Name	Message_TipSleepingBuildHut
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"Sleeping restores fitness! Build a loghut so you have a place to sleep!"

Table 46: EFWI adaptations - Fitness.

Name	DecrSatietyLoss
Description	decreases the amount of satiety loss per minute
Game Element	Game Variable: SatietyLoss
Parameter(s)	-0.1
Name	IncrSatietyLoss
Description	increases the amount of satiety loss per minute
Game Element	Game Variable: SatietyLoss
Parameter(s)	$+0.1$
Name	DecrSatietyGain
Description	decreases the amount of satiety gained by eating
Game Element	Game Variable: SatietyGain
Parameter(s)	-0.1
Name	IncrSatietyGain
Description	increases the amount of satiety gained by eating
Game Element	Game Variable: SatietyGain
Parameter(s)	$+0.1$
Name	IncreaseBerryProb
Description	increases the probability of successfully gathering berries
Game Element	Game Variable: BerryProb
Parameter(s)	$+0.1$
Name	DecreaseBerryProb
Description	decreases the probability of successfully gathering berries
Game Element	Game Variable: BerryProb
Parameter(s)	-0.1
Name	Message_TipBerries
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"Those red berries on those bushes look eatable!"
Name	Message_TipTradeFood
Description	sends a notification to one player
Game Element	Game event: Player_Notification
Parameter(s)	"Maybe one of your teammates can give you some food if you ask nicely."
Name	GiveBerriesToPlayer
Description	gives 5 berries to the player
Game Element	Game action: Give_Items_to_Player
Parameter(s)	<Berries, 5>

Table 47: EFWI adaptations - Satiety.

Name	DecrHealthLoss
Description	decreases the amount of health loss per minute
Game Element	Game Variable: HealthLoss
Parameter(s)	-0.1
Name	IncrHealthLoss
Description	increases the amount of health loss per minute
Game Element	Game Variable: HealthLoss
Parameter(s)	$+0.1$
Name	IncrHealthGain
Description	increases the amount of health gained by sleeping and eating
Game Element	Game Variable: HealthGain
Parameter(s)	$+0.1$
Name	DecrHealthGain
Description	decreases the amount of health gained by sleeping and eating
Game Element	Game Variable: HealthGain
Parameter(s)	-0.1
Name	Message_TipHealing
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"Remember: You cannot swim. Also, starving will affect your health. Eating and Sleeping well heal you."

Table 48: EFWI adaptations - Health.

Name	DecrPalmCarryTolerance
Description	decreases the tolerance radius for carrying palms
Game Element	Game Variable: CarryPalmTolerance
Parameter(s)	-0.1
Name	IncrPalmCarryTolerance
Description	increases the tolerance radius for carrying palms
Game Element	Game Variable: CarryPalmTolerance
Parameter(s)	$+0.1$
Name	IncrPalmScatterProb
Description	increases the probability for a dropped palm to scatter
Game Element	Game Variable: PalmScatterProb
Parameter(s)	$+0.1$
Name	DecrPalmScatterProb
Description	decreases the probability for a dropped palm to scatter
Game Element	Game Variable: PalmScatterProb
Parameter(s)	-0.1
Name	Message_TipCarryPalm
Description	sends a notification to all players
Game Element	Game event: Player_Notification
Parameter(s)	"You need to coordinate yout movement to not drop the palm. Walk at the same speed and in the same direction."
Name	Message_TipFellPalms
Description	sends a notification to a player
Game Element	Game event: Player_Notification
Parameter(s)	"You can fell palms and use those palms to build the log hut."
Name	MessageAll_TipBuildHut
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You need to build the log hut! Fell palms and carry them towards the marked center of the island."

Table 49: EFWI adaptations - Build log hut.

Name	Message_TipInfoSeamap
Description	sends a notification to a player
Game Element	Game event: Player_Notification
Parameter(s)	"You can get the seamap from Hank. Go talk to him!"

Table 50: EFWI adaptations - Seamap.

Name	Message_TipInfoFindBottle
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You need to find a bottle and fill it with gas for the signal fire."
Name	Message_TipSearchBottle
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You still need to find a bottle and fill it with gas for the signal fire. Maybe search at the beach."
Name	Message_TipInfoFillBottle
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You need to fill the bottle. Find the blue geysir at night and fill the bottle while a teammate provides light for you using a flashlight."
Name	Message_TipSearchGeysir
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You need to fill the bottle. Find the blue geysir at night and fill the bottletogether with a teammate."

Table 51: EFWI adaptations - Bottle.

Name	DecrPalmCarryTolerance (Repeated for clarity)
Description	decreases the tolerance radius for carrying palms
Game Element	Game Variable: CarryPalmTolerance
Parameter(s)	-0.1
Name	IncrPalmCarryTolerance (Repeated for clarity)
Description	increases the tolerance radius for carrying palms
Game Element	Game Variable: CarryPalmTolerance
Parameter(s)	$+0.1$
Name	Message_TipGetThread
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"Hank will give you a thread to build a raft if you find his volleyball."
Name	Message_TipRaft
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You need to build the the raft to reach the other island. Fell palms and carry them towards the beach."

Table 52: EFWI adaptations - Build raft

Name	DecrHeronFleeRadius
Description	decreases the radius of available space needed for the heron to flee
Game Element	Game Variable: HeronFleeRadius
Parameter(s)	-0.1
Name	IncrHeronFleeRadius
Description	increases the radius of available space needed for the heron to flee
Game Element	Game Variable: HeronFleeRadius
Parameter(s)	$+0.1$
Name	MessageA_TipInfoHeron
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You need to hunt a heron to get better food and to find Hank's volleyball!"
Name	Message_TipInfoHunt
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You need to work together to hunt the heron. Surround it with at least three playersto push it towards the cliff."
Name	Message_TipInfoFire
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You can light a fire near your hut. This will help you sleep better thus increasing your regenereation effects."
Name	Message_TipInfoCook
Description	sends a notification to all players
Game Element	Game event: Player_Notification_All
Parameter(s)	"You can use the fire near your hut to cook meat. Cooked meat is much better for your saturation than uncooked meat or berries."

Table 53: EFWI adaptations - Hunt heron.

Adaptation	Prerequisite	Player model	Effect on challenge	Effect on skills
IncrSatLoss	$\mathit{SatietyLoss} < 1.0$		$\begin{pmatrix} \mathit{KeepSatHigh} & +0.1 \end{pmatrix}$	
DecrSatLoss	$\mathit{SatietyLoss} > 0.0$		$\begin{pmatrix} \mathit{KeepSatHigh} & -0.1 \end{pmatrix}$	
IncrSatGain	$\mathit{SatietyGain} < 1.0$		$\begin{pmatrix} \mathit{KeepSatHigh} & -0.1 \end{pmatrix}$	
DecrSatGain	$\mathit{SatietyGain} > 0.0$		$\begin{pmatrix} \mathit{KeepSatHigh} & +0.1 \end{pmatrix}$	
IncrBerryProb	$\mathit{BerryProb} < 1.0$		$\begin{pmatrix} \mathit{KeepSatHigh} & +0.1 \end{pmatrix}$	
DecrBerryProb	$\mathit{BerryProb} > 0.0$		$\begin{pmatrix} \mathit{KeepSatHigh} & -0.1 \end{pmatrix}$	
Message_TipBerries				$\begin{pmatrix} \mathit{KnowledgeBerries} & +0.5 \end{pmatrix}$
Message_Tip TradeFood				$\begin{pmatrix} \mathit{Communication} & +0.2 \end{pmatrix}$
GiveBerriestoPlayers			$\begin{pmatrix} \mathit{KeppSatHigh} & -0.1 \end{pmatrix}$	
IncrFitnessLoss	$\mathit{FitnessLoss} < 1.0$		$\begin{pmatrix} \mathit{KeepFitnessHigh} & +0.1 \end{pmatrix}$	
DecrFitnessLoss	$\mathit{FitnessLoss} > 0.0$		$\begin{pmatrix} \mathit{KeepFitnessHigh} & -0.1 \end{pmatrix}$	
IncrFitnessGain	$\mathit{FitnessGain} < 1.0$		$\begin{pmatrix} \mathit{KeepFitnessHigh} & -0.1 \end{pmatrix}$	
DecrFitnessGain	$\mathit{FitnessGain} > 0.0$		$\begin{pmatrix} \mathit{KeepFitnessHigh} & +0.1 \end{pmatrix}$	
IncrHealthLoss	$\mathit{HealthLoss} < 1.0$		$\begin{pmatrix} \mathit{KeepHealthHigh} & +0.1 \end{pmatrix}$	
DecrHealthLoss	$\mathit{HealthLoss} > 0.0$		$\begin{pmatrix} \mathit{KeepHealthHigh} & -0.1 \end{pmatrix}$	
IncrHealthGain	$\mathit{HealthGain} < 1.0$		$\begin{pmatrix} \mathit{KeepHealthHigh} & -0.1 \end{pmatrix}$	
DecrHealthGain	$\mathit{HealthGain} > 0.0$		$\begin{pmatrix} \mathit{KeephealthHigh} & +0.1 \end{pmatrix}$	
Message_TipHealing	$\mathit{HutBuild} = \mathit{false}$			$\begin{pmatrix} \mathit{KnowledgeSwim} & +0.2 \\ \mathit{KnowledgeEat} & +0.2 \\ \mathit{SleepInHut} & +0.2 \end{pmatrix}$

Table 54: EFWI adaptation objects 1/3.

Adaptation	Prerequisite	Player model	Effect on challenge	Effect on skills
DecrPalmCarryTolerance			$\begin{pmatrix} \mathtt{CarryPalm} & +0.1 \end{pmatrix}$	
IncrPalmCarryTolerance			$\begin{pmatrix} \mathtt{CarryPalm} & -0.1 \end{pmatrix}$	
IncrPalmScatterProb			$\begin{pmatrix} \mathtt{CarryPalm} & +0.1 \\ \mathtt{BuildHut} & +0.1 \\ \mathtt{BuildRaft} & +0.1 \end{pmatrix}$	
DecrPalmScatterProb			$\begin{pmatrix} \mathtt{CarryPalm} & -0.1 \\ \mathtt{BuildHut} & -0.1 \\ \mathtt{BuildRaft} & -0.1 \end{pmatrix}$	
MessageTipCarryPalm			$\begin{pmatrix} \mathtt{BuildHut} & -0.1 \\ \mathtt{BuildRaft} & -0.1 \end{pmatrix}$	$\begin{pmatrix} \mathtt{CarryPalm} & +0.3 \\ \mathtt{Teamwork} & +0.3 \end{pmatrix}$
MessageTipFellPalms	$\mathtt{PalmsFelled} = 0$			$\begin{pmatrix} \mathtt{KnowledgeWood} & +0.5 \end{pmatrix}$
MessageTipBuildHut	$\mathtt{PalmsFelled} > 0$		$\begin{pmatrix} \mathtt{BuildHut} & -0.3 \end{pmatrix}$	$\begin{pmatrix} \mathtt{KnowledgeWood} & +0.3 \\ \mathtt{BuildHut} & +0.5 \end{pmatrix}$
MessageInfoSeamap			$\begin{pmatrix} \mathtt{GetSeamap} & -0.2 \end{pmatrix}$	$\begin{pmatrix} \mathtt{KnowledgeRaft} & +0.3 \end{pmatrix}$
MessageTipGetThread			$\begin{pmatrix} \mathtt{BuildRaft} & -0.1 \end{pmatrix}$	$\begin{pmatrix} \mathtt{KnowledgeWilson} & +0.5 \end{pmatrix}$
MessageTipBuildRaft			$\begin{pmatrix} \mathtt{BuildRaft} & -0.3 \end{pmatrix}$	$\begin{pmatrix} \mathtt{KnowledgeWood} & +0.3 \\ \mathtt{KnowledgeRaft} & +0.5 \\ \mathtt{BuildRaft} & +0.2 \end{pmatrix}$
Message_TipSleeping	$\mathtt{HutBuild} = \mathtt{true}$			$\begin{pmatrix} \mathtt{SleepInHut} & +0.2 \end{pmatrix}$
Message_TipSleeping BuildHut	$\mathtt{HutBuild} = \mathtt{false}$			$\begin{pmatrix} \mathtt{BuildHut} & +0.2 \\ \mathtt{SleepInHut} & +0.2 \end{pmatrix}$

Table 55: EFWI adaptation objects 2/3.

Adaptation	Prerequisite	Player model	Effect on challenge	Effect on skills
MessageTipFindBottle	$\mathit{BottleFound} = \mathit{false}$			$\begin{pmatrix} \mathit{KnowledgeBottle} & +0.5 \end{pmatrix}$
MessageSearchBottle	$\mathit{BottleFound} = \mathit{false}$			$\begin{pmatrix} \mathit{FindBottle} & +0.5 \end{pmatrix}$
MessageSearchGeysir	$\mathit{BottleFound} = \mathit{true}$			$\begin{pmatrix} \mathit{KnowledgeGeysir} & +0.5 \end{pmatrix}$
MessageTipFillBottle	$\mathit{BottleFound} = \mathit{true}$			$\begin{pmatrix} \mathit{FillBottle} & +0.5 \end{pmatrix}$
DecrHeronFleeRadius			$\begin{pmatrix} \mathit{HuntHeron} & +0.1 \end{pmatrix}$	
IncrHeronFleeRadius			$\begin{pmatrix} \mathit{HuntHeron} & -0.1 \end{pmatrix}$	
MessageTipInfoHeron				$\begin{pmatrix} \mathit{KnowledgeHunt} & +0.5 \end{pmatrix}$
MessageTipInfoHunt				$\begin{pmatrix} \mathit{HuntHeron} & +0.5 \\ \mathit{Teamwork} & +0.3 \end{pmatrix}$
Message_TipBuildFire	$\mathit{HutBuild} = \mathit{true}$			$\begin{pmatrix} \mathit{BuildFire} & +0.5 \end{pmatrix}$
Message_TipBuildCook	$\mathit{HutBuild} = \mathit{true}$ $\mathit{HeronsHunted} > 0$		$\begin{pmatrix} \mathit{KeepSatHigh} & -0.2 \end{pmatrix}$	$\begin{pmatrix} \mathit{CookHeron} & +0.5 \end{pmatrix}$

Table 56: EFWI adaptation objects 3/3.

B EVALUATION DETAILS

»The best way to learn is to do; the worst way to teach is to talk.«

— Paul Halmos

This appendix contains details about the evaluations described in Chapter 9. The data shown here is the foundation of the qualitative and quantitative analysis in Chapter 9.

B.1 Player agent simulation $2^k r$ factorial design response values of the variation of the initial skill configurations

Table 58 shows the influence of the parameters A, B, C, and D on the observed response values *health*, *satiety*, *fitness*, *berries gathered*, *meat roasted*, *hut build time*, *bottle found time*, and *bottle filled time*. The influence is given as the proportion of the total variation of the respective response. It is shown for all four parameters, and all interactions of the parameters, and the error of measurement. A, B, C, and D are the sets of skills as shown in Table 57. For the low value, i.e., '$-$', all skills of the related set are set to 0.0. For the high value, i.e., '$+$', all skills of the related set are set to 1.0. Table 59 contains the absolute effect of the variation of A, B, C, and D on the mean response values.

Set	Task	Skills
A	Berries	KnowledgeBerries
B	Palm	KnowledgeWood, CarryPalm, BuildHut, SleepInHut
C	Bottle	KnowledgeGeysir, KnowledgeBottle, FindBottle, FillBottle
D	Hunt	KnowledgeHunt, HuntHeron, CookHeron, BuildFire
E	Raft	KnowledgeRaft, KnowledgeWilson, BuildRaft

Table 57: EFWI skill clusters.

Table 60 contains the response values for the variation of the player model with the same response values.

Table 61 shows the influence of the variation of the interaction skills on the observed response values and Table 62 contains the absolute effect of the variation of the interaction skills on the mean response values.

System Parameter	Health	Satiety	Fitness	Berries Gathered	Herons Hunted	Meat Roasted	HutB. Time	Bot.Found Time	Bot.Filled Time
	Proportion of variation								
A	0.03	0.31	0.01	0.08	< 0.01	0.00	0.00	0.00	0.03
B	0.90	0.26	0.89	0.63	0.27	0.51	0.92	0.00	0.02
C	< 0.01	0.01	< 0.01	0.02	< 0.01	0.03	< 0.01	0.87	0.51
D	0.02	0.07	0.01	0.03	< 0.01	< 0.01	0.01	0.01	0.01
$A * B$	< 0.01	0.01	< 0.01	0.01	0.05	0.01	< 0.01	0.01	0.01
$A * C$	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
$A * D$	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
$B * C$	< 0.01	0.01	< 0.01	0.02	0.03	< 0.01	< 0.01	< 0.01	< 0.01
$B * D$	< 0.01	0.01	< 0.01	0.01	< 0.01	< 0.01	0.01	< 0.01	0.02
$C * D$	< 0.01	< 0.01	< 0.01	0.01	0.01	0.01	< 0.01	0.01	0.01
$A * B * C$	< 0.01	0.01	< 0.01	< 0.01	0.01	0.03	< 0.01	0.01	0.01
$A * B * D$	< 0.01	0.03	< 0.01	< 0.01	< 0.01	0.01	< 0.01	< 0.01	< 0.01
$A * C * D$	< 0.01	0.01	< 0.01	0.02	< 0.01	0.01	< 0.01	< 0.01	< 0.01
$B * C * D$	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.02
$A * B * C * D$	0.01	0.01	0.01	< 0.01	< 0.01	0.02	0.01	< 0.01	0.01
Error	0.03	0.26	0.07	0.17	0.61	0.37	0.06	0.08	0.34

Table 58: $2^k * r$ factorial design response proportion for the initial skill variation.

	Absolute Influence on Response												
		A			B			C			D		
		$mean$		F	$mean$		F	$mean$		F	$mean$		F
	mean	$A = 0$	$A = 1$		$B = 0$	$B = 1$		$C = 0$	$C = 1$		$D = 0$	$D = 1$	
Health	65.02	61.68	68.36	29.10	47.06	82.98	841.60	65.09	64.94	0.02	67.66	62.37	18.25
	±19.14	±19.38	±18.69	p<0.01	±6.04	±6.19	p<0.01	±19.27	±19.42	p=0.91	±20.30	±17.93	p<0.01
Satiety	51.96	46.77	57.15	37.83	47.13	56.79	32.73	50.88	53.04	1.63	54.41	49.51	8.44
	±9.49	±8.88	±6.99	p<0.0 1	±9.24	±7.06	p<0.01	±10.57	±8.35	p=0.21	±8.70	±9.78	p<0.01
Fitness	31.46	29.44	33.47	2.77	7.34	55.57	396.21	31.11	31.80	0.08	34.10	28.82	4.74
	±25.86	±14.93	±27.15	p=0.11	±6.00	±10.83	p<0.01	±25.40	±26.86	p=0.78	±27.81	±24.07	p=0.04
Berries Gathered	102.21	92.92	111.50	13.949	129.17	75.25	117.42	106.71	97.71	3.27	108.04	96.38	5.50
	±34.22	±32.49	±34.02	p<0.01	±18.11	±23.41	p<0.01	±31.26	±37.06	p=0.08	±27.12	±30.73	p=0.03
Herons Hunted	5.29	5.33	5.25	0.029	4.38	6.21	13.93	5.21	5.38	0.12	5.33	5.25	0.03
	±1.80	±1.27	±2.23	p=0.87	±1.61	±1.50	p<0.01	±1.64	±1.97	p=0.74	±1.86	±1.78	0.87
Meat Roasted	10.23	9.75	10.71	0.30	4.33	16.13	44.64	8.83	11.62	2.50	10.04	10.42	0.05
	±8.31	±8.31	±8.47	p=0.59	±6.39	±5.28	p<0.01	±7.72	±8.81	p=0.1	±9.00	±7.76	0.83
Hut Build Time (s)	1546	1593	1500	1.36	2461	632	532.58	1564	1529	0.20	1471	1621	3.60
	±964	±965	±981	p=0.25	±294	±258	p<0.01	±981	±967	p=0.7	±1056	±879	p=0.07
Bot. Found Time (s)	612	651	573	1.81	612	612	0.00	1161	63	357.71	550	674	4.55
	±593	±631	±565	p=0.19	±594	±606	p=0.97	±302	±1	p<0.01	±532	±655	0.04
Bot. Filled Time (s)	1133	1265	1000	3.31	1104	1162	0.16	1640	625	48.42	1051	1214	1.24
	±717	±757	±665	p=0.08	±665	±780	p=0.69	±456	±553	p<0.01	±715	±726	0.27

Table 59: $2^k * r$ factorial design absolute response values for the initial skill variation.

System Parameter	Mean Values: Set 1 (ambiti.) mean	SD	Set 2 (curious) mean	SD	Set 3 (acting) mean	SD	Set 4 (interac.) mean	SD	Set 5 (moving) mean	SD	Set 6 (mixed) mean	SD	Set 7 (average) mean	SD
Health	74.10	3.80	54.20	3.87	59.66	11.08	49.98	0.06	56.24	11.03	56.70	11.22	65.02	19.13
Satiety	65.63	1.93	67.86	3.12	71.90	4.19	62.52	3.47	66.15	4.79	64.91	4.04	51.96	9.49
Fitness	41.28	5.94	9.98	3.76	21.16	17.51	5.76	0.05	14.06	13.81	14.85	15.04	31.46	25.86
Berries Gathered	101.00	8.54	140.33	7.09	130.00	16.70	97.33	7.37	127.67	25.70	122.67	20.55	102.21	34.22
Herons Hunted	5.00	1.00	4.00	2.00	8.00	0.00	8.00	1.00	5.67	1.53	3.33	1.53	5.29	1.80
Meat Roasted	15.67	3.21	6.00	4.00	11.33	10.02	5.00	1.00	11.67	10.02	5.67	6.66	10.23	8.31
Hut Build Time (s)	817	137	2161	335	1766	539	1650	154	1747	509	2020	753	1546	964
Bot. Found Time (s)	1049	143	808	135	2078	787	1559	246	1082	268	732	218	612	593
Bot. Filled Time (s)	1682	660	1226	355	2504	339	1768	382	2531	292	944	33	1124	717

Table 60: Influence of the player model variation on the response values.

System Parameter	Proportion of variation Health	Sat.	Fitn.	Berries Gath.	Herons Hunted	Meat Roast.	HutB. Time	Bottle FoundT.	Bottle FilledT.
IA	0.03	< 0.01	0.01	0.07	0.13	0.02	0.29	0.01	0.21
Skills	0.87	0.69	0.84	0.65	0.44	0.75	0.59	0.87	0.63
IA $*$ Skills	0.01	< 0.01	10.01	0.05	0.02	0.02	0.10	0.01	0.02
Error	0.10	0.30	0.13	0.423	0.41	0.21	0.02	0.12	0.14

Table 61: $2^k * r$ factorial design response proportion for the initial interaction skill variation.

		Absolute Influence on Response IA			Skills		
	mean	$mean$ IA = 0	IA = 1	F	$mean$ Skills = 0	Skills = 1	F
Health	48.52 ±8.96	48.47 ±8.96	48.57 ±10.69	1.60 p=0.24	41.03 ±5.00	56.01 ±5.91	71.29 p<0.01
Satiety	59.80 ±17.02	57.48 ±17.02	62.12 ±18.81	0.00 p=0.98	44.33 ±2.33	75.26 ±8.78	18.14 p<0.01
Fitness	25.91 ±22.45	23.36 ±22.45	28.47 ±25.76	0.80 p=0.40	5.55 ±0.03	46.27 ±13.73	50.67 p<0.01
Berries Gathered	107.08 ±24.41	114.50 ±24.41	99.67 ±34.09	2.43 p=0.16	129.67 ±8.78	84.50 ±24.26	22.51 p<0.01
Herons Hunted	5.25 ±1.97	4.67 ±1.97	5.83 ±1.33	2.45 p=0.16	4.17 ±1.17	6.33 ±1.51	8.45 p=0.02
Meat Roasted	7.92 ±9.24	6.67 ±9.24	9.17 ±10.52	0.72 p=0.42	0.00 ±0.00	15.83 ±7.03	28.74 p<0.01
Hut Build Time (s)	2008 ±356	2386 ±355.68	1631 ±852	91.98 p<0.01	2549 ±186	1468 ±678	189.02 p<0.01
Bottle Found Time (s)	668 ±655	611 ±655.49	725 ±761	0.51 p=0.49	1273 ±369	63 ±1	p=58.18 p<0.01
Bottle Filled Time (s)	1356 ±500	1668 ±500	1044 ±778	12.32 p=0.01	1889 ±443	822 ±458	36.02 p<0.01

Table 62: $2^k * r$ factorial design absolute response values for the initial interaction skill variation.

B.2 GAME ADAPTATION SYSTEM PARAMETERS

The following Table 63 shows the proportion of influence of the varied system parameters β, γ, and δ on the response values, which are the player performance values. Table 64 contains the absolute variation of the mean value depending on the varied parameters.

Table 65 shows the proportion of influence of the varied system parameters β, γ, and δ on the challenge. Table 66 contains the absolute variation of the mean value depending on the varied parameters.

Table 67 shows the proportion of influence of the varied system parameters β, γ, and δ on the time to learn the skills. Table 68 contains the absolute variation of the mean value depending on the varied parameters.

	Proportion of variation									
System Parameter	Health	Sat.	Fitn.	Berries Gath.	Herons Hunted	Meat Roast.	HutB. Time	Bottle FoundT.	Bottle FilledT.	Raft Built
β	0.01	0.02	0.01	0.09	0.18	0.00	0.01	0.00	0.02	0.04
γ	0.48	0.45	0.62	0.47	0.02	0.35	0.78	0.28	0.02	0.65
δ	0.03	0.02	0.02	0.01	0.04	0.00	0.01	0.03	0.01	0.01
$\beta * \gamma$	0.02	0.01	0.07	0.07	0.01	0.08	0.01	0.00	0.00	0.01
$\beta * \delta$	0.02	0.00	0.02	0.00	0.02	0.02	0.00	0.00	0.01	0.00
$\gamma * \delta$	0.01	0.01	0.01	0.03	0.20	0.02	0.01	0.06	0.04	0.00
$\beta * \gamma * \delta$	0.03	0.02	0.00	0.01	0.04	0.04	0.00	0.00	0.00	0.01
Error	0.40	0.47	0.25	0.31	0.51	0.49	0.17	0.62	0.90	0.28

Table 63: $2^k * r$ factorial design player performance response proportion for the game adaptation system parameters variation.

		Absolute Influence on Response											
		Challenge (β)				Learning(γ)				Interaction(δ)			
		$mean$		F	p	$mean$		F	p	$mean$		F	p
System Parameter	mean	$\beta = 0$	$\beta = 1$			$\gamma = 0$	$\gamma = 1$			$\delta = 0$	$\delta = 1$		
Health	55.63	57.20	50.20	0.71	0.41	44.13	67.13	38.10	< 0.01	58.47	52.79	2.33	0.14
Saturation	48.80	47.41	54.26	1.14	0.29	41.53	56.08	31.01	< 0.01	50.16	47.45	1.07	0.31
Fitness	25.75	28.20	23.30	1.91	0.18	10.05	41.45	78.70	< 0.01	28.58	22.92	2.56	0.12
Berries Collected	89.60	98.05	81.15	8.84	0.01	109.35	69.85	48.30	< 0.01	92.10	87.10	0.77	0.39
Herons Hunted	4.68	5.40	3.95	11.21	< 0.01	4.90	4.45	1.08	0.31	4.35	5.00	2.25	0.14
Meat Roasted	7.75	7.30	8.20	0.28	0.60	3.65	11.85	23.16	< 0.01	7.85	6.65	0.01	0.91
Hut Build Time (s)	1686	1790	1582	2.68	0.11	2446	926	143.50	< 0.01	1603	1768	1.69	0.20
Bottle Found Time (s)	1083	1094	1073	0.02	0.88	1342	824	14.57	0.00	1163	1003	1.69	0.20
Bottle Filled Time (s)	1650	1729	1571	0.64	0.43	1725	1576	0.57	0.46	1720	1581	1.38	0.25
Raft Built	0.45	0.35	0.55	4.57	0.04	0.05	0.85	73.14	< 0.01	0.40	0.50	1.14	0.29

Table 64: $2^k * r$ factorial design response absolute response values for the game adaptation system parameters variation.

	Proportion of variation									
	Keep Sat.	Keep Fitn.	Keep Health	Build	Build	Hunt	Get	Find	Fill	Carry
System Parameter	High	High	High	Loghut	Raft	Heron	Seamap	Bottle	Bottle	Palm
β	0.20	0.02	0.01	0.17	0.00	0.03	0.08	0.16	0.01	0.04
γ	0.09	0.60	0.71	0.58	0.98	0.01	0.12	0.43	0.17	0.04
δ	0.01	0.03	0.02	0.00	0.00	0.02	0.01	0.00	0.00	0.00
$\beta * \gamma$	0.02	0.07	0.02	0.00	0.00	0.00	0.13	0.11	0.00	0.04
$\beta * \delta$	0.03	0.02	0.00	0.04	0.00	0.02	0.05	0.02	0.02	0.00
$\gamma * \delta$	0.05	0.01	0.00	0.01	0.00	0.13	0.00	0.00	0.02	0.00
$\beta * \gamma * \delta$	0.00	0.00	0.01	0.01	0.00	0.02	0.00	0.01	0.00	0.00
Error	0.61	0.25	0.23	0.19	0.02	0.78	0.61	0.27	0.78	0.89

Table 65: $2^k * r$ factorial design challenge response proportion for the game adaptation system parameters variation.

		Absolute Influence on Response											
		Challenge (β)				Learning(γ)				Interaction(δ)			
		$mean$		F	p	$mean$		F	p	$mean$		F	p
System Parameter	mean	$\beta = 0$	$\beta = 1$			$\gamma = 0$	$\gamma = 1$			$\delta = 0$	$\delta = 1$		
Keep Satiety High	-0.25	-0.36	-0.13	10.35	< 0.01	-0.17	-0.33	4.75	0.04	-0.22	-0.27	0.33	0.57
Keep Fitness High	0.48	0.43	0.52	2.08	0.16	0.76	0.20	78.42	< 0.01	0.42	0.54	3.93	0.06
Keep Health High	-0.26	-0.28	-0.25	0.75	0.39	-0.05	-0.48	99.62	< 0.01	-0.30	-0.23	3.01	0.09
Build Loghut	0.58	0.67	0.50	28.38	< 0.01	0.74	0.42	99.28	< 0.01	0.58	0.59	0.07	0.79
Build Raft	-0.27	-0.26	-0.29	3.12	0.09	0.23	-0.78	2072.58	< 0.01	-0.28	-0.27	0.01	0.93
Hunt Heron	-0.44	-0.50	-0.39	1.16	0.29	-0.48	-0.41	0.42	0.05	-0.40	-0.49	0.71	0.41
Get Seamap	0.10	0.11	0.09	4.32	0.05	0.11	0.09	6.10	0.02	0.10	0.10	0.36	0.56
Find Bottle	-0.92	-0.93	-0.91	18.73	< 0.01	-0.93	-0.90	49.72	< 0.01	-0.92	-0.92	0.01	0.94
Fill Bottle	0.27	0.29	0.24	0.44	0.51	0.36	0.17	7.08	0.01	0.27	0.26	0.01	0.94
Carry Palm	0.00	0.00	0.01	1.30	0.26	0.00	0.01	1.27	0.27	0.01	0.04	0.01	0.93

Table 66: $2^k * r$ factorial design response challenge values for the game adaptation system parameters variation.

	Proportion of variation								
System Parameter	Cook Heron	Hunt Heron	Knowl. Hunt	Gather Berries	Fill Bottle	Knowl. Geysir	Find Bottle	Knowl. Bottle	Sleep In Hut
β	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.01
γ	0.52	0.79	0.79	0.78	0.54	0.52	0.91	0.88	0.74
δ	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.01
$\beta * \gamma$	0.01	0.05	0.05	0.00	0.00	0.01	0.00	0.01	0.00
$\beta * \delta$	0.02	0.06	0.00	0.02	0.00	0.02	0.00	0.00	0.00
$\gamma * \delta$	0.07	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.00
$\beta * \gamma * \delta$	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00
Error	0.33	0.07	0.07	0.18	0.44	0.43	0.07	0.09	0.24

System Parameter	Build Fire	Build Hut	Build Raft	Carry Palm	Knowl. Wood	Knowl. Raft	Knowl. Wilson	Team-work	Commu-nication
β	0.00	0.27	0.00	0.07	0.01	0.00	0.01	0.02	0.18
γ	0.87	0.38	0.99	0.70	0.51	0.97	0.84	0.33	0.02
δ	0.01	0.01	0.00	0.01	0.05	0.00	0.01	0.21	0.27
$\beta * \gamma$	0.00	0.27	0.00	0.02	0.06	0.00	0.01	0.25	0.02
$\beta * \delta$	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.12	0.40
$\gamma * \delta$	0.01	0.01	0.00	0.00	0.01	0.00	0.01	0.01	0.00
$\beta * \gamma * \delta$	0.00	0.00	0.00	0.01	0.02	0.00	0.02	0.00	0.00
Error	0.11	0.06	0.00	0.19	0.31	0.02	0.07	0.06	0.10

Table 67: $2^k * r$ factorial design skill learning time response proportion for the game adaptation system parameters variation.

		Absolute Influence on Response											
		Challenge (β)				Learning(γ)				Interaction(δ)			
		$mean$		F	p	$mean$		F	p	$mean$		F	p
System Parameter	mean	$\beta = 0$	$\beta = 1$			$\gamma = 0$	$\gamma = 1$			$\delta = 0$	$\delta = 1$		
CookHeron	1332	1231	1434	5.17	0.30	1649	1016	50.30	< 0.01	1369	1295	0.69	0.41
HuntHeron	974	776	1173	24.45	< 0.01	776	1173	367.74	< 0.01	1069	880	5.51	0.03
KnowledgeHunt	352	337	368	0.42	0.52	640	65	141.55	< 0.01	399	306	3.70	0.06
GatherBerry	314	317	311	0.06	0.81	465	164	141.00	< 0.01	304	324	0.62	0.44
FillBottle	2147	2125	2168	0.06	0.81	2700	1593	39.23	0.01	2176	2117	0.11	0.74
KnowledgeGeysir	1138	1127	1149	0.05	0.83	1467	809	38.61	< 0.01	1134	1142	0.01	0.94
FindBottle	1168	1100	1236	2.20	0.15	2119	218	430.40	< 0.01	1229	1108	1.77	0.19
KnowledgeBottle	795	723	868	2.99	0.09	1524	66	302.63	< 0.01	858	733	2.24	0.15
SleepInHut	1071	1001	1140	1.61	0.21	1615	525	98.54	< 0.01	1020	1121	0.86	0.36
BuildFire	1760	1800	1720	0.58	0.45	2596	923	255.47	< 0.01	1678	1841	2.44	0.13
BuildHut	620	895	344	140.19	< 0.01	950	289	201.43	< 0.01	561	678	6.29	0.02
BuildRaft	1498	1530	1466	7.71	0.01	2589	407	9079.42	< 0.01	1497	1500	0.02	0.89
CarryPalm	1261	1512	1010	11.97	< 0.01	2050	473	118.37	< 0.01	1150	1373	2.37	0.13
KnowledgeWood	75	78	73	0.69	0.41	97	54	52.74	< 0.01	82	68	5.34	0.23
KnowledgeRaft	1194	1257	1131	4.43	0.04	98	62	18.59	< 0.01	91	60	27.68	< 0.01
KnowledgeWilson	662	722	602	3.76	0.06	2339	49	1464.24	< 0.01	1230	1158	1.44	0.24
Teamwork	1068	944	1193	10.01	< 0.01	1589	548	175.74	< 0.01	1490	647	115.05	< 0.01
Communication	2027	1630	2423	59.77	< 0.01	2157	1896	6.50	0.02	2510	1543	88.99	< 0.01

Table 68: $2^k * r$ factorial design response skill learning time values for the game adaptation system parameters variation.

B.3 GAME ADAPTATION EFFECTIVENESS USING SIMULATED PLAYERS

The following Tables 69, 70, and 71 contain the data of the evaluation of the game adaptation effectiveness using simulated players. Table 69 contains the comparison of the player performance values between the groups with adaptation and without adaptation. Table 70 contains the comparison of the challenge values between the groups with adaptation and without adaptation. Table 71 contains the comparison of the times to learn the skills between the groups with adaptation and without adaptation.

	No Adaptation						With Adaptation					
Start Skills	0.0		0.5		1.0		0.0		0.5		1.0	
	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD
Health	44.13	0.89	64.08	13.29	72.75	18.51	80.79	13.19	61.78	15.82	83.69	16.56
Satiety	45.82	5.37	52.84	4.99	56.40	4.12	54.57	4.45	54.42	3.34	55.34	4.90
Fitness	8.05	5.59	24.50	14.01	55.15	13.43	41.14	6.24	43.38	7.95	52.51	10.07
Berries Gathered	125.00	8.97	118.60	21.73	50.60	13.05	66.80	11.80	49.20	6.83	48.20	5.54
Herons Hunted	6.20	2.05	4.00	0.00	3.80	1.10	4.00	1.41	3.40	.55	3.40	.89
Meat Roasted	.60	1.34	5.80	5.59	11.20	3.70	12.00	6.04	10.60	1.34	10.00	3.08
Hut Build Time (s)	460	1031	1336	849	584	215	777	106	743	153	595	209
Bottle Found Time (s)	1357	277	1300	398	63	2	660	202	138	93	63	1
Bottle Filled Time (s)	1665	242	1505	330	662	370	1634	282	834	616	303	175
Raft Build	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
Raft Build Time (s)	–	0.00	–	0.00	1699	427	2062	422	1707	227	1556	222

Table 69: Game adaptation effectiveness - comparison of the player performance values between the groups with adaptation and without adaptation using simulated players.

	No Adaptation						With Adaptation					
Start Skills	0.0		0.5		1.0		0.0		0.5		1.0	
	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD
Keep Satiety High	-0.29	0.24	-0.15	0.12	-0.43	0.21	-0.26	0.15	-0.18	0.15	-0.23	0.12
Keep Fitness High	0.82	0.06	0.54	0.25	-0.05	0.24	0.17	0.10	0.22	0.15	0.05	0.24
Keep Health High	-0.02	0.05	-0.23	0.18	-0.61	0.10	-0.45	0.10	-0.38	0.10	-0.51	0.15
Build LogHut	0.81	0.01	0.79	0.01	0.32	0.30	0.30	0.07	0.26	0.12	0.03	0.11
Build Raft	0.24	0.00	0.25	0.01	0.00	0.08	-0.81	0.06	-0.87	0.03	-0.88	0.01
Hunt Heron	-0.48	0.25	-0.07	0.11	-0.84	0.04	-0.57	0.31	-0.82	0.03	-0.69	0.32
Get Seamap	0.12	0.00	0.12	0.00	0.07	0.04	0.07	0.03	0.04	0.02	0.03	0.01
Find Bottle	-0.94	0.00	-0.92	0.01	-0.90	0.02	-0.90	0.02	-0.87	0.02	-0.88	0.02
Fill Bottle	0.44	0.15	0.41	0.25	-0.96	0.01	0.06	0.09	-0.75	0.42	-0.96	0.00
Carry Palm	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.01	0.04	0.01	0.04

Table 70: Game adaptation effectiveness - comparison of the average challenge values between the groups with adaptation and without adaptation using simulated players.

	No Adaptation						With Adaptation					
	0.0		0.5		1.0		0.0		0.5		1.0	
Start Skills	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD
CookHeron	1898	503	1353	190	0	0	1037	88	875	195	0	0
HuntHeron	1635	658	1258	282	0	0	495	177	215	46	0	0
KnowledgeHunt	658	69	1013	275	0	0	614	376	68	9	0	0
GatherBerry	848	101	191	53	0	0	227	38	145	24	0	0
FillBottle	2700	0	2700	0	0	0	1074	196	956	0	0	0
KnowledgeGeysir	1484	290	1356	225	0	0	833	172	863	317	0	0
FindBottle	2204	274	1975	592	0	0	387	82	1949	1095	0	0
KnowledgeBottle	1502	321	1399	212	0	0	890	570	71	14	0	0
SleepInHut	1673	485	806	532	0	0	675	195	532	156	0	0
BuildFire	2681	42	2168	507	0	0	952	105	746	147	0	0
BuildHut	1880	120	434	82	0	0	435	37	314	2	0	0
BuildRaft	2700	0	2231	147	0	0	569	20	212	9	0	0
CarryPalm	2410	464	877	311	0	0	626	124	218	32	0	0
KnowledgeWood	126	10	71	6	0	0	91	12	10	21	0	0
KnowledgeRaft	2510	425	2630	83	0	0	213	13	10	21	0	0
KnowledgeWilson	957	269	638	430	0	0	198	26	13	29	0	0
Teamwork	2688	17	2461	184	0	0	357	57	203	31	0	0
Communication	2700	0	2700	0	0	0	2406	658	441	198	0	0

Table 71: Game adaptation effectiveness - comparison of the average times to learn the skills between the groups with adaptation and without adaptation using simulated players.

B.4 GAME ADAPTATION EFFECTIVENESS WITH REAL PLAYERS

The following Tables 72, 73, and 74 contain the evaluation data of the game adaptation effectiveness evaluation using real players. Table 72 shows the comparison between the two groups with adaptation and without adaptation regarding player performance values. Table 73 shows the comparison between the two groups with adaptation and without adaptation regarding perceived game experience. Table 74 shows the comparison between the two groups with adaptation and without adaptation regarding perceived teamwork and collaboration.

Response	Adapt.	Mean	SD	F	p
Health	0	44.93	8.75	11.55	0.01
	1	68.83	13.07		
Satiety	0	16.36	10.48	18.66	<0.01
	1	42.74	8.75		
Fitness	0	29.37	22.34	2.48	0.15
	1	47.54	12.91		
Berries Gathered	0	93.20	35.26	2.19	0.18
	1	67.20	17.43		
Herons Hunted	0	0.40	0.89	8.10	0.02
	1	2.20	1.10		
Meat Roasted	0	7.60	3.05	22.07	<0.01
	1	9.40	4.16		
Hut Build Time	0	1566.20	912.60	6.18	0.04
	1	508.80	268.54		
Bottle Found Time	0	2648.20	115.83	131.69	<0.01
	1	694.60	362.62		
Bottle Filled Time	0	2700.00	0.00	4671.13	<0.01
	1	1196.80	49.18		
Raft Built	0	0.00	0.00	181.219	<0.01
	1	1.00	0.00		

Table 72: Game adaptation effectiveness - comparison of the player performance values between the groups with adaptation and without adaptation using real players.

Item	GM	Mean	Std. Dev	F	p
Negative Emotion	0	6.22	1.96	3.25	0.08
	1	7.27	1.72		
Cognitive Load	0	4.30	2.21	23.38	<0.01
	1	7.12	1.39		
Positive Emotion	0	4.92	1.78	18.44	<0.01
	1	7.12	1.52		
Motivation	0	4.15	1.34	31.86	<0.01
	1	6.95	1.76		
Immersion	0	3.50	1.77	22.03	<0.01
	1	5.92	1.47		
Flow	0	5.10	1.82	19.64	<0.01
	1	7.53	1.64		
Arousal	0	4.38	1.53	15.44	<0.01
	1	6.47	1.81		

Table 73: Game adaptation effectiveness - comparison of the game experience questionnaire data between the groups with adaptation and without adaptation using real players.

Adaptation	Group Collaboration				Task solving				Perception of Group				Flow Experience			
	Mean	SD	F	p	Mean	SD	F	p	Mean	SD	F	p	Mean	SD	F	p
Hunting Heron																
0	3.64	0.74	14.88	<0.01	3.23	0.76	10.42	<0.01	3.26	0.70	24.88	<0.01	3.68	0.95	2.53	0.12
1	4.54	0.70			4.08	0.87			4.38	0.67			4.17	0.95		
Carrying Palm																
0	3.64	1.08	11.35	<0.01	3.33	0.98	19.20	<0.01	3.31	0.91	13.01	0.01	3.11	1.15	7.35	0.01
1	4.54	0.53			4.44	0.57			4.29	0.79			3.95	0.76		
Steering Raft																
0	2.65	1.11	7.48	0.01	2.69	1.14	6.05	0.02	2.94	0.13	2.66	0.12	3.19	0.24	2.23	0.15
1	4.11	0.78			4.11	0.80			4.00	1.04			3.89	0.92		
Overall																
0	3.62	0.93	14.95	<0.01	3.23	0.79	13.97	<0.01	3.16	0.83	20.12	<0.01	3.64	0.88	3.93	0.06
1	4.53	0.50			4.10	0.69			4.21	0.63			4.13	0.65		

Table 74: Game adaptation effectiveness - comparison of the teamwork and collaboration questionnaire data between the groups with adaptation and without adaptation using real players.

B.5 GAME MASTERING EFFECTIVENESS

The following Tables 75, 76, and 77 contain the evaluation data of the Game Mastering effectiveness evaluation using real players. Table 75 shows the comparison between the two groups with adaptation and without Game Mastering regarding player performance values. Table 76 shows the comparison between the two groups with adaptation and without Game Mastering regarding perceived game experience. Table 77 shows the comparison between the two groups with adaptation and without Game Mastering regarding perceived teamwork and collaboration.

Response	GM	Mean	STD	F	p
Health	0	47.45	9.05	40.42	<.01
	1	81.39	5.81		
Satiety	0	17.09	10.41	32.62	<.01
	1	44.04	2.90		
Fitness	0	30.14	21.61	2.00	.19
	1	48.53	7.36		
Berries Gathered	0	93.20	35.26	2.71	0.14
	1	137.60	45.06		
Herons Hunted	0	0.40	0.89	7.00	.03
	1	2.20	0.45		
Meat Roasted	0	0.60	1.34	12.89	.01
	1	8.80	4.44		
Hut Build Time	0	1556.20	912.60	6.98	.03
	1	241.40	81.08		
Bottle Found Time	0	2648.20	115.83	79.46	<0.01
	1	897.00	423.74		
Bottle Filled Time	0	2700.00	0.00	40.84	<0.01
	1	1261.60	503.29		
Raft Built	0	0.00	0.00	–	–
	1	1.00	0.00		

Table 75: Game Mastering effectiveness - comparison of the player performance values between the groups with adaptation and without adaptation using real players.

Item	GM	Mean	Std. Dev	F	p
Negative Emotion	0	6.22	1.96	2.77	0.10
	1	7.22	1.84		
Cognitive Load	0	4.30	2.21	0.20	0.66
	1	4.60	2.00		
Positive Emotion	0	4.92	1.78	22.20	<0.01
	1	7.83	2.11		
Motivation	0	4.15	1.34	36.11	<0.01
	1	7.42	2.03		
Immersion	0	3.50	1.77	5.92	0.02
	1	5.27	2.72		
Flow	0	5.10	1.82	9.82	<0.01
	1	7.08	2.17		
Arousal	0	4.38	1.53	11.20	<0.01
	1	6.40	2.22		

Table 76: Game Mastering effectiveness - comparison of the game experience questionnaire data between the groups with adaptation and without adaptation using real players.

GM	Collab.				Self Perc.				Group Per.				Flow			
	Mean	SD	F	p	Mean	SD	F	p	Mean	SD	F	p	Mean	SD	F	p
Hunting Heron																
0	3.64	0.74	13.61	<0.01	3.23	0.76	13.42	<0.01	3.26	0.70	30.50	<0.01	3.68	0.95	1.47	0.23
1	4.43	0.61			4.16	0.85			4.35	0.53			4.04	0.94		
Carrying Palm																
0	3.64	1.08	11.39	<0.01	3.33	0.98	28.47	<0.01	3.31	0.91	32.16	<0.01	3.11	1.15	9.93	<0.01
1	4.60	0.64			4.64	0.45			4.64	0.47			4.12	0.80		
Steering Raft																
0	2.65	1.11	17.22	<0.01	2.69	1.14	10.77	<0.01	2.94	0.13	22.21	<0.01	3.19	0.24	7.57	0.01
1	4.13	0.45			4.21	0.73			4.21	0.53			4.25	0.75		
Overall																
0	3.62	0.93	10.30	<0.01	3.23	0.79	14.70	<0.01	3.16	0.83	22.90	<0.01	3.64	0.88	4.19	0.05
1	4.45	0.69			4.23	0.86			4.32	0.70			4.18	0.76		

Table 77: Game Mastering effectiveness - comparison of the teamwork and collaboration questionnaire data between the groups with adaptation and without adaptation using real players.

C QUESTIONNAIRE DETAILS

»One learns from books and example only that certain things can be done. Actual learning requires that you do those things.«

— Frank Herbert

THIS APPENDIX contains details about the two questionnaires used for the evaluations of the game adaptation effectiveness (Section 9.5) and for the Game Master effectiveness (Section 9.6). The data shown here is the foundation of the qualitative and quantitative analysis and plotted graphs in those sections.

C.1 GAME EXPERIENCE QUESTIONNAIRE

Based on Nacke [117], a user experience questionnaire has been constructed and evaluated ($N = 145, \alpha = .93$) for user experience measurement. Based on the analysis of the usability standards ISO 9241-10/-11, ISO 14915-1 and ISO 13407 and user experience research by Mandryk et al. [99] and Nacke [118], a questionnaire for the evaluation of Serious Games in an interdisciplinary study was elaborated between the Multimedia Communications Lab and the faculty for psychology at the Technische Universität Darmstadt [55].

The following explanation of the game experience questionnaire is based on the questionnaire description in [178]: The game experience questionnaire is an abstraction of several aspects of user experience and usability. A summerization of flow theory aspects can be found in [117]. Seven sub-scales define the user experience score. These are *negative emotion*, *positive emotion*, *cognitive load*, *motivation*, *immersion*, *flow*, and *arousal*. The game design score is defined by ten sub-scales. Those are:

- Quests
- Environment
- Effectance
- Curiosity
- Personalization
- Interface
- Feedback
- Social needs
- Storytelling
- Structure

For each sub-scale, three questions to the sub-scale's topic are contained. A 10-point Likert scale (1 to 10) was used for the question items with '1' meaning 'do not agree at all' and '10' meaning 'fully agree'. A first evaluation of 145 questionnaires showed a Cronbach's Alpha = .93 for the overall user experience score of this questionnaire. Hence, it is assumed that the overall user experience score (the mean of the 21 user experience questions) seems to build one homogeneous factor. The theoretical background for the questionnaire is based on the following definition of immersion: "immersion in the game world derives from the player becoming the game character, in the sense of the player having the experience of acting within the game world" [117] (p. 146).

The questionnaire items are listed in Table 79 . They are marked with the prefixes XXN to show their function regarding the seven scales of game experience as shown in Table 78, where XX indicates the scale and N the number of the item for the scale. The original German wording is provided, as well as an English translation in *italics*.

NE	Negative emotion
CL	Cognitive load
PE	Positive emotion
CL	Motivation
IM	Immersion
FL	Flow
AR	Arousal

Table 78: Game experience questionnaire scales.

	Question (German/English)
NE1	Das Spiel hat Langeweile vermieden *The game prevented boredom*
NE2	Das Spiel hat Frustration vermieden *The game prevented frustration*
NE3	Ich habe mich nicht über das Spiel geärgert *The game did not annoy me*
CL1	Das Spiel hat mich angenehm gefordert *The game was demanding in an enjoyable way*
CL2	Das Spiel hat meine Fantasie angeregt *The game stimulated my fantasy*
CL3	Ich war durch die Aufgaben und Möglichkeiten im Spiel nicht überfordert *I was not overchallenged by the game's tasks and possibilities*
PE1	Das Spiel hat Spass gemacht *The game was fun*
PE2	Das Spiel gab mir das Gefühl, eigenbestimmt und kompetent zu sein *The game let me feel autonomous and competent*
PE3	Ich fand das Spiel ansprechend gestaltet *I perceived the game as appealing*
MO1	Das Spiel war mitunter so einnehmend, dass ich unbedingt wissen wollte, wie es weiter geht *The game was occasionally so engaging that I definitely wanted to know how it would go on*

(cont.)	Question(German/English)
MO2	Einen Entwicklungsprozess festzustellen motivierte mich stark, weiter zu machen *I was motivated strongly to play on because I could recognize that the game developed*
MO3	Teilweise spielte ich nur noch um des Spieles willen *Sometimes, I played only because of the game*
IM1	Das fühlte ich mich wie ein Teil der Spielwelt *Sometimes I felt like a part of the game world*
IM2	Ich hatte während des Spiels das Gefühl, die Spielfigur zu sein *I felt that I was the avatar during the game*
IM3	Das Spiel bot die Möglichkeit, ein eigenständiges Selbstkonzept zu entwickeln, dem es Spass machte zu folgen *It was possible to develop an own playing concept during the game which was fun to follow*
FL1	Das Spiel war so spannend, dass es meine ganze Aufmerksamkeit auf sich og *The game was so thrilling that it attracted my full attention*
FL2	Das Spiel war so interessant, dass ich gar nicht merkte, wie schnell die Zeit vergeht *The game was so interesting, that I forgot about time*
FL3	An manchen Stellen war das Spiel so fesselnd, dass ich vollkommen vim Spiel eingenommen wurde *Sometimes the game was so compelling that I was completely immersed in the game*
AR1	Manchmal war ich im Nachhinein sehr erleichtert, da ich Scheitern befürchtete *Sometimes I was relieved because I was in fear of failing*
AR2	Ich merkte, dass ich teilweise stark emotional beteiligt war (Spannung, Trauer, Erleichterung, Freude, Wut) *I noticed that I was sometimes strongly emotionally involved (tension, grief, relief, joy, anger)*
AR3	Ich fühlte mich durch das Spiel in einen angenehmen Zustand versetzt *The game made me feel pleasant*

Table 79: Game experience questionnaire items.

C.2 INTERACTION AND TEAMWORK QUESTIONNAIRE

In addition to the game experience questionnaire, another questionnaire was designed to measure the players' perception of teamwork and communication within the group. The questionnaire contains four identical parts of questions, testing for the three main collaborative parts of the game (i.e., building the hut, hunting the heron, and steering the raft), as well as one part for the overall perception of teamwork and communication for the complete game. Each part included the same 17 questions about the quality of experience. For the sub-scale *perception of the group cooperation* five questions were asked, and for each of the sub-scales *task solving*, *perception of the group performance* and *flow experience* four questions were asked, totaling in 17 questions for each of the four game parts (building the hut, hunting the heron, steering the raft, overall), summing up to 68 questions. The questions could be answered using a 5-point Likert scale with '1' meaning 'do not agree at all' and '5' meaning 'fully agree'. In addition, players were asked to state their age, their sex, and how many hours per week they play computer/video games.

The questionnaire items are listed in Table 81 . They are marked with the prefixes XXN to show their function regarding the four scales of teamwork and communication as shown in Table 80, where XX indicates the scale and N the number of the item for the scale. The original German wording is provided, as well as an English translation in *italics*.

GC	Perception of the group cooperation
TS	Task solving
GP	Perception of the group performance
FE	Flow experience

Table 80: Teamwork and communication questionnaire scales.

	Question (German/English)
GC1	Die Aufgabenaufteilung in der Gruppe war gut *Group task sharing was good*
GC2	Die Kommunikation in der Gruppe war gut *Group communication was good*
GC3	Die Zusammenarbeit in der Gruppe war gut *Teamwork in the group was good*
GC4	Die Atmosphäre in der Gruppe war gut *The group had a good vibe*
GC5	Die Hilfsbereitschaft in der Gruppe war gut *Helpfulness in the group was good*
TS1	Meine Zufriedenheit mit der Qualität der *eigenen* Aufgabenbearbeitung war gut *I was satisfied with the quality of* my own *handling of the tasks*
TS2	Das Ziel der Aufgabe war *mir* klar verständlich *The task goal was clear to* me

(cont.)	Question(German/English)
TS3	Ich wusste, was ich zum Lösen der Aufgabe zu tun hatte *I knew what I had to do to solve the task*
TS4	Ich habe wenig Zeit bei der Bearbeitung der Aufgabe einfach nur verschwendet *I wasted only little time while solving the task*
GP1	Meine Zufriedenheit mit der Qualität der *Gruppen-* Aufgabenbearbeitung war gut *I was satisfied with the quality of* the group's *handling of the tasks*
GP2	Das Ziel der Aufgabe war *der Gruppe* klar verständlich *The task goal was clear to* the group
GP3	Die Gruppe wusste, was sie zum Lösen der Aufgabe zu tun hatte *The group knew what it had to do to solve the task*
GP4	Die Gruppe habe wenig Zeit bei der Bearbeitung der Aufgabe einfach nur verschwendet *The group wasted only little time while solving the task*
FE1	Die Aufgabe hat mich angenehm gefordert *The tasks challenged me exactly right*
FE2	Ich war bei der Aufgabenbearbeitung nicht frustriert oder verärgert *I was not frustrated or angered while solving the task*
FE3	Die Aufgabe zog meine ganze Aufmerksamkeit auf mich *The task drew my complete attention*
FE4	Ich merkte bei der Aufgabenbearbeitung nicht, wie schnell die Zeit vergeht *While solving the task I did not notice how fast time passed*

Table 81: Teamwork and collaboration questionnaire items.

D AUTHOR'S PUBLICATIONS

D.1 MAIN PUBLICATIONS

1. Viktor Wendel, Marc-André Bär, Robert Hahn, Benedict Jahn, Max Mehltretter, Stefan Göbel, and Ralf Steinmetz. *A Method for Automatic Situation Recognition in Collaborative Multiplayer Serious Games*. *EAI Endorsed Transactions*, 15(4), 2015.

2. Viktor Wendel, Stefan Krepp, Michael Gutjahr, Stefan Göbel, and Ralf Steinmetz. *Game Mastering in Collaborative Serious Games - An Evaluation of Instructor Influence on Learners' Performance*. *International Journal of Game-based Learning*, 5(4), 2015.

3. Viktor Wendel, Johannes Alef, Stefan Göbel, and Ralf Steinmetz. *A Method for Simulating Players in a Collaborative Multiplayer Serious Game*. In: ACM, editor, *SeriousGames '14 Proceedings of the 2014 ACM International Workshop on Serious Games*, pages 15–20. ACM, 2014. ISBN 978-1-4503-3121-0.

4. Viktor Wendel, Marc-André Bär, Robert Hahn, Benedict Jahn, Max Mehltretter, Stefan Göbel, and Ralf Steinmetz. *Automatic Situation Recognition in Collaborative Multiplayer Serious Games*. In: Academic Conferences Limited, editor, *Proceedings of the 7th European Conference on Game-based Learning*, pages 610–619. Academic Conferences Limited, acpi, Berlin, Germany, 2014. ISBN 978-1-910309-55-1.

5. Viktor Wendel, Michael Gutjahr, Stefan Göbel, and Ralf Steinmetz. *Instructor Support in Collaborative Multiplayer Serious Games for Learning - Game Mastering in the Serious Game 'Woodment'*. In: *Proceedings of the 6th International Conference on Computer Supported Education*, volume 3, pages 49–56. SciTePress, 2014. ISBN 978-989-758-022-2.

6. Viktor Wendel, Sebastian Ahlfeld, Stefan Göbel, and Ralf Steinmetz. *Rule-based Automatic Adaptation of Collaborative Multiplayer Serious Games - Supporting Instructors in Game-based Learning Scenarios*. In: *Proceedings of the 5th International Conference on Computer-supported Education*, pages 536–547. SciTePress, Aachen, 2013. ISBN 978-989-8565-53-2.

7. Viktor Wendel, Michael Gutjahr, Philipp Battenberg, Roman Ness, Sebastian Fahnenschreiber, Stefan Göbel, and Ralf Steinmetz. *Designing A Collaborative Serious Game For Team Building Using Minecraft*. In: Paula Escudeiro Carlos Vaz de Carvalho, editor, *Proceedings of the 7th European Conference on Game-based Learning*, pages 569–578. ACPI, Instituto Superior de Engenharia do Porto, Portugal, 2013. ISBN 978-1-909507-63-0.

8. Viktor Wendel, Michael Gutjahr, Stefan Göbel, and Ralf Steinmetz. *Designing Collaborative Multiplayer Serious Games*. *Education and Information Technologies*, 2(18):287–308, 2013. ISSN 1360-2357.

9. Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Game Mastering in Collaborative Multiplayer Serious Games*. In: Stefan Göbel, Wolfgang Müller, Bodo Urban, and Josef Wiemeyer, editors, *E-Learning and Games for Training, Education, Health and Sports - LNCS*, volume 7516, pages 23–34. GameDays 2012, Springer, Darmstadt, Germany, 2012. ISBN 978-3-642-33465-8.

10. Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Collaborative Learning in Multiplayer Serious Games*. In: Winfred Kaminski, editor, *Clash of Realities Proceedings 2012*. Koped Verlag, München, 2012.

11. Viktor Wendel, Michael Gutjahr, Stefan Göbel, and Ralf Steinmetz. *Designing Collaborative Multiplayer Serious Games for Collaborative Learning*. In: Jose Cordeiro Markus Helfert, Maria Joao Martins, editor, *Proceedings of the CSEDU 2012*, volume 2, pages 199–210. INSTINCC, SciTePress 2012, Porto, Portugal, 2012. ISBN 978-989-8565-07-5.

12. Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Seamless Learning In Serious Games - How to Improve Seamless Learning-Content Integration in Serious Games*. In: *Proceedings of the CSEDU 2011*, volume 1, pages 219–224. SciTePress - Science and Technology Publications, 2011. ISBN 978-989-8425-49-2.

13. Viktor Wendel, Annika Kliem, Stefan Göbel, Josef Wiemeyer, and Ralf Steinmetz. *Virtual Sports Teacher: A 3D Serious Game for Physical Education with Game Master Support*. In: AACE, editor, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011*, pages 2830 – 2839. AACE, 2011. ISBN 1-880094-35-X.

14. Viktor Wendel, Maxim Babarinow, Tobias Hörl, Sergej Kolmogorov, Stefan Göbel, and Ralf Steinmetz. *Woodment: Web-Based Collaborative Multiplayer Serious Game*. In: Xiaopeng Zhang, Shaochun Zhong, Zhigeng Pan, Kevin Wong, and Ruwei Yun, editors, *Transactions on Edutainment IV LNCS*, volume 6250, pages 68–78. Springer, 2010. ISBN 978-3-642-14483-7.

15. Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Game Design und Game Development in einer Serious Games Vorlesung*. In: Ulrik Schroeder, editor, *Interaktive Kulturen - Proceedings der Mensch & Computer 2010, DeLFI 2010, Entertainment Interfaces 2010*, pages 283–289. Logos Verlag, Berlin, 2010. ISBN 978-3-8325-2578-1.

16. Viktor Wendel, Sandro Hardy, Stefan Göbel, and Ralf Steinmetz. *Adaption und Personalisierung von Exergames*. In: J. Wiemeyer, D. Link, R. Angert, B. Holler, A. Kliem, N. Roznawski, D. Schöberl, and M. Stroß, editors, *Sportinformatik trifft Sporttechnologie: Abstractband zur Tagung der dvs-Sektion Sportinformatik und der deutschen interdisziplinären Vereinigung für Sporttechnologie*, pages 97–99. Institut für Sportwissenschaft der Technischen Universität Darmstadt, Darmstadt, 2010.

17. Viktor Wendel, Felix Hertin, Stefan Göbel, and Ralf Steinmetz. *Collaborative Learning by Means of Multiplayer Serious Games*. In: Xiangfeng Luo, Marc Spaniol, Lizhe Wang, Qing Li, Wolfgang Neidl, and Wu Zhang, editors, *Proceedings of ICWL 2010*, volume 6483, pages 289 – 298. Springer, 2010. ISBN 978-3-642-17406-3.

D.2 CO-AUTHORED PUBLICATIONS

18. Stefan Göbel, Florian Mehm, Viktor Wendel, Johannes Konert, Sandro Hardy, Christian Reuter, Michael Gutjahr, and Tim Dutz. *Erstellung, Steuerung und Evaluation von Serious Games*. *Informatik Spektrum*, 2014. ISSN 0170-6012, DOI 10.1007/s00287-014-0824-2.

19. Christian Reuter, Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Game Design Patterns for Collaborative Player Interactions*. In: Staffan Björk and Annika Waern, editors, *the of Game - Proceedings of DiGRA 2014*, page 16. 2014.

20. Johannes Konert, Viktor Wendel, Kristina Richter, and Stefan Göbel. *Serious Games and Virtual Worlds in Education, Professional Development, and Healthcare*, chapter 6, pages 85–104. IGI Global, Hershey, USA, 1 edition, 2013. ISBN 9781466636736.

21. Stefan Göbel, Florian Mehm, and Viktor Wendel. *Adaptive Digital Storytelling for Digital Educational Games*, chapter 5, pages 89–104. An Alien's Guide to Multi-Adaptive Educational Computer Games. Informing Science Press, Santa Rosa, USA, 2012. ISBN 1-932886-56-7.

22. Christian Reuter, Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Multiplayer Adventures for Collaborative Learning With Serious Games*. In: Patrick Felicia, editor, *Proceedings of the 6th European Conference on Games Based Learning*, pages 416–423. Academic Conferences Limited, 2012. ISBN 978-1-908272-69-0.

23. Christian Reuter, Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Towards Puzzle Templates for Multiplayer Adventures*. In: Stefan Göbel, Wolfgang Müller, Bodo Urban, and Josef Wiemeyer, editors, *E-Learning and Games for Training, Education, Health and Sports*, pages 161–163. Springer, 2012. ISBN 978-3-642-33465-8.

24. Annika Kliem, Viktor Wendel, Christian Winter, Josef Wiemeyer, and Stefan Göbel. *Virtual Sports Teacher - A Serious Game in Higher Education*. In: *Serious Games - Theory, Technology & Practice*, pages 61–72. TU Darmstadt, Darmstadt, 2011. ISBN 978-3-928876-27-8.

25. Johannes Konert, Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Towards an Analysis of Cooperative Learning-Behaviour in Social Dilemma Games*. In: *Proceedings of the 5th European Conference on Games Based Learning (ECGBL 2011)*, pages 329–332. Academic Conferences International, Academic Publishing International, Berks, UK, The National and Kapodistrian University, Athens, Greece, 2011. ISBN 978-1-908272-19-5.

26. Stefan Göbel, Sandro Hardy, Viktor Wendel, Florian Mehm, and Ralf Steinmetz. *Serious Games for Health - Personalized Exergames*. In: *Proceedings ACM Multimedia 2010*, pages 1663–1666. ACM Multimedia, 2010. ISBN 978-1-60558-933-6.

27. Stefan Göbel, Viktor Wendel, Christopher Ritter, and Ralf Steinmetz. *Personalized, Adaptive Digital Educational Games using Narrative Game-based Learning Objects*. In: *Lecture Notes in Computer Science*, volume 6249, pages 438–445. Springer, 2010. ISBN 978-3-642-14532-2.

28. Florian Mehm, Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Bat Cave: A Testing and Evaluation Platform for Digital Educational Games*. In: Bente Meyer, editor, *Proceedings of the 3rd European Conference on Games Based Learning*, pages 251–260. Academic Conferences International, Reading, England, 2010. ISBN 978-1-906638-79-5.

29. Florian Mehm, Viktor Wendel, Sabrina Radke, Stefan Göbel, Sebastian Grünwald, Robert Konrad, and Ralf Steinmetz. *Re-Authoring eines Lernadventures*. In: Holger Diener, Steffen Malo, Bodo Urban, Dennis Maciuszek, and Alke Martens, editors, *Spielend Lernen*, pages 27–42. Fraunhofer Verlag, Stuttgart, 2010. ISBN 978-3-8396-0186-0.

30. Rastin Pries, Viktor Wendel, Barbara Staehle, and Dirk Staehle. *Genetic Algorithms for WMN Planning*. In: *13-th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2010)*, pages 226–234. ACM New York, NY, USA, Bodrum, Turkey, 2010. ISBN 978-1-4503-0274-6.

E SUPERVISED STUDENT THESES

E.1 DIPLOMA AND MASTER THESES

KOM-D-0434 Felix Hertin. *Design and Development of a Mobile Multiplayer Serious Game with Game-Master Support based on Google Android Technology*. Diploma Thesis. Technische Universität Darmstadt, Germany, August 2011.

KOM-D-0435 Sabine Gund. *Design and Development of a Browser-based Serious Game for Comparative Evaluation of the 'Seamless Learning' Concept in Serious Games*. Diploma Thesis. Technische Universität Darmstadt, Germany, October 2011.

KOM-D-0450 Christopher Rodenberger. *Conception and Implementation of Game Mastering and Team Leader Components for a Collaborative 3D Multiplayer Serious Game*. Diploma Thesis. Technische Universität Darmstadt, Germany, March 2012.

KOM-D-04531 Josef Bräuer. *Development of Methods and Concepts for the Application of Game Mastering Approaches in a Game-based Training Scenario of a Non-game Software*. Diploma Thesis. Technische Universität Darmstadt, Germany, July 2013.

E.2 MASTER THESES

KOM-D-0440 Christian Reuter. *Development and Implementation of Methods and Concepts for Multiplayer Adventures*. Master Thesis. Technische Universität Darmstadt, Germany, October 2011.

CO-SUPERVISED Christian Riether. *Virtual Actors: An Approach for Autonomous, Synthetic Characters*. Master Thesis. Technische Universität Darmstadt, Germany, April 2012.

KOM-D-0454 Sebastian Ahlfeld. *Development and Implementation of Methods and Concepts for a Player and Learner Centered Adaptation Engine for Collaborative Multiplayer Serious Games*. Master Thesis. Technische Universität Darmstadt, Germany, July 2012.

KOM-M-0480 Jeanette Forster. *Development and implementation of automatized adaptation mechanisms for collaborative multiplayer Serious Games*. Master Thesis. Technische Universität Darmstadt, Germany, October 2013.

E.3 BACHELOR THESES

KOM-S-0386 Stefan Krepp. *Development and Implementation of Game-Mastering Methods and Concepts for a Collaborative Multiplayer Serious Game*. Bachelor Thesis. Technische Universität Darmstadt, Germany, April 2011.

KOM-S-0411 Philipp Dürr. *Development and Implementation of an Incentive System for a Mobile 3D Serious Game*. Bachelor Thesis. Technische Universität Darmstadt, Germany, Februar 2013.

KOM-B-0469 Johannes Alef. *Design of Methods and Concepts for a Simulation Model of Virtual Players and Learners in Multiplayer Serious Games*. Bachelor Thesis. Technische Universität Darmstadt, Germany, September 2011.

KOM-B-0470 Markus Grau. *Design of Methods and Concepts for Configuration and Runtime Controlling of Collaborative Mutliplayer Serious Games Using Minecraft*. Bachelor Thesis. Technische Universität Darmstadt, Germany, january 2014.

E.4 STUDENT RESEARCH PROJECT (STUDIENARBEIT)

KOM-B-0470 Christopher Rodenberger. *Development and Implementation of Game-Mastering Methods and Concepts for a Collaborative Multiplayer Serious Game*. Studienarbeit. Technische Universität Darmstadt, Germany, July 2011.

F

CURRICULUM VITÆ

PERSONAL INFORMATION

Name	Viktor Wendel
Date of Birth	July 22, 1983
Place of Birth	Bad Kissingen
Nationality	German

EDUCATION

11/2009–12/2014	Technische Universität Darmstadt Department of Electrical Engineering and Information Technology Associate researcher at Multimedia Communications Lab (KOM)
10/2003–09/2009	Julius-Maximilian-Universität Würzburg Department of Computer Science and Mathematics Diploma thesis at Lehrstuhl für Informatik III, Kommunikationsnetze Diplom-Informatiker
06/2003–08/2003	Heinz-Kalk-Krankenhaus, Bad Kissingen Internship
08/2002–05/2003	Heinz-Kalk-Krankenhaus, Bad Kissingen Civilian Service
06/2002	Jack-Steinberger Gymnasium, Bad Kissingen Abitur, majoring in Mathematics and English

AWARDS AND HONORS

09/2012	Best Paper Award at the GameDays 2012 for: Viktor Wendel, Stefan Göbel, and Ralf Steinmetz. *Game Mastering in Collaborative Multiplayer Serious Games*. In: Bodo Urban, Josef Wiemeyer, Stefan Göbel, Wolfgang Müller, editor, E-Learning and Games for Training, Education, Health and Sports - LNCS, volume 7516, pages 23-34. GameDays 2012, Springer, Darmstadt, Germany, 2012. ISBN 978-3-642-33465-8.
05/2014	Bavarian Economy Special Award (Sonderpreis der bayrischen Wirtschaft) as part of the German Computer Game Award

TEACHING ACTIVITIES

2010–2014	Lab Exercise Game Technology, Several Topics Assigned to 18 Student Groups
2010–2014	Lecture Communication Networks I, Assistant Lecturer, Exercise coordination, and Exam Preparation
2010–today	Seminar Serious Games, Several Topics Assigned to 31 Students
2010–today	Lecture Serious Games, Assistant Lecturer, Exercise coordination, and Exam Preparation
2010–2014	Supervisor of 12 Bachelor's and Master's theses (including "Diplomarbeiten")

SCIENTIFIC ACTIVITIES

Organization	GameDays, Darmstadt, Germany (2010–2015)
Publication Chair	GameDays, Darmstadt, Germany (2014–2015)
Program Chair	European Conference on Game-based Learning (ECGBL) (2012–2015)
Workshop Chair	Workshop 'Serious Games in der Lehre' at GameDays 2013 and 2014, Darmstadt, Germany
Editor	ACM SIGMM Records (2010 – today)
Reviewing Board	GameDays (2010–today)
Reviewing Board	International Journal of Game-Based Learning (IJGBL) (2014–today)
Reviewing Board	European Conference on Game-based Learning (ECGBL) (2012–today)

G ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe.

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 10. Februar 2015

colophon

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Final Version as of July 21, 2015 (`classicthesis`).