Designing adaptivity in educational games to improve learning

Singh, Javin

Awarding institution:
King's College London

Designing adaptivity in educational games to improve learning

Javin Singh

Supervisors:
Dr. Simon Miles
Prof. Michael Kolling
Dr. Petr Slovak

Department of Informatics
Kings College London
United Kingdom

December 2021

A thesis presented for the degree of
Doctor of Philosophy

Acknowledgements

I would like to acknowledge my supervisors particularly Dr. Simon Miles whose patience and guidance was crucial to the success of this study.

Dedication

I would like to dedicate this thesis to my loving family who have stood by me from the very beginning.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done. This dissertation contains fewer than 100,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Abstract

The study of pedagogy has shown that students have different ways of learning and processing information. Students in a classroom learn best when being taught by a teacher who is able to adapt and/or change the pedagogical model being used, to better suit said students and/or the subject being taught. When considering other teaching mediums such as computer-assisted learning systems or educational video games, research also identified the benefits of adapting educational features to better teach players. However, effective methods for adaptation in educational video games are less well researched.

This study addresses four points regarding adaptivity within educational games. Firstly, a framework for making any game adaptive was extracted from the literature. Secondly, an algorithm capable of monitoring, modelling and executing adaptations was developed and explained using the framework. Thirdly, the algorithm's effect on learning gains in players was evaluated using a customised version of Minecraft as the educational game and topics from critical thinking as the educational content. Lastly, a methodology explaining the process of utilising the algorithm with any educational game and the evaluation of said methodology were detailed.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AC	Adaptation command
ACT	Analytical and critical thinking
AES	Adaptable elements set
AI	Artificial intelligence
ASD	Adaptation statistics dataset
BI	Base interactions needed to complete a scenario
BN	Bayesian Network
BT	Base time needed to complete a scenario
$CCTST$	California Critical Thinking Skills Test
$CCTT$	Cornell Critical Thinking Tests
CI	Interactions taken to complete a learning scenario
CIM	Completion interaction multiplier
CIT	Completion interaction threshold
CM	Completion multiplier
CT	Completion threshold
CTM	Completion time multiplier
CTT	Completion time threshold
DAG	Directed acyclic graph
DDA	Dynamic difficulty adjustment
FLS	Fuzzy logic system
GES	Game element set
HMM	Hidden markov model
MDM	Massive-scale Game Data Mining

NPC Non playable character

PBD Player behaviour dataset

PEM Player experience modelling

PM Player model

$RAES$ Recommended adaptable elements set

RSD Running state dataset

SBD Scenario benchmark dataset

SPS Simulated player script

TI Time taken to complete a learning scenario

$WGCTA$ Watson-Glaser Critical Thinking Appraisal

Chapter 1

Introduction

The United Nations (2015) explain that education is the key to escaping social and economic poverty. The United Nations elaborate that a lack of analytical and critical thinking (ACT) based education hinders both students and the workforce. This is due to ACT being a transferable skill especially when applied to problem solving and making sound conclusions (Popkewitz, 1999). ACT is therefore a desirable skill to teach to all age groups.

Braghirolli et al. (2016) explain that due to their ability to entertain, *educational games* provide players with greater motivation to learn compared to their textbook counter parts. Therefore, educational games could be employed as a teaching medium due to their ability to educate as well as entertain. Educational games convey their educational content by setting players a series of objectives that can only be accomplished when the player has understood the content. Zorn et al. (2013) showcase this using the video game Minecraft as a platform to teach visual based programming. Using Minecraft's versatility, they created customised educational functionality and combined them with the present entertainment factors to create learning objectives. This proved successful in producing significant increases in coding ability and interest towards coding amongst the players. Part of this success was credited with familiarity and entertaining aspects of Minecraft providing players with the motivation to learn.

However, research done by Pashler et al. (2008) has shown that individuals learn using different pedagogical models and at different rates, depending on the content. Teachers undergo training to recognise and adapt to these differences in learning. Unlike teachers, current educational games lack the ability to adapt to their target learners. Thus, educational games could be made more effective by incorporating functionality that could allow them to adapt to their players, akin to teachers.

When examining entertainment video games, Changchun et al. (2009) explain that certain video games like Resident Evil 4 attain critical success and popularity by employing systems that subtly adapt to the player, by altering the difficulty of certain aspects unbeknownst to the player. The player's lack of awareness of this adaptation allows for an appropriate level of challenge to be maintained and increase the game's entertainment capability. This is due to the chance that players, especially those who are struggling, might feel patronised and/or disheartened if made aware of the adaptation and may cease playing. This may suggest that adaptation must occur without the knowledge of the player to avoid making the player feel discouraged, so that he/she completes the game and achieves the maximum educational benefit.

While adaptivity has been applied to the challenge/difficulty aspects of video games, Soflano, Connolly, and Hainey (2015) detail that adaptivity in a video game's educational mechanisms are less explored. It is hypothesised that an educational video game possessing adaptivity geared towards education would be able to better educate individuals, when compared to an educational game without adaptivity.

1.1 Research question

With the premise and motivation established, this study asks the following research question:

How can educational games in general be made adaptive to as many individual players as possible so as to best improve their learning gains?

Answering this question entails exploring pedagogy and how it relates to educational games, the parts of educational games that can be made adaptive, the principles behind when adaptation would be beneficial, the design of an algorithm to execute adaptations, a methodology for utilising the algorithm in any educational game and the means to evaluate the effects of the former points on the learning gains of a player.

Furthermore, as the research question pertains towards adaptivity in educational games being accessible to as many players as possible, several motivating requirements for the algorithm were identified. These are detailed below:

Motivating Requirement 1 (No dependency on specialised equipment) *Requiring additional equipment for the algorithm to function directly impacts the number of systems it could potentially run on based on the availability of such equipment. Therefore, it is beneficial to accessibility for the algorithm to be able to function without the need of additional equipment.*

Motivating Requirement 2 (Player convenience) *In the interest of accessibility, functionality of the algorithm should refrain from practices that might cause inconvenience to and potentially put off players from playing the educational game. Possible inconveniences could include a dependency on interviews which might be difficult to schedule and being monitored by a researcher while playing the educational game. Avoiding inconveniences like these may boost players' willingness to play the educational game, thus benefiting from adaptation.*

Motivating Requirement 3 (Immediate adaptation) *Adaptation should not depend on the player playing a game for extensive periods or multiple times, as this may not be realistic given the setting in which they are learning. Thus, it is a priority for adaptivity to be functional as soon as possible while playing an educational game.*

1.2 Contributions

The study's aim of designing an approach to incorporating adaptivity into educational games to increase their educational effectiveness will be realised through the following four contributions:

1. The extraction of a framework, from academic literature, to make games in general adaptive. The adaptive framework was split into the four issues of what to adapt, when to adapt, how to adapt, and how to collect data for the first three issues.

2. The design of an adaptive algorithm to identify the adaptations that could be beneficial to learning, and execute them. This process was split into the algorithm design specification and the algorithm

implementation. The algorithm design specification explored and justified solutions and/or techniques to address the four issues posed by the framework. This included identifying and modelling components found in games whose adaptation could positively impact learning (what to adapt), establishing in-game conditions where adaptivity could benefit learning (when to adapt), exploring techniques to generate an internal model of the player's proficiency at the game, used to reason over and allow for educational adaptation, called the *player model* (how to adapt), and exploring techniques to monitor players to collect data for addressing the previous three issues (data collection). The algorithm implementation drew from the design specification to detail the implementation of the adaptive algorithm. This included a high level description of design choices and functionality, input and output data structures, and pseudo-code illustrating how the algorithm should be coded.

3. The system testing and user evaluation of the algorithm. The system testing aimed to confirm that the algorithm functioned as intended and to measure the correctness of its output. This would be accomplished with the use of simulated players that followed a generated script while the algorithm monitored and adapted to them. The results and conclusions served to inform the design of the user evaluation. The purpose of the user evaluation was to measure the effects on the learning gains of users caused by the algorithm, whilst they played for an educational game. The evaluation was conducted by modifying the entertainment game Minecraft to allow for the creation of educational scenarios that aimed to teach the identification and credibility assessment of information. A comparison between the learning gains of users who played the game with and without the adaptable content being used was be conducted to determine the algorithm's effectiveness.

4. The methodology for utilising the adaptive algorithm with any educational game, and its evaluation. The utilisation methodology was sub divided into ensuring that the educational game chosen/made to be paired with the algorithm is compatible, designing in-game scenarios to teach educational content, ensuring that the algorithm is triggered at the correct instances, coding the algorithm itself, and developing adaptive content for the algorithm to use. The purpose of the methodology's evaluation was to determine its effectiveness at instructing other game designers to incorporate the algorithm with other games. This was conducted by reaching out to game designers via email, providing them with the utilisation methodology and the supplementary background material required to understand it, and having them take part in a 10 question interview or an open ended questionnaire concerning the effectiveness of the methodology.

1.3 Thesis structure

With the motivation, research question and contributions explained, the following is a brief overview of the remaining chapters of the thesis and the content explored in each.

Chapter 2. Literature review - examines the landscape of pedagogy, pedagogy in educational games, educational game design, adaptivity in both entertainment and educational games, and presents an adaptive framework to make games in general adaptive by drawing from the previous topics. The adaptive framework is split into the four issues of what to adapt, when to adapt, how to adapt, and how to collect data for the first three issues. Possible solutions and/or techniques for addressing all four of these issues are also explored.

Chapter 3. Algorithm design specification - explains and justifies the choices of solutions and/or techniques selected to address the four issues of the adaptive framework. First, was identifying and modelling game elements (components found in games) whose adaptation could positively impact learning (what to adapt). Second, was establishing in-game conditions where adaptivity could benefit learning (when to adapt). Third, was exploring and selecting a technique to generate an internal model of the player's proficiency at the game, used to reason over and allow for educational adaptation, called the *player model* (how to adapt). Fourth, was exploring and selecting a technique to monitor players to collect data for addressing the previous three issues (data collection).

Chapter 4. Algorithm implementation - uses the algorithm design specification to detail the implementation of an adaptive algorithm, meant to increase the educational effectiveness of educational games. This includes a high level description of design choices and functionality, input and output data structures, and pseudo-code illustrating how the algorithm should be coded.

Chapter 5. System testing - details the process of system testing the functionality and accuracy of the algorithm using simulated players. Simulated players followed a generated script to mimic the behaviour of actual players while the algorithm monitored them and made adaptations. The output player models were compared to generated reference player models as a measure for accuracy.

Chapter 6. User evaluation - tests the extent of the algorithm's effect on the learning gains of users while playing an educational game. Additionally, details regarding the educational game developed for testing, adaptive content for the algorithm, the educational content chosen, and the method for running the evaluation are explained and justified.

Chapter 7. Methodology for Utilisation - explains and justifies the utilisation methodology created for the purpose of instructing other game designers to incorporate the algorithm with other educational games, so as to make them adaptive. Also details an evaluation aimed at determining the effectiveness of the utilisation methodology at fulfilling its purpose, by drawing from the opinions and reflections of other game designers.

Chapter 8. Conclusions - presents a review of all work done, along with evaluation of the results achieved against the intended contributions and research question. Furthermore, it details possible future work, as well as drawing a formal conclusion.

Chapter 9. Appendices - contains all the additional work and resources used in this study. This includes the full specifications for the educational game content described in Chapter 6, all documentation used for the user evaluation described in chapter 6 and all documentation used for the methodology evaluation described in Chapter 7.

Chapter 2

Literature review

With the motivation and research contributions of this study established, this chapter delves into the relevant background research. This includes four topics. First is a review on pedagogy applied to educational games. Second is an overview on the design of educational games and how pedagogy applies to them. Third is an overview of the ways that entertainment and educational games have been made adaptive. Fourth extracts a framework for adpativity from the explored games. This is subdivided to the identification of possible adaptable components of games, the purpose and importance of player models for adaptivity in video games, techniques used to generate player models, and techniques for monitoring a player's behaviour to generate player models. These topics were chosen to address the challenges of determining the best methods and design choices for an algorithm to make educational games adaptive. Concepts and ideas explored in this chapter directly contributed to the design of the algorithm and educational game specified in later chapters.

2.1 Pedagogy

As the aim of this study is directed towards learning and teaching within the context of educational games, the field of pedagogy was deemed directly relevant, and thus researched. Li (2012) defines pedagogy as the study of the teaching and learning of knowledge and/or skills, and the interactions that take place during the learning. He goes on to explain that the theory and practice of pedagogy vary greatly due to the differences in social, political and cultural beliefs. Although many pedagogical models exist, researchers such as Boghossian (2006) and Amineh and Asl (2015) note that five pedagogical models are most often used in the context of education. These include behaviorism, cognivitism, contructivism, social contructivism and Socratic pedagogy. All five models are explored below.

2.1.1 Behaviorism

According to Freiberg (1999), behaviorism focuses on gaining knowledge externally through direct instruction and/or the observation of external stimuli and any effects/responses that follow. Thus, any knowledge gained using this method is directly observable, derivable and/or passed from one person to any other. Skinner (1983) explains that the aim of applying the behaviorist model is to make learners undergo some form of conditioning. The purpose of this conditioning is to produce the desired behavioural result in learners. In the context of a traditional classroom, the behaviorist approach involves the teacher conditioning his/her students (learners) to produce the correct answers to either verbal or written questions using information directly imparted by the teacher and/or text book(s). The teacher, using the behaviorist model, would interpret students' correct answers as a sign of successful conditioning, and then continue to reinforce the behavioral conditioning by assigning good grades and vice versa. This is known as lecture-based pedagogy and is the most common form of conditioning used to achieve the desired behaviour of students in a classroom.

When using a behaviorist approach, Amsel (1989) notes that the extent to which students in a classroom are engaged with the educational process goes as far as he/she exhibiting the expected behaviour (e.g. answering a teacher's question when prompted or reproducing the correct answer in a written test). Thus no responsibility is placed on students in determining the information to study or its interpretation and application. This is instead dictated by the teacher's instructions.

Boghossian (2006) further explains that the conditioning of behaviour is more difficult to impart and measure in a classroom than in other contexts. For example, in a hypothetical martial arts class, if students do not form into a straight line, they are punched in the shoulder. After a few people get out of line, and get punched, the line becomes noticeably straighter. Hence, in this example, students are being conditioned to form a straight line (desired behaviour) by being punched in the shoulder (external stimuli) when not exhibiting said behaviour. It is noted that observable physical behaviours that result from physical stimuli are more noticeable and easier to condition than non physical stimuli and behaviours.

2.1.2 Cognitivism

Huo (2019) explains that cognitivism focuses on learners' processing and applying new knowledge to fit in with existing knowledge. Cognivitists view learning as an active process where learners attempt to acquire new knowledge and restructure their understanding of the topic so that their existing knowledge accommodates the new knowledge. Derry (1996) describes knowledge acquisition as a mental activity involving internal coding and structuring by the learner. He also suggests that learning happens best under conditions that are aligned with human cognitive architecture, which refers to the processing and storing of knowledge. Thus,, the emphasis in cognivitist learning is placed on what learners already know, what new knowledge they discover and how they come to acquire it.

Ertmer and Newby (1993) expand on this by stating that the cognivitist approach, when teaching in a classroom, focuses on making knowledge meaningful by helping learners organise and relate new knowledge to prior knowledge in memory. They clarify that teachers should accomplish this via a reciprocal instructional approach that is based on a learner's existing knowledge, as opposed to the more rigid and inflexible structure of behaviorist instruction. This could provide individual learners with a more customised learning experience but may be problematic when attempting to teach groups of learners, each of whose pre-existing knowledge could vary.

Alternatively, Yilmaz (2011) suggests that problem based learning could serve as another method of cognivitist learning. Problem based learning involves presenting learners with an open-ended problem that has clear and correct solution(s), and tasking them with finding said solution(s). As opposed to traditional behaviorist instruction, that teaches knowledge and/or skills first and then introduces the problem, this method introduces the problem at the beginning of the instruction. The instruction should also take into account the learners' existing knowledge and teach the missing necessary knowledge and/or skills. Instead of a structured set of resources, this approach should provide learners with access to substantial resources for research to find the solution(s), with the teacher acting as a guide throughout the process.

2.1.3 Contructivism

Sener (1997) explains that contructivism differs from behaviorism by including the belief that learners are responsible for constructing their own knowledge. The construction of knowledge pertains to learners playing an active role in their own learning, by seeking to find the meaning in their observations and/or experiences. The subjective meaning each learner draws from this, becomes knowledge. The contructivist approach postulates that the subjective meaning that each learner draws from observations and/or experience is just as valid as anyone else's. Due to this, Poerksen (2004) explains that there exists no objective criteria for what constitutes knowledge as different learners may find different meaning in the same observation, and thus construct different knowledge. Therefore, a contructivist approach to learning is based on the learning that occurs through a learner's active involvement in the construction of his/her own meaning and knowledge.

When examining contructivism through the lens of a traditional classroom setting, researchers like Cook (1992) and Hare, Howard, and Pope (2005) advocate the use of negotiation of knowledge between the teacher and learners. This involves learners negotiating their goals, asking questions, and trying to find the answers themselves. By doing this, they argue that knowledge acquisition will be more meaningful due to a sense of ownership in learners for their work and own learning. Hare, Howard, and Pope (ibid.) elaborate that teachers who apply this method, make use of a flexible curriculum centred around the experience of their students.

Regarding the flexible curriculum, Hoover (1996) and Clements (1997) identify four considerations that teachers should make when applying a contructivist teaching approach. First, teaching should not be done as the transmission of knowledge from unknown to known. Instead, constructivist teachers should guide students and provide them with opportunities to test the adequacy of their current understandings. Second, constructivist teachers should consider the prior knowledge of their learners and provide learning situations that exploit inconsistencies between learners' current knowledge and their new experiences. Third, since learners' involvement is mandatory in constructivism, learners' participation in the active search for knowledge should be facilitated and encouraged by teachers. Fourth, time is needed to build the new knowledge actively. During this time, learners reflect on their new experiences and try to consider the relationship between these experiences and the previous ones, in order to improve the accuracy of their knowledge. Teachers should ensure that their learners are afforded this time.

2.1.4 Social Contructivism

Amineh and Asl (2015) state that social constructivists believe that knowledge is first constructed in a social context and is then understood and applied by individuals. Thus, social contructivism (also referred to as collaborative pedagogy) in the context of learning would involve individuals collaborating and constructing knowledge together. A single learner by his/herself would therefore be unable to make use of this method. Similar to contructivism, social contructivism is predicated on an active approach to learning by learners. However, Brown, Collins, and Duguid (1989) explain that the two differ due to social contructivism including loose criteria for what constitutes valid knowledge, which arrises when learners discuss their meaning and knowledge as a group. This allows for multiple learners to observe the same stimuli, form differing meanings and knowledge and evaluate each piece of knowledge for validity, as a group.

When applied in a traditional classroom environment, Shunk (2000) explains that social contructivism is executed as a hybrid of behaviorism and contructivism. It combines the instructional approach and valid knowledge criteria from behaviorism with the active learner driven approach to learning from contructivism.

However, the instructional approach used by teachers involves reciprocal collaborations with the learners, and is therefore milder when compared to the behaviorist approach. Furthermore, the formation of valid knowledge criteria is a collaborative effort between the teacher and learners, as opposed to the purely teacher driven behaviorist approaches. This could take the form of a short lecture from a teacher explaining core information followed by a reciprocal discussion where the learners are set a goal as a class. From there, the learners would then seek and/or derive knowledge individually. This would lead to a class discussion of the individual meanings and knowledge derived, where the learners filter the valid knowledge from invalid knowledge with the assistance of the teacher if necessary.

Wertsch (1997) explains that the advantage of social contructivist teaching is the combination of the uniqueness and individuality of learners, some form of structure and criteria of validity, and the active role learners' play in their own learning. Thus, it encourages unique meaning and knowledge from each learner, who may have wildly differing views due to individual differences and/or differing backgrounds, to be compared and contrasted with each other in order for the learners to attain a more accurate version of their own knowledge. He also adds that learners may improve their own thinking and reasoning abilities by interacting with other learners, especially those who posses a greater competence.

2.1.5 Socratic pedagogy

The philosophy of Socratic pedagogy (also refered to as inquiry based pedagogy) is that there exist an objective truth in all matters and that the truth can be known through reflection and discourse, or, more specifically, through the elenetic process (Boghossian, 2002). The elenetic process is defined as a systematic question and answer process aimed at differentiating objective truth from the other subjective knowledge. By applying this method, a learner is neither told nor constructs knowledge, instead he/she accepts that the objective truth exists amongst other knowledge, and that he/she must discover it by applying the elenetic process. The elenetic process may be applied individually (reflective) or in a group (collaborative).

As explained by Boghossian (2005), Socratic pedagogy applied in the context of a classroom could take the form of a question and answer session between the learners and the teacher to uncover and/or filter objective truth from subjective knowledge. The teacher would also set the topic or aim of the discussion, as well as serve as the mediator. While sharing common ground with the previous pedagogical methods, Socratic pedagogy exhibits distinct differences. It differs from the behaviorist approach due to the learners being encouraged to ask and answer questions in the search for objective truth as opposed to listening a teacher's lecture. Furtheremore, there is no replication or regurgitation of information or facts, nor is there any behavioural expectation.

It differs from the contructivist approach as it believes that there is objective truth to all matters that takes precedence over the subjective views of individual learners. Constructivist learning involves helping learners discover their own knowledge and/or truths. This is incompatible with Socratic pedagogy which puts forward that objective truth and a learner's subjective truth, are distinct and do not hold the same weight. It also differs from social contructivism due to its focus being towards obtaining an objective truth using a structured elenetic process either individually or in group, instead of allowing a group to debate and set it's own criteria for validity. To this end, Boghossian (2002) believes that applying Socratic pedagogy in the classroom will help both the learners and the teacher find the truth.

With the five major pedagogical models used for education detailed, focus now shifts to educational game design as a whole and how it melds with pedagogy.

2.2 Educational Games

Educational games should both educate and entertain their players, as explained by Aggarwal (2013). While it is generally accepted that computer games were primarily designed for entertainment, their capability and potential to educate very quickly became obvious to researchers like Ahn and Dabbish (2008). Thus, the distinction between educational games and purely entertainment games lie in their primary goal. The primary goal of an educational game is to educate its players as opposed to entertaining them. Michael and R. Chen (2006) found that educational games succeed at teaching due to the blending of educational and entertainment design. This implies that players' entertainment should be the secondary goal of educational games that wish to maximise their effectiveness at teaching.

Researchers such as Connolly et al. (2012) and Sedano et al. (2013) noted a distinct increase in motivation and active participation when interacting with such games, when compared to a standard classroom. Connolly et al. (2012) go on to explain the effectiveness of educational games, especially when aimed at contextually circular/skilled based tasks, where memorisation was ineffective. An example includes recovering from the struggle of substance abuse (Verduin et al., 2013).

Rapeepisarn et al. (2008) explain that this increase in motivation is due to educational games blending educational and entertainment design, causing players to focus on the entertainment aspects instead of the educational aspects. This allows players to learn and have fun at the same time. Hence both educational and entertainment design principles were further researched.

2.2.1 Educational design

There exist many educational games that employ varying methods to teach their educational content. Riemer and Schrader (2015) suggest that the three major methods of teaching that most educational games make use of comprise:

Quizzes. These present the educational content directly to players akin to that of an e-textbook, and then administer a series of questions to test the extent of comprehension in players.

Simulations. Simulations aim to recreate situations from the real world with an emphasis on authenticity. This allows players to experience and/or handle events that are too dangerous, expensive or impractical in reality, such as business management, dealing with public health issues and city scale traffic control. Simulations teach players through a combination of directly presenting educational content and experimentation within the simulation. Players are required to understand the content, recognise patterns in the simulation through their experimentation and apply their findings to achieve the desired outcome.

Adventures. Adventures contain scenarios that take place in a virtual world, consisting of fantasy-based stories and/or goal(s). Players are directly involved in the stories and/or goal(s), and progress through the scenarios to acquire and comprehend educational content, so that they may apply it to accomplish or conclude each scenario's goal or story. Adventure style educational games were found to cause players to put greater focus on scenarios' stories and/or goals, (Rapeepisarn et al., 2008). Building on this, Lepper and Cordova (1992) explain that the educational content in adventure style educational games are more indirectly presented to the player as information needed to complete scenarios.

Riemer and Schrader (2015) also conducted a study that compared player's opinions towards each of the three types of teaching methods and found that players learned the best and had the most enjoyable experience with the adventure type. Due to the primary goal of educational games, focus now shifts to how the five pedagogical models of education described above fit into and/or mirror aspects of the three major teaching methods used by educational games.

2.2.1.1 Pedagogy applied to educational games

While educational games are more successful when they both educate and entertain, their primary goal is to educate their players. Hence, the pedagogical techniques that educational games could use were looked into. Veermans (2019) notes that educational games deploy a hybrid pedagogical model combing aspects from the five models above. The following explains the aspects of each pedagogical model that educational games could use and their weaknesses:

Behaviorism. The instructional approach and learner conditioning are adopted. This is because educational games require some form of instruction to play them the correct way (the way their developers intend). This is conditioned through the desired outcomes or goals being achieved when the player exhibits the "correct" behaviour(s), (Ahlbrand, 2017). However, only applying behaviorist techniques to educational games could cause player's to succeed by behaving in the expected way without understanding any of the educational content.

Cognivitism. When applied to educational games, Zarraonandia (2015) states that simulated problem based scenarios are adopted from cognivitism. These facilitate players' cognitive processes by providing a structured knowledge based for players to research and try solution(s). Furthermore, the open ended and learner driven nature of problem based scenarios are in harmony with the active media aspect of educational games. Active media is defined as media that requires player/viewer interaction, such as video games, as opposed to passive media which does not, such as movies or television programs. Cognivitism applied to educational games is however more limited in scope than in a class room setting due to needing specific and finite knowledge bases for research. This is due to the lack of a teacher who would normally provide assistance and mediate learners if they go off topic.

Contructivism. Educational games, being an active form of media, provide learners with some form of agency and choice, thus making them not entirely instructional. According to Almala (2006), contructivism is applied to educational game design by the design of scenarios that provide players with specific experiences. The player can then reflect on those experiences and draw his/her own meaning and knowledge. However, there exists a weakness when only using contructivist techniques in that no instruction or preset knowledge (educational content) may be used.

Social contructivism. As Sultan, Woods, and Koo (2011) explain, social contructivism is applied to educational games in a similar manner to contructivism, by designing scenarios that provide groups of players with specific experiences. It also includes mechanisms for players to engage in group discussions and/or debates to express individual meaning and knowledge, in order to set the criteria for valid knowledge. This could be done via a voice chat feature in the educational game. However, there exists a weakness when only using social contructivist techniques in that no instruction or preset knowledge (educational content) may be used.

Socratic pedagogy. Adopting Socratic pedagogy into educational game design involves recreating the elenetic process (Rania, 2009). It involves the designing of two aspects of education games. First is a problem based scenario with a finite knowledge base and a definite solution similar to that of the cognitivist approach. Second, are mechanics and/or tools to initiate and walk players through the elenetic process, to filter the through the knowledge base to discover the correct solution (truth). However, it is difficult to incorporate the full extent of Socratic pedagogy into educational games for two reasons. First, is the limited scope that definite solutions and finite knowledge bases place on possible lines of questioning and exploration. Second, if attempted by a single player, the effectiveness of this method will be directly affected by that player's ability to self reflect and question his/her own knowledge and/or beliefs.

From the survey above, the features and drawbacks of incorporating each pedagogical model can be surmised. Thus, in theory, applying a combination of multiple models could lead to a more effective learning experience. Ibrahim and Jaafar (2009) suggest that the appropriate combination of pedagogical models is influenced by the intended educational content. For example, considering the educational content as a topic in mathematics, some form of instruction would be necessary to establish rules, formulae and/or methods. Due to the nature of mathematics, contructivism and social contructivism would be be inapplicable, as the correct answer and/or solution is not based on learners' personal meaning or experience.

Drawing from the study conducted by Riemer and Schrader (2015), the adventure style educational game produces the most pronounced positive effect on learning gains and player entertainment. They explain that this is due to the application of scenarios that task players with problems to solve, while presenting the educational content as a part of the solution. These scenarios will be referred to as *educational scenarios*. Educational scenarios motivate players beyond the learning of educational content by providing game play based goal(s) to work towards. Hence the adventure style teaching mechanism serves as the most promising one moving forward.

When viewing educational scenarios in the context of the five pedagogical models explained above, it was reasoned that they utilise and/or have the potential to utilise all five. However, it was also reasoned that educational scenarios most strongly utilise cognitivist, Socratic and behaviorist pedagogical models. Educational scenarios utilise the cognitivist and Socratic models as they contain educational content (knowledge) that is intended to be taught to players, by presenting them with problems. Players must take the initiative to search for, learn and apply said educational content to solve said problems. Elements of behaviorism are used in the form of instructions regarding the game's mechanics and the conditioning of players to play by the game's rules and context.

Educational scenarios could also utilise both contructivist and social contructivist pedagogical models. Contructivism could be utilised, as educational scenarios provide specific experiences to players due to their more open ended nature. This allows for a higher degree of player agency and choice when compared to quizzes and simulations. However, knowledge gained from this contructivist approach could be highly subjective to each player and thus, may not be related to the intended educational content of the scenario. Educational games that can be played by more than one player at a time could also incorporate the discussions and debates portion of the social contructivism theory.

Lastly, educational scenarios take advantage of the findings of researchers like Michael and R. Chen (2006) and Rapeepisarn et al. (2008) by incorporating entertainment features along with the educational content, to increase the overall effectiveness at teaching. Being the most promising teaching mechanism identified, educational scenarios were further researched into moving forward.

2.2.1.2 Educational scenarios

Due to the promising basis and results of adventure style educational games, learning through playing educational games that are divided into distinct educational scenarios was explored. To reiterate, educational scenarios refer to situations within an educational game where the player is presented with a problem/dilemma which requires the application of the intended educational content to solve. Lepper and Cordova (1992) explain that players overcoming educational scenarios can help further reinforce the educational content due to players analysing and evaluating information in the game to determine logical steps that lead to concrete conclusions. This approach allows the focus of directly learning content to be subverted from the immediate attention of the player. Amory et al. (1999) explain the advantage of this effect being higher levels motivation to complete the game due the player's immersion and enjoyment, with the learning of the content being viewed as a side effect.

When examining the structure of educational scenarios, the works of Huo (2019), Ibrahim and Jaafar (2009), Hullett and Mateas (2009) and Niehaus and Riedl (2009) implicitly split educational scenarios into four stages. These stages have been extrapolated, named and defined (by myself) as follows:

Educational Scenario Stage 1 (Task introduction) *Involves introducing the player to the educational scenario by presenting a problem, explaining that the player's task is to resolve that problem and providing motivation to complete said task.*

Educational Scenario Stage 2 (Content acquisition) *Pertains to the player seeking and finding the educational content he/she needs to complete his/her task.*

Educational Scenario Stage 3 (Content comprehension) *Represents the comprehension of educational content that the player needs to understand before he/she can attempt to complete the task.*

Educational Scenario Stage 4 (Content application) *Involves challenging the player to apply the educational content to solve and/or find a solution to the presented problem, thereby completing the task and proving that the player has learned the educational content.*

These four stages are executed within educational games by educational scenarios. A player would have had to progress through all four stages within an educational scenario before it can be assumed that he/she has learned the educational content.

Challenges concerning educational games however, arise when attempting to balance the educational content with good entertainment value to make the game enjoyable to play. Marsh and Costello (2012) reveal that the educational potential and content of many educational games are overlooked due the lack of engaging gameplay and/or story. Thus, players may not feel motivated to complete the game whilst learning its content, suggesting that educational games may not achieve their primary goal without such a balance. Therefore, factors attributed to entertainment factors were further researched, so as to not have educational content overlooked.

2.2.2 Entertainment design

As explained before, incorporating entertainment design into educational games leads to higher motivation in players to keep playing, thereby increasing the appeal and effectiveness of the educational game at teaching. This suggests that educational games should have a secondary objective of entertaining their players to maximise effectiveness. Unfortunately, the concept what players find entertaining is a highly subjective field which greatly depends on individual preference, independent from good game design. For example, a player who does not enjoy racing and/or cars is unlikely to find racing games entertaining regardless of the quality and features of those games.

To remedy this problem, there exist two generalised design principles that aim to instill the entertainment value of games of any genre if applied correctly. These are *ludology*, which refers to the study and act of playing video games (Aarseth, Smedstad, and Sunnanå, 2003) and *narratology* which refer to the characters, setting(s) and story presented by video games (Jenkins, 2004). These are explored below.

2.2.2.1 Ludology

Ludology is the study of video games, the act of playing them and the players who do so. Two ludological models were identified and explored below.

Nacke and Lindley (2010) denote an affective ludological model centred around the three emotions of boredom, immersion and flow. They argue that design choices in video games in general should be judged based on the most likely of the three emotion(s) they could cause in players. They further detail the three emotions as:

Boredom. Boredom in the context of video games refers to situation(s) where a game fails to interest and/or engage the player. This is an undesirable outcome as players who are experiencing boredom are unlikely to be paying full attention to the game. Boredom in educational games is especially detrimental as a player's learning is reasoned to be negatively impacted from a lack of attention. Nacke and Lindley (ibid.) suggest that game design decisions such as repetitive textures and enemies, over simplified and linear goals, and dull/muted sound and visual effects could contribute towards boredom in most players. Thus, it falls to game designers to ensure that as few design choices that could bore players are made.

Immersion. Immersion is described as the opposite of boredom. Players who are immersed in video games are described as engaged with and giving a video game the majority of, if not their full attention. This is a desirable outcome for video games of any kind as it leads to higher levels of player enjoyment which in turn leads to players playing the game more enthusiastically and/or for longer periods of time. Mayra and Ermi (2011) subdivided immersion into the three distinct forms of sensory, challenge-based and imaginative immersion. Sensory immersion comprises the audio and visual executions of games. This is easily recognisable, since it can be altered by manipulating visual and audio components, such as creating higher fidelity graphics and sounds or playing at a higher native resolution. Challenge-based immersion describes the gameplay experience of a player balancing his/her abilities against the challenges of the game. Challenges in this definition can include different mixtures of physical and mental performance. Imaginative immersion refers to a player's absorption of the narrative of a game or identification with a character.

Flow. According to Csikszentmihalyi (1975), the flow experience refers to the "holistic sensation that people feel when they act with total involvement" with any type of game. He clarifies this as the ideal middle ground between the boredom of low involvement and the anxiety of a challenge beyond the ability of the player. Thus, game design choices should aim to challenge and engage players without overwhelming them. However, J. Chen (2006) notes that the flow experience greatly differs from player to player based on individual preference and skill level.

The affective ludological model is more subjective and empirical due to players' individual choices and preferences in video games. For example, one player might be engaged by a linear experience that has a well developed narrative and characters, while another player might be engaged by an open world with a focus towards freedom of choice. Jennett et al. (2008) note that when designing games using affective ludology, the game's genre and target audience should be clearly defined before design decisions can have their merit evaluated.

On the other hand, is the multi-dimensional ludological model. Aarseth, Smedstad, and Sunnanå (2003) explain that this model examines the mechanics and key aspects of gameplay. Such a model allows for both the design of the game and the manner in which the player interacts with it to be considered. They argue that being a form of media, every video game should also be considered from an artistic stand point which cannot fully be modelled empirically. However, dimensions representing a game's features and functionality can be defined. A combination of Aarseth, Smedstad, and Sunnanå (ibid.), Jarvinen (2007) and Brand (2015) specify the following 11 ludological dimensions of video games:

Perspective. This refers to the view-point experienced visually by the player, determined by the position of the "camera" in the virtual game world. This affects the players field of view and what they can see.

Topography. According to Aarseth, Smedstad, and Sunnanå (2003), a game's topography can be either geometrical, with continuous freedom of movement as in an open world game where the player is able to traverse any point from any angle, or topological, allowing the player only discrete and non-overlapping positions to move between like that of a grid.

Environmental control. This refers to either the player's ability to control or change the game world. A game with dynamic control incorporates the possibility that the game world may change or be controlled. On the other end of the scale, static worlds remain constant regardless of time or player action.

Pacing. If time in the game employs a continuous game-world, in which players are not required to pause or to take turns, it is defined as having real-time pace. The counterpart to real-time pacing would be turn-based pacing, whereby players would have to wait for each other to finish turns before they themselves can act, one such example is virtual chess.

Time represented. Time in games can be presented in terms of real-world time, abstract or arbitrary time. Games with a real-world time scale employ that time in the virtual world, which moves at a speed mirroring that of the real physical world. In an arbitrary time representation, the player experiences time on a different scale, compared to real time. An abstract time scale would suggest that the time in the game world never changes.

Teleology. Games can be split into having either finite or infinite teleology. Games that exhibit no definite way of winning or conclusion are said to have an infinite teleology. Other games are determined by a clear successful outcome of the player and are described as having finite teleology, in terms of a final goal.

Player structure. Refers to the number of players who can simultaneously play the game and the way they interact with each other. The simplest player structure is the single-player game, in which only one person is able to play the game at any given time and can only interact with virtual objects. In contrast to this, multiplayer games refer to games in which more than one person can simultaneously play.

Mutability. This is described as the mechanic that games use to control player behaviour, with the motivation of various rewards. Mutability refers to the mechanic in rewards that affect the player's standing and ability to play the game. Games with no mutability are static. Mutability also extends to that of temporary and permanent rewards.

Savability. The ability of the player to save the game-state and the progress they have made. Some games employ conditional savability, in which there exist specific points and times where the player is able to save. A game that can be saved at any time in the progress of play is unlimited in this dimension. Games with no savability at all are both rare and not recommended by Aarseth, Smedstad, and Sunnanå (2003).

Determinism. Refers to the extent with which the player is able to predict the mechanics and outcome of the game world. Non-deterministic games proceed without predictability, as they use a randomised algorithm to denote an experience which is unlikely to be exactly replicated if the player were to play again. By contrast, in deterministic games, the player will experience exactly the same response from the game's mechanics and narrative every time the player accomplishes a goal.

Rules. Aarseth, Smedstad, and Sunnanå (ibid.) argue that rules can be classified into three dimensions based on time, space and function. Space rules are determined by a player's character being in a particular location, thus changing the condition of the game-play (such as completing a level or activating a trap). Time-based rules are those that change the status of play in an important way, based on a clock or timer. Function-based rules gear more towards fulfilling a specific objective found in games.

2.2.2.2 Narratology

Aarseth (2012) describes narratology as the study of the narrative structures that a game could present. He explains that narratives in video games are told using the four components of the game world, objects, characters and events. These are further detailed:

Game world. The game world refers to the virtual world that the player interacts in and with. Aarseth (2005) explains that game worlds present narratives via environmental story telling. This is mostly conveyed by the landscape of the game world. For example, a decrepit and empty city could convey the message that some form of disaster occurred and might have killed or caused the inhabitants to relocate. He also explains that there exist five types of landscapes. These include the linear corridor, the one room, the fragmented rooms, the hub-shaped labyrinth and the open world. The differences in how players traverse through and interact with each type of world allow for different pacing and tones of narratives. For example, the linear corridor and one room approach would funnel players in a specific direction where they can be presented with the environmental story in a specific order and time. However, in an open world game, the player is free to explore and might encounter different environmental stories in different orders and times.

Objects. Objects refer to items that can be found within the game world. Aarseth (ibid.) explains that objects can present a narrative to players in two ways. First, objects can enhance the environmental story telling of worlds through their context. Going back to the previous example of a destroyed and empty city, corpses and weapons littering the ground could further indicate that the city was destroyed by an external attack rather than a natural disaster. Second, objects can present narratives to player's via directly exposition. Using the same example, exposition could come in the form of a diary that the player is meant to find and interact with. The diary could directly convey what its owner was experiencing during the attack on the city and shed more light on the narrative.

Characters. This refers to the NPCs found in the game world. Similar to objects, Aarseth (2012) explains there are two ways that characters contribute towards the narrative. The first is enhancing the environmental story telling of worlds through context. For example, a player suddenly coming across many soldier NPCs could indicate that he/she has entered military territory. Second, is that characters can directly tell and/or contribute to the narrative via their dialog and or/actions. For example, an NPC might physically take the player on a tour of the destroyed city while explaining to him/her what happened.

Events. Events are the things that happen in a game and very often incorporate the previous three components to tell a story. Aarseth (ibid.) explains that there are two types of events that can convey a narrative. The first is a non-playable event where the control is taken away from the player and is presented with narrative elements akin to a movie. This is referred to as a cut scene. Using the previous example, a cut scene showing the attack and destruction of the city with narration could provide the player with the narrative of why the city is destroyed and abandoned. The second is a playable event, where the player is interacting with the game world, characters and objects while the narrative is unravelling around him/her. Using the previous example, this could come in the form of the player actively attempting to defend the city from attackers only to be overwhelmed and forced to retreat.

Due to video games being an active form of media, Arsenault (2014) suggests that a mixture each of the types of the four components is optimal when presenting players with narratives. Jenkins (2004) echoes this approach to narratology to provide player's with a variety of ways to engage with the narrative. He details that differing degrees of the four components can be combined to form the following four types of narratives:

Evoked narrative. This type of narrative draws upon a pre-existing story and/or franchise. A prominent example of this would be any game set in the "Star Wars" universe; since Star Wars, as a movie franchise onto itself, exhibits its own narrative and world. Thus, "Star Wars" games could be said to be part of and build upon the general Star Wars narrative. Thus, it is possible for narrative architecture to be determined by either audiences or by game writers and designers.

Enacted narrative. This occurs through experience with the game elements; including a combination of the game world, character dialog, context of objects, cut scenes and story telling within playable events. In most cases, games use a combination of the four components, rather than just one to set up to deliver the narrative. Furthermore, the player could also be offered decisions that branch the narrative into different but predetermined directions. This allows different players to experience different narrative within the same game.

Embedded narrative. This narrative is present when the player can uncover further details of the narrative that are not directly presented. This could be accomplished by stumbling across spaces and objects or artefacts that have embedded meaning or importance to the overarching narrative.

Embedded narratives can also share a connection with enacted narrative. One example is that of a player discovering secondary narratives that relate back to events in the primary narrative through optional exploration.

Emergent narrative. An emergent narrative occurs when the player imagines or authors the story himself/herself, by building the world he is playing in. Players of games with emergent narrative usually have control over time frame, weather, characters, infrastructure, etc. which they create in their own image to imagine their own customised story

2.2.3 Support design principles

In addition to the educational and entertainment principles, several additional support design principles were identified while researching principles of good video game design. These include *abstraction* and motivational factors.

As Bateman and Boon (2005) explain, abstraction refers to a video game establishing its own internal rules and consistencies. They go on to state that players take note of established rules and notice if the game's abstraction is not adhered to. This presents the chance to potentially break the player's mental and emotional immersion of the game and subsequently decrease motivation to continue playing. As such, Cowley and Charles (2016) suggest the early interweaving of game rules with the narrative and context of the world to uphold the abstraction. This would theoretically allow players to stay as invested as possible and thereby staying motivated learn the educational content.

Habgood and Ainsworth (2011) explain the need for motivational factors to be incorporated into educational games. They go on to distinguish between two kinds of motivators, intrinsic and extrinsic. *Intrinsic motivators* refer to approach focusing on motivating to learn through the enjoyment of learning and rewards related to completing the game and/or objectives in itself. Conversely, *extrinsic motivators* pertain towards the threat of punishment or the effects of failing to accomplish said goal/learning. This could be illustrated using the analogy of a football player playing his best due to his enjoyment of the sport and competition (intrinsic), or fear of the shame of defeat and punishment he may face from his trainer (extrinsic).

Lepper and Henderlong (2000) suggest that an intrinsic approach and rewards would better serve the purpose of an educational game. It was reasoned that imbuing the player with a desire to continue playing to completion would allow for more comprehensive learning, along with overcoming pitfalls and mistakes voluntarily without need for external reward or punishment. Combining problems found in game with intrinsic motivators could therefore create scenarios that players are motivated to solve and thereby learn from.

This is due to players gravitating towards actions and mechanics that they perceive as enjoyable. Therefore, it was reasoned that removing the focus from content that many learners deem as dry/unenjoyable in place of content that they do enjoy would increase learning ability. Saljo (1979) also notes that players' motivation is also affected by the difficulty of the educational content relative to their skill level. Thus, one of the challenges of educational game design would be balancing motivators and content difficulty based on the abilities of the target player(s). Methods, techniques and suggestions on how intrinsic motivators could be implemented in an educational game are delved into in the following chapters.

With educational game design explored, the next section delves into adaptive both entertainment and educational games.

2.3 Adaptivity in video games

As the focus of this study is directed towards the adaptivity of educational games, examples of adaptivity in video games were explored. Adaptivity in video games is defined as when a game alters or changes itself to better fulfill its primary purpose. Lopes and Bidarra (2011) note that the focus of adaptivity differs in entertainment and educational games due to the differences in their primary purpose. Entertainment games' primary purpose is to entertain their players, thus adaptivity is directed towards making players have more fun. Educational games on the other hand, have the primary purpose of educating players, making their adaptivity be geared towards better teaching players. However, Harteveld et al. (2010) note that there is overlap in the adaptivity of both types of games in the manner that they attempt to find a balance between their players' skills and the challenges presented. This is done with the intention of always presenting players with a level of challenge that they are capable of overcoming. Methods and examples of adaptivity in both entertainment and educational games were further researched.

2.3.1 Adaptivity in entertainment games

Examples of and techniques used in entertainment games to achieve adaptivity are explained below.

2.3.1.1 Dynamic difficulty adjustment

Chanel et al. (2011) explain that one of the most commonly used form of adaptivity in video games is called *dynamic difficulty adjustment* (DDA). DDA is defined as the automated changing of parameters, scenarios and behaviours in real-time, based on a player model containing predefined metrics of proficiency. Metrics of player proficiency vary based on the way in which a video game challenges the player. DDA is also implemented without the player's knowledge to avoid patronising the player and potentially making him/her lose interest in the video game.

Lopes and Bidarra (2011) explain that DDA can be applied to NPC behaviour to alter the extent of challenge presented to the player. This is so that players of all skill levels may be sufficiently challenged without feeling bored or discouraged. Some examples of applications of DDA being applied to NPC behaviour in entertainment games include:

1. Mario Kart Wii. Nintendo (2008) developed a racing game that would increase the maximum speed of losing players while reducing the maximum speed of winning players to provide more challenge to the fast players and a helping hand to struggling ones.

2. Resident Evil 4. Capcom (2005) made an action game that would alter enemy health, weapons and reaction time based on how long and how much damage the player took in the previous encounter.

3. Crash Bandicoot N. Sane Trilogy. An action and platforming game made by VicariousVisions (2017) that alters the frequency of checkpoints, the quality of items and the speed of obstacles based on the number of times the player died in each level.

Furthermore, Brathwaite and Schreiber (2008) explain that DDA can also be used on game mechanics to also alter the extent of challenge. RemedyEntertainment (2001) showcase DDA applied to game mechanics by varying the strength of the "aim assistance" mechanics in their 2001 release "Max Payne". Aim assistance is defined as a mechanic that aids players without their knowledge by automatically shifting their aim to the intended target regardless of the player's input.

“Max Payne” alters the strength of it’s aim assistance mechanic based on the number of hits taken and the accuracy of each player, in order to alter the difficulty of aiming and shooting. The strength of aim assistance refers to the maximum distance that the mechanic is allowed to automatically shift the player’s aim by. Players who have been classified as struggling receive an increase of aim assistance thus allowing the mechanic to correct their aim from a farther distance.

Lastly, “Metal Gear Solid 5”, made by Kojima (2015), serves as an example of DDA applied to both NPC behaviour and game mechanics. Metal Gear Solid is an open world, action-stealth game that highly encourages players to carry out military infiltration operations non lethally, without being spotted by and without raising enemy suspicion. Kojima (ibid.) explains that enemy NPCs will take note of player tactics and adapt to them. For example, enemies will react to players who tend to infiltrate during the night by placing more guards at every garrison during the night time. Additionally the guards would be instructed to remain in pairs to reduce the chances of being snuck up on.

In addition to the adaptive NPC behaviour, Metal Gear Solid 5 also implements adaptive game mechanics. Examples of this include enemies procuring body armour if the player is deemed to be skilled at using tranquilliser darts and enemies using night vision goggles if the player has established a preference infiltration during the night.

2.3.1.2 Procedural content generation

Shaker, Togelius, and Nelson (2016) define *procedural content generation* (PCG) as the ability for a game to allegorically create its own content with either limited or no player input. They go on to define the content that can be generated by PCG as “most of what is contained in a game”. This includes the game world and objects, narrative and scenarios and/or objectives components. They clarify however that they do not classify the game engine, game mechanics and the AI driving NPC behaviour as content that can be procedurally generated as doing so could render a game nonsensical and unplayable.

However, Shaker, Togelius, and Nelson (ibid.) also explain that PCG can be further split into generic PCG and adaptive PCG. Generic PCG refers to where content is generated without taking player behaviour into account. Examples of this include:

1. Minecraft. An open world, exploration, crafting and adventure game made by Mojang (2011) that uses PCG to generate its game world and objects. The PCG process involves using a randomised code called a “seed”, to algorithmically determine the types and sizes of landscapes along with placements of villages, NPCs , caves and other game objects.

2. The Binding of Isaac. Nicalis (2014) developed a top down dungeon shooter that uses a similar randomised “seed” codes to procedurally generate the lay out of its levels, enemy and boss placements, and items.

3. Civilization VI. This is a turn based strategy game developed by Firaxis (2016) that uses PCG to alter the layout of game worlds for every play session.

Conversely, adaptive PCG records and analyses a players interactions and behaviour within a game to generate content that might appeal more to each player. An example of an adaptive PCG framework is the *“Experience-Driven Procedural Content Generation”* (EDPCG) model developed by Yannakakis and Togelius (2015). They explain that using their model can allow for the players’ experiences and preferences to be modeled, allowing for personalised content to be generated. Examples of adaptive PCG applied to entertainment games include:

1. Galactic Arms Race. This is an action, role playing space shooter developed by EvolutionaryGames (2012) where weapons presented to players evolve and change based on their frequency of use and proficiency.

2. Super Mario Bros. A platform and adventure game developed by Nintendo (1985), that was modified by Pedersen, Togelius, and Yannakakis (2010) using the EDPCG model to generate whole levels based on players' preferences.

2.3.1.3 Director systems

Valve (2008) explains their director system as collection of artificial intelligence routines, designed to make each play session of their "Left 4 Dead" and "Left 4 Dead 2" games different and unpredictable. These routines are applied towards adaptation in the game world and objects, narrative and scenarios and/or objectives components. According to Valve (2009) the director system's adaptivity requires an open world with multiple possible objectives, routes and enemy types each with differing degrees of difficulty attributed to them. The director system chooses a different objective and route for each play session with the remaining possible objectives and routes being initially blocked off and/or made unavailable.

Valve (2008) states that, the director system also monitors each player's behaviour and ascertains their skill level at specific points in each game world. Factors that influence a player's skill level include the amount of damage taken, the number of times that player has been killed and the number of times that player has been killed by specific enemy. The director system can then execute adaptations at predefined points by comparing the players' overall skill level to developer defined difficulty of the objective and route chosen. Possible adaptations include:

1. Game world and objects. The routes available for players to traverse and the weapons and items provided can change based on players' skill level.

2. Scenarios and/or objectives. The overall objective, types and quantities of enemies, and the frequency at which each type of enemy spawns can change based on players' skill level.

3. Narrative. Valve (2009) Argue that the dynamic nature of the objectives, routes, weapons, items and enemies leads to the narrative of each play session being adapted to its players.

2.3.1.4 Goal based systems

The goal based system is another method to make NPC baheviour adaptive. Cakaj (2010) explains this system as a collection of multiple possible goals for NPCs, each with multiple routes to accomplish. The NPCs would then use a logic system based to determine the most appropriate goal and/or method to pursue, based on players' present and past behaviour. This allows NPCs to react differently to players depending on the situation and identified patterns in players' behaviours. Examples of the goal based AI system include:

1. First Encounter Assault Recon (FEAR). Monolith (2005) developed a first person shooter game where enemy soldiers behaved akin to a military squad having three potential goals of attacking the player, holding their position or retreating. Their goals would vary based on a combination of the weapon

they witness the player holding, the amount of damage done to the player and the number of their squad members that had been killed by the player. Based on their goals, the NPCs could adapt their behaviour. Examples include taking advantage if their numbers or superior weapons to aggressively attack the player, losing sight of the player and proceeding to slowly search, deciding to adopt a defensive position and panicking and retreating if loses are too great.

2. Pro evolution soccer 2018. Konami (2017) developed a foot ball simulator whose opponent AI system specifically records and observes the player's actions for patterns. Using established player attack and defense patterns, individual NPCs can be given altering short term goals in an attempt to counter and foil the players attempts to score and block.

2.3.1.5 Player choice based adaptation

According to Lopes and Bidarra (2011), player choice based adaptations can be used to make narratives in entertainment games adaptive. This involves creating large scale narratives with multiple branching story lines and outcomes. Determining the storylines that are shown and not shown can be based on direct player choice, subtle examining of player choice or a combination of the two.

Direct player choice involves explicitly presenting players with choices that will branch the narrative depending the the option picked. Subtle examining of players' choices involves either counting or assigning points to particular actions that a player could perform, akin to a scale. Specific narrative branches can then be presented based on the count or number of points. Examples of player choice based adaptations include are:

1. Heavy Rain. An crime and mystery game developed by QuanticDream (2010) with 17 different endings and narrative paths to reach them, controlled solely by choices that are directly presented to the player.

2. Metro exodus. An open world action and survival first person shooter made by 4AGames (2019). This makes use of the second method to adapt it's narrative. At many points during the game, players are presented with "moral" choices such as freeing prisoners and sparing soldier who have surrendered. Players may choose to do what the developers have deemed to be morally correct, thereby earning morality points. The dialog and final ending of the game adapts to the player based on the number of morality points he/she has accrued.

3. Mass effect 3. Bioware (2012) created a science fiction epic that makes use of both methods of adaptive narrative. Players, at many points are presented with options to say and do specific thinks that impact the course and direction of the narrative. Furthermore many actions can either be classified as "paragon" and "renegade" actions. Performing each type of action will award the player with "paragon" and "renegade" points. Entire dialog options, missions, story threads and endings can either become available of unavailable based on players' number of "paragon" and "renegade" points.

2.3.2 Adaptivity in educational games

With examples of adaptivity in entertainment games explored, examples of adaptivity in educational games come next.

2.3.2.1 Player proficiency adaptivity

Westra and Dignum (2010) propose that adaptations in educational games should be based on players' individual proficiency at the educational game and content. This is so that adaptations can aid players who may be struggling or further challenge players who are thriving when it comes to learning. Liu, Moon, and Kim (2020) explain that there exist two approaches to player proficiency.

The first approach is a competency based proficiency attempts to model players' intellectual and emotional states and adapt game components to match them. This approach underscores the development and delivery of game resources and tasks tailored to players' competency levels. They propose the use of a threshold rule to determine when to employ adaptive game mechanics. For example, an adaptive educational that uses computational algorithms to yield competency scores that indicate players' cognitive level and/or affective state. If players' competency scores, derived from their in game performance, pass an arbitrary threshold value that the game developer sets, the game immediately provides adaptivity, such as giving support or adjusting the difficulty level. Examples of competency based proficiency adaptivity using thresholds include:

1. Jagust, Boticki, and So (2018) created an educational game design to teach mathematics that adapted the time limits given to players to solve math problems using the performance and errors monitored while solving previous problems.

2. Clark et al. (2016) applied two thresholds to the Fuzzy Chronicles, an educational game developed to support students learning about Newtonian dynamics. Game play is divided into levels, each containing a challenge to navigate a spaceship through obstacles to a target location. Players complete levels of the game by considering the appropriate magnitude and direction of forces to propel their ships in a desired direction at a desired speed. Players' performances are scored based on the time they take and the number of errors they make in completing each level. Players are provided with assistance either when they have crossed a time threshold or an allowable errors threshold. Both thresholds vary from level to level based on players' performance in previous levels.

3. Sandberg, Maris, and Hoogendoorn (2014) enhanced an educational game designed to improve and test players on their English language proficiency, called MEL. They made the difficulty of quizzes that MEL would test players with adaptive, in terms of the number and difficulty of questions posed. Factors affecting the adaptive difficulty included players' success rate, the number of hints given and the last difficulty level each player was rated at. Crossing predefined thresholds of success rates and number of hints allowed cause adaptivity to alter the next quiz.

The second approach is a preference based proficiency aimed at modelling players' interests and style of play. This approach employs the development and delivery of game resources and tasks tailored to players' liking. Examples of preference based proficiency adaptivity in education games include:

1. Bontchev and Georgieva (2018) used experiential learning theory to identify the four player types of the competitor, dreamer, logician, and strategist. Players would be categorised based on their in-game actions and performance. Each type of player is the presented with different instructional goals and in-game strategies tailored to their style of play. They further argue that a validated and reliable player style measuring instrument is important for providing "style-based adaptation" in educational games.

2. Shabihi and Taghiyareh (2017) made use of questionnaires to sort players into one of four personality types. They proceeded to develop an educational game and tailor made content for each personality

type. They further divided players into two groups. One that was presented with the content made for their self identified personality type and the other presented with mismatch content. Although their game was not capable of dynamically adapting and changing content presented to players, their results showed that players who played the game with preferred game content outperformed those who did not in terms of in game scoring.

2.3.2.2 Adaptive narratives

Lopes and Bidarra (2011) explain that there is a strong interest in adaptive and customised story-based experiences in educational games. This is done by modelling players' preferred story options within a finite pool of narrative options designed by developers. Players can then be presented with the option that matches or most closely matches their preferred narrative. This was implemented by the following researchers in the following ways:

1. Wang et al. (2017) developed an interactive narrative planner that they paired with the educational game Crystal island. The interactive narrative planner monitors the players' behaviour and adapts the sequence of events that unfold. Adaptation occurs as the alteration of a recurring series of one or more adaptable story events that can unfold into several different possible narrative trajectories, each leading to potentially different player experiences and outcomes. Wang et al. (ibid.) illustrate their system with the following example: "the player explores the game environment for a while, then talks with a sick NPC, Teresa, which triggers an adaptable event, for which the interactive narrative planner selects the narrative planning action of allowing Teresa to reveal limited information about her symptoms. Later the player character continues exploring the virtual world, conducting tests in the virtual laboratory, and submits a diagnosis worksheet of her anticipated solution to the science mystery, which unfortunately turns out to be incorrect. This triggers another adaptable event, leading the interactive narrative planner to select the adaptable narrative planning action of providing a detailed explanation of the player's errors. These adaptable events are recurring, i.e., each event can be triggered multiple times, and adaptable event occurrences are determined by player actions and narrative rules".

2. Sharma et al. (2007) developed a drama management system which determines players' preferences in narratives from the behaviour of earlier players. This method requires training data taken from initial play sessions without using adaptivity. The training data includes narrative choices made and is analysed for patterns. Post training, adaptivity can be activated and subsequent players' choices are matched against the pattern that fits closes. The narrative corresponding with the players' predicted pattern can then be displayed.

3. Thue et al. (2007) present an interactive narrative generation system which presents different combinations of multiple story events selected from a pool. The system determines the combination of story events to present, by modeling players' narrative preferences according to five predetermined player types. The player types include fighters (who prefer combat), power gamers (who prefer gaining special items and riches), tacticians (who prefer thinking creatively), storytellers (who prefer complex plots) and method actors (who prefer to take dramatic actions). Players' preferences are determined by matching their in game actions and choices to the predetermined player types. Story events are annotated for inclusion in the narrative based on the player type identified.

4. Fairclough (2006) made use of a case-based approach to adaptive narratives by monitoring players' evolving relationship with salient NPCs. Each permutation of relationships with all salient characters require their own narrative thread. The narrative presented to players can then adapt based on the narrative choices they make with each salient NPC.

2.3.2.3 Adaptive educational scenarios

Lopes and Bidarra (2011) explain adaptive educational scenarios as ones that add, remove, replace, alter and/or generate events within themselves in order to better teach players. The is done by mapping the content an educational game offers, the manner at which the scenarios challenges players and a model of what players' comprehend with each other. Examples of adaptive educational scenarios in educational games include:

1. Ashmore and Nitsche (2007) proposed a new quest generator to include in the Charbitat, developed by Nitsche et al. (2006). Quests consist of key and lock puzzles, where obstacles must be overcome using educational content to a find a key to unlock the lock and complete the quest. The game world of Charbitat is procedurally generated and segregated into tiles, each of which contains a quest. Quest generation occurs when player's traverse to new tiles. Possible locations for keys and locks and the types of obstacles presented vary depending on the players' progress and the layout of the generated world tile. Quests become unique for each player because they are influenced by the generated game world and his/her previous actions in completing quests.

2. Sullivan, Mateas, and Wardrip (2010) proposed an approach called the Grail Game Manager. This a rule-based system which dynamically generates quest structures using the player's past behaviour and the current world state. The Grail Game Manager is able to decompose quests from its quest library into separate components (goals, actions, rewards, NPCs, dialog options) that can be dynamically recombined upon generation. This process filters possible quest components through preconditions based on players behaviour, success and the current world state, thus creating a personalised experience.

3. Pelanek (2016) developed an adaptive educational game aimed at teaching geography. His game employed the quiz method to teach players and used a system to adapt the difficulty of questions to players. This was done in three steps. First is the estimation of the probability that a player has knowledge about a topic, before being asked a question regarding that topic. The estimate is based on previous answers of that player and on the answers of other players on that topic. Second is the estimation of a player's actual knowledge on a topic, based on the estimation of prior knowledge and a sequence of previous answers of that player on that topic. Third is the construction of a suitable question, based on the estimation of a player's knowledge and the recent history of answers.

4. Gutierrez and Atkinson (2011) developed an adaptive system to provide adaptive assistance in an educational game meant to teach foreign languages. Their system aims to categorise type of error made by players. This is done by analysing the text of the wrong answers given by players. The system then chooses the most appropriate response to aid players in resolving the error. The types of errors and the best responses to address them were taken from a data set of student-teacher interactions. This implementation included seven different types of responses categorised under two meta-linguistic feedback strategies. The the first would directly present the player with the correct answer, while the second prompted players to correct their answer when erroneous.

2.3.2.4 Knowledge space theory

Knowledge space theory can be used to achieve adaptivity in educational games. The knowledge space theory was conceptualised as a behaviourist approach to model the structure of a learner's *knowledge state*, as explained by Dietrich et al. (2019). A learner's knowledge state is defined as his/her understanding of a specific *problem state* and the possible actions that he could take to solve it. A problem state pertains to a specific issue that the learner would need to find a solution to. A learner's *knowledge space* refers to all the learner's possible knowledge states. The learner's knowledge space directly impacts the problem space, which is the set of all problem states a problem may take.

For example, consider the problem of a learner being unable to leave a room. Problem states may include the door being locked, the door being stuck or pushing/pulling the door the wrong way. Possible knowledge states would be the learner recognising the possible problem states and subsequently checking if the door is locked, checking for obstructions and attempting to push/pull the door the other way.

Whenever a learner performs an action, the problem state changes. This represents a cycle where a learner's action will affect the problem state which would then update the learner's own knowledge state inducing a more appropriate follow up action.

Actions may be scored by their utility, where the utility of an action lies in its contribution to improving the problem state. For example, if the problem state was to solve the issue of a lamp not lighting up, actions with positive utility could be to check that there is electricity flowing to it or to check if the bulb is blown. Negative utility actions could include shaking the lamp or continuously turning it on and off.

Positive or negative changes to the problem state based on a learner's action could then serve as input for determining that learner's current knowledge state. To elaborate, a learner who consecutively performs negative utility actions could be described as having a poor knowledge state regarding that problem state. Thus, steps could be taken to increase the learner's knowledge state so that he/she is able to resolve that problem state.

Peirce, Conlan, and Wade (2008) created the ALIGN system as a possible implementation of the knowledge space theory to offer adaptive assistance and support to a player while playing an educational game. The ALIGN system accomplishes this by taking the following four steps:

1. Detection of a player's action.
2. Accumulation of player's actions and previous adaptations to predict the state of the player's knowledge state.
3. Calculation of the action the player is most likely to execute based on his/her knowledge state.
4. Offer of assistance and/or support if the player's most likely action is of negative utility.

Kickmeier-Rust et al. (2006) illustrate a practical implementation of the ALIGN system by the development of an adaptive educational game called ELEKTRA. This game was aimed at teaching students high school physics and employed adaptive support systems through altering the behaviour of NPCs to give a combination of hints and motivation to players who might be struggling. Similar to games that employed DDA, adaptivity in the ELEKTRA game was carried out without the player's knowledge.

2.3.2.5 Categories of adaptivity in educational games

The works of Csikszentmihalyi (1990) explain that adaptivity in educational games can be divided into two categories. These include *active adaptivity* and *passive adaptivity*. Active adaptivity involves the adaptation of aspects in video games that the player is expected to engage with and/or execute his/herself. Passive adaptivity involves adaptation of the support systems in video games that aid the player with performing the tasks that he/she are expected to. These two adaptivity areas overlap in terms of the *flow experience* they create. The flow experience refers to the player being challenged and made to learn at a pace he/she enjoys while simultaneously being entertained.

Koster (2004) explains that there exist three areas of active adaptivity to consider:

Ability. Cowley, Charles, et al. (2008) explain that players typically learn and improve at a game system through completing tasks by applying skills that they would have picked up while playing it. Adaptivity involves the game assessing the proficiency of the player in a given task and adjusting to be slightly above the player's proficiency to ensure the player is always being challenged.

Learning. Espejo and Harnden (1989) state that adaptivity in learning revolves around controlling the way in which players are taught content and/or aspects of the game he/she is expected to engage with. By estimating the player's preferences in being presented information, learning adaptivity can alter the manner in which information is presented and/or the rate at which information is given.

Interaction style. Bateman, Lowenhaupt, and Nacke (2011) explain that adapting games based on player interaction style involves changing the type of challenge presented, reward structure and/or aesthetics to better suit the player's preference. This presents the challenge of the game having to profile the player's preferences.

Three possible forms of passive adaptivity are put forth by Csikszentmihalyi (1990):

Motivational support. This refers to a system that can provide hints and encouragement to players whom may be struggling or losing interest to continue.

Meta-cognitive support. Kapa (2001) explains this as a system that makes the player consider his/her own thoughts. This is a way for educational games to consider gaps in a player's knowledge and actively work to cover them.

Meta-reflection support. This eludes to a system that encourages the player to consider different points of view and potentially approaches to achieve a goal that he/she previously failed at.

2.3.3 Similarities and differences

When examining the examples of adaptivity in entertainment and educational games, there exist similarities in the adaptable video game components used and way that both examine players' behaviour and choices in order to inform adaptation. However, the manner in which game components are adapted differ. This is reasoned to be due to the difference in the primary goals of both types of games.

Entertainment games employ adaptations to increase the amount of fun their players have. Whereas, educational games use adaptations to improve the learning in their players. A comparison illustrating the difference between the two is Resident Evil 4's (Capcom, 2005) and the Fuzzy Chronicle's (Clark et al., 2016). Both games monitor the amount of time players take to complete levels and compare the completion times to time thresholds that were established through earlier play testing. They differ from this point in the sense that Resident Evil 4 alters the enemy health, weapons and reaction time to adapt the difficulty of combat to be closer to players' skill at combat. While the Fuzzy Chronicles alters the amount of assistance given to players to help them apply the educational content they have learned.

A similar comparison between Pro evolution soccer 2018 (Konami, 2017) and the educational game made by Pelanek (2016) yields a similar result. Pro evolution soccer 2018 uses an AI system to record and observe players' actions for patterns in order to block their attempts to score while using different techniques to attempt to score over players. On the other hand, the educational game made by Pelanek (ibid.) aimed to categorise the knowledge level of geography that players possessed and pose quiz questions of a difficulty suited to each player.

These comparisons suggest that adaptations made in entertainment games lean more towards challenging players skills, while adaptations made in educational games lean towards aiding players in learning and/or testing knowledge. Despite the difference in the manner that adaptation is implemented, the adaptive frameworks used in both entertainment and educational games bear many similarities. Thus, focus now shifts to extracting and defining an adaptive framework for video games in general.

2.4 Adaptive framework

Although the examples of adaptivity explored above differ in terms of aim and implementation, they employ a common framework to achieve their goals. This framework can be split into the addressing of the following four issues:

Adaptivity Issue 1 (What to adapt) *This involves determining video game component(s) whose adaptation could benefit the video games' purpose (entertainment for entertainment games and learning for educational games).*

Adaptivity Issue 2 (When to adapt) *This refers to determining the situations and/or conditions where adaptation could be of benefit to players for each of the adaptable components identified in Adaptivity Issue 1.*

Adaptivity Issue 3 (How to adapt) *This refers to the determining of element(s) to adapt and what they should be adapted to when condition(s) identified in Adaptivity Issue 2 are met.*

Adaptivity Issue 4 (Data collection) *This pertains to monitoring for and/or collecting the relevant data to determine when the situations and/or conditions for adaptation identified in Adaptivity Issue 2 occur and/or are met.*

Methods to address each of the adaptivity issues are explored below.

2.4.1 Adaptable components of games

Concerning Adaptivity Issue 1, although the primary goals of entertainment and educational games differ, the components they comprise remain quite similar. A review of the works of Reeves and Read (2009), Lopes and Bidarra (2011), Brathwaite and Schreiber (2008), Juul (2005) identified 11 elements that video games in general could comprise:

Non player characters (NPCs). This refers to the characters other than the player, that can interact with players and impact the game worlds. Examples include enemies that attempt to harm the player and a character that could sell helpful items. Lopes and Bidarra (2011) explain that two aspects of NPCs can be made adaptive, their behaviour and their appearance. NPC behaviour is described as the actions and patterns non-player characters execute. This could range from attacking the player to engaging the player in dialog. NPC appearance simply refers to how NPCs look, ie their clothing, ethnicity, gender, etc.

Environments. The space and world that the video game takes place in is represented by the environment element. For example, the Mass effect games have multiple cities and locations, spanning across multiple planets, each with their own unique visual design. Information and interactions take place within this world to further the player's understanding and interest.

Narratives. Narratives refer to the stories that are told within the game world(s) and their significance towards the world(s). For example, a game might tell a story of a missing person and the attempts made to find him/her. This plays a large role in setting the scene, context for interactions and creating interest and motivation for players.

Information. This comprises of all instructions, knowledge and/or educational content that is found within a game.

Feedback. These are the effects/outcomes in the game to make the player aware of the effects of their actions, so as to indicate what subsequent action(s) he/she could take. Examples of this range from a sound that plays when the player interacts with something to an enemy crying out and falling over due to being shot by the player.

Tasks. Refers to the objective(s) given to the player that must be accomplished to either proceed or complete the game. For example players may be given the objective to navigate from a start point to an end point. In this example, the task element is represented by the direct objective of navigating from the start point to the end point.

Levels. Levels refer to the challenge(s) that are presented to the player to overcome to accomplish the task(s). Building on the previous example, the levels element could manifest as physical obstacles or enemies attempting to hinder the player from getting to the end point. The player would need to overcome the challenges provided by the levels element in order to complete his/her overall task. When further examining the levels element, the approach of Liu, Moon, and Kim (2020) regarding challenges that players may face, proved to be directly relevant. Similar to how they argue that players' proficiency at overcoming challenges can be measured across two dimensions, it was reasoned that the challenge(s) posed by the levels element could also be split into the following two elements:

Level style. The level style element represents the first dimension which refers to the type of challenge(s) presented to the player. Each level style corresponds to a type of challenge that a player could be given. To use the previous example, different level styles could manifest as physical obstacles or enemies preventing players from reaching their goal from reaching their goal. One style presents a navigational challenge while the other presents a combat challenge.

Level difficulty. The level difficulty element represents the second dimension which refers to the severity of challenge presented to the player. This can be implemented by altering the amount of aid given to the player during a challenge and/or how severe the challenge is. Building on the previous example, the first method could involve adapting the number of hints given to players for the navigational level style and the number of helpful items presented for the combat level style to each player. Conversely, the second method could involve adapting the number of obstacles presented for the navigational level style and the number of enemies to eliminate.

Motivators. Motivators are a type of support system present in a game used to aid the player with progressing when difficulties are experienced. Examples of this include hints explaining how to progress or items that could make combat easier.

Rules/Mechanics. These are the actions, systems, patterns and limitations that are possible within a game that must be followed once established to maintain continuity and a sense of fairness. For example, the game Minecraft consists of a world of blocks of varying materials. It is a game rule that players can break down and collect any block except for "bedrock". Thus when a Minecraft player encounters bedrock, he/she should understand that it cannot be broken down or collected.

Communication. This element refers to the way in which the game conveys information to its players. Examples include a game where a player is given the necessary information by speaking to NPCs, and a game where information is presented in text form as pop ups.

Time pressure. This is a time limit imposed on accomplishing tasks to increase tension and excitement in players. For example, a game that tasks players to escape from a room could impose a time limit after which players are considered to have failed.

Due to the diversity of video game design, some video games will contain/employ a subset of these 12 elements. Moving forward, these identified elements could serve as a basis for the adaptable elements in educational games. These 12 elements will be referred to as the *game element set* (GES) in future chapters.

2.4.2 Adaptation conditions

Regarding the challenge of when to adapt, posed by Adaptivity Issue 2, the examples explored in Section 2.3 make use of conditions that must be met before adaptation can take place. These are referred to as *adaptation conditions*. Unfortunately the adaptation conditions used in the games explained above were ad hoc in nature and specific to the purpose and gameplay of their respective games.

For example, the game Resident Evil 4 utilised the adaption conditions of players' shooting accuracy, level completion time, deaths and amount of damage taken being over or under thresholds that were set from play testing. Conversely, the game that Sandberg, Maris, and Hoogendoorn (2014) developed, employed the adaptation condition of the completion of a quiz so that the next quiz presented may be more adapted towards individual players.

Thus, there exists no general method for the derivation of adaption conditions. However, the logic and method used to derive the adaption conditions of this study has been detailed in Section 3.3 in an attempt to make the process more generalised.

2.4.3 Player models

When addressing Adaptivity Issue 3, regarding how to adapt when adaptation condition(s) are met, many of the examples explored in Section 2.3 utilise a player model. Charles et al. (2005) define a player model as a set of data which implies information about a single player. A review of player models utilised in video games made by Jagust, Boticki, and So (2018), Bioware (2012), Wang et al. (2017), Pelanek (2016), Bontchev and Georgieva (2018), Kojima (2015) and Capcom (2005), uncovered the following types of implied information that player models could contain:

Player Model Implied Information 1 (Players' proficiency at skills) *This refers to how good players are at using the skill(s) demanded by the game. For example an action game could demand combat and navigational skills from players. Some players might be more skilled at combat while others are not. The same variation could also apply to navigational skills.*

Player Model Implied Information 2 (Players' preferences of skills) *This refers to the idea that in games that demand more than one skill from players, it is possible for players to have preference in terms of the skill(s) they apply. Using the example above, it is possible for some players to prefer the application of navigation skills compared to combat skills and vice versa. Liu, Moon, and Kim (2020) explain that in some cases, players' preferred skills are the one(s) they are most proficient at.*

Player Model Implied Information 3 (Absence or presence of knowledge) *This refers to players being aware of or understanding the information the game wants them to. For example, in an educational game that also uses navigational challenges, it is possible for players to be skilled at navigation but lack understanding in the educational content, therefore stalling progress.*

Player Model Implied Information 4 (Probability estimates) *This refers to probability estimates of players performing actions and/or reaching points of progress. Using data captured from players' previous actions and progress, the probabilities of players reaching each subsequent point of progress can be estimated. For example a player that has needed multiple attempts and assistance in order to complete each combat encounter prior to the time of estimation, would probably be assigned a low probability of completing the subsequent encounters without the same amount if not more assistance.*

Player models could contain more types of implied information. However, for the purpose of adaptation to better fulfill games' primary purposes, the ones described above serve as a promising starting point. This is because the pieces of information explained above could potentially denote what sort of change could be beneficial when it is determined that change is beneficial. Therefore, using a player model with at least one of the implied pieces information above could allow a video game to calculate which game elements to adapt and the time at which to adapt them, to better suit the primary goal of the game.

To illustrate the effectiveness of player models at informing adaptations, Charles et al. (2005) proposed a high-level framework to implement adaptivity in online educational games. This framework uses a player model to capture a player's proficiency at skills, and preferences at using each skill, by monitoring that player during gameplay. Whenever an adaptation of the game is identified and performed, the framework measures its effectiveness, which can lead to either a new adaptation and/or an update of the player model. Similarly, Westra and Dignum (2010) developed a system to make the tasks within educational scenarios adaptive. This was accomplished by using a player model that estimated each player's proficiency at the skills needed to accomplish the different types of tasks presented. This model was continuously updated to accurately quantify player's skills and make the more effective adaptations.

Although the reviewed adaptive systems and their target game elements differ, there is common ground in their usage of an adaptable player model structure to quantify players' proficiency at the required skills, and preferences at using each skill to inform adaptations made. However, the skills required by a game can differ based on the genre, design and chosen educational content. The required skills of a video game are identified by examining the manner in which a video game challenges a player and identifying the skill(s) a player is required to apply to achieve completion. For example, in a video game that tasks a player to traverse a virtual world while defending him/herself against enemies using guns, relevant skills for a player model could include the player's proficiency at combat and navigation.

Calculating a player's proficiency at a skill required by a video game is done by processing a *player behaviour dataset* (PBD). Player behaviour data refers to data points within a video game that pertain to the skill(s) being measured by the player model. Taking the example stated earlier, data such as the number of times a player was hit, the number of shots fired by a player, the number of shots landed on enemies by a player, the number of enemies present and the number of enemies killed, pertain to a player's proficiency at the combat skill. Data such as the number of areas visited, total number of areas, obstacles overcome, total number of obstacles present, the time it took to complete the level and the average time taken to complete the level, pertain to a players proficiency at the navigation skill. A player model and PBD consisting of implied information and data regarding these skills could be expressed as collections of integers representing the data measured. This example is illustrated by Table 2.1 showing an example of PBD and Table 2.2 as an example player model.

Table 2.1: An example of a player behaviour data set.

Skill	**Data**	**Value**
Combat	Number of shots fired by player	40
	Number of shots landed on enemies	8
	Number of enemies killed	3
	Number of enemies encountered by player	10
	Number of time enemies attacked the player	13
	Number of times player was hit	11
Navigation	Number of areas visited by player	6
	Total number of areas	6
	Obstacles overcome by player	12
	Total number of obstacles encountered by the player	13
	Time it took to complete	12 minutes
	Average time taken to complete	15 minutes

Table 2.2: An example of a player model.

Skill	Data	Value
Combat	Accuracy of shooting	20%
	Proficiency at killing enemies	30%
	Dodging proficiency	15%
Navigation	Exploration proficiency	100%
	Proficiency at overcoming obstacles	92%
	Navigation speed proficiency	80%

In this example, drawing from the PBD, the player model can be processed. Regarding combat, this player had a shooting accuracy of 20%, killed 30% of the enemies presented and a dodging proficiency (the ratio of the number of times a player was not hit to the number of times that player was attacked) of 15%. This suggests that this player is lacking in combat proficiency. Whereas, this player visited a 100% of all areas present, overcame 92% of obstacles and had a navigation speed proficiency (the ratio of the speed of navigation of a player compared to the average) of 80%, indicating high proficiency at navigation. This model implies that this player is more proficient at navigational challenges as opposed to combat. Therefore, this player model could inform possible adaptations involving increasing the difficulty and frequency of navigational challenges and/or reducing the difficulty and frequency of combat challenges to boost the fun this player experiences.

Although player models can vary greatly based on the different required skills that are needed to complete video games, they are most often expressed in two ways. Firstly, researchers such as Clark et al. (2016), Lin et al. (2013), Sullivan, Mateas, and Wardrip (2010) and Chen and Lee (1999) express a player model as a data structure containing a mixture of numerical data pertaining to Player Model Implied Information 1 (a player's estimated proficiency at the required skills) and Player Model Implied Information 2 (alphanumerical data indicating a player's preference regarding the required skill(s)). Representation of a player's proficiency would take the form of a numerical value identifying where that player would rank on a predefined scale of performance. Table 2.2 indicates this. Representing a player's preference would include identifying that player's favourite skill to apply from the pool of required skills (further details in Chapter 3). This first type of player model will be referred to as a *performance player model*.

Conversely, Pelanek (2016), Gutierrez and Atkinson (2011), Garca et al. (2005) and Huang et al. (2008) express player models as data structures quantifying the probabilities of a player reaching all possible points of progress and/or performing actions within a video game (Player Model Implied Information 4). This could be expressed as directed acyclic graph (DAG) structure. The nodes represent all the possible points of progress a player could reach. Each edge is labeled by an action taken and the probability that the modeled player would take that action, when within the game's progress point denoted by the edge's starting node. This second type of player model will be referred to as a *probability player model*. Figure 2.2 (further below) serves as an example of what a probability player model could look like.

A player model for an educational game would comprise data pertaining to a player's ability to acquire, comprehend and apply educational content instead of data regarding his/her skills with weapons and/or navigation for example. This could take the form of either a performance player model or a probability player model. However, Cin and Baba (2008) explain that most of the existing adaptive educational systems struggle to learn from player behaviour. This is due to the various sources of uncertainty when ascertaining a player's ability at each of the three skills from data representing a player's behaviour within an educational game. The uncertainty stems from a player's in game actions not always directly indicating his/her mental state and/or ability, which may lead to the generation of an inaccurate player model. For example, a player who repeats the same actions to the point of redundancy, might be doing so due a lack of understanding on how to progress, wanting to elicit a different outcome, a lack of attention and/or frustration. An inaccurate player model may render an adaptive educational system ineffective due to wrongly assessing the appropriate level of challenge for a player and/or presenting a player with options that he/she does not find appealing.

There exist multiple techniques to generate and/or update player models. As these techniques vary in terms of effectiveness, advantages and disadvantages, the more commonly utilised techniques were delved into. These include *fuzzy logic*, *decision trees*, *Bayesian networks*, the *hidden Markov models*, *neural networks* and *genetic algorithms*.

2.4.3.1 Fuzzy logic

Fuzzy logic is a modeling technique developed to imitate human reasoning as explained by Zadeh (1965). Fuzzy logic can be described as an extension of the traditional set theory expressed as a measure of partial truths, lying in between absolute truth and absolute falsity. Partial truths consist of either statements or numerical values that fall within a range of possible statements or values. For example, partial truth statements regarding the proficiency of a player may be expressed as a range of statements including the player has no proficiency, below average proficiency, average proficiency, above average proficiency or is at maximum proficiency.

A fuzzy logic system (FLS) is the process of applying fuzzy logic to a concept with the aim of modeling it. An FLS comprises four components which include fuzzification, an inference engine, a rule base and defuzzification. The fuzzification stage involves taking in all possible inputs and identifying the subset of inputs that affect the chosen concept. This subset of inputs is then fuzzified and fed into the inference engine as fuzzy input. The inference engine calculates the value of a fuzzy output based on the fuzzy input and rules taken from the rule base. Magdalena (2015) explains that rules in the rule base could take the form of formulae that use the fuzzy input(s) to calculate the values of the fuzzy output(s) and/or conditional statements that denote the values of the fuzzy output(s) based on the ranges that fuzzy input values fall under. The rule base may be extracted from numerical data and/or supplied by experts. Lastly the defuzzification stage involves converting the fuzzy output from the inference engine into precise output representing the current state of the concept. These values of precise outputs would fall between a minimum and a maximum, with the minimum representing an absolutely false outcome and the maximum representing an absolutely true outcome. Figure 2.1 illustrates the structure and data flow of a typical FLS described above.

Almohammadi et al. (2017) explain that FLSs are capable of modelling a player's proficiency and/or preference towards skills within an e-learning platform, from interactions that player makes within the e-learning platform that are seemingly unrelated to said skills. The relationship between a player's interactions and inferences regarding his/her proficiency and/or preference towards skill(s) is denoted by the rule base and would vary based on the skill being modeled. They go on to explain that an FLS quantifies a player's proficiency and/or preference towards skill(s) as a numerical value. This would take the form of a number within a range, with the minimum value indicating a complete lack of proficiency/preference and the maximum value indicating that the player has mastery of/ preference towards that skill. The data generated from the platform and the player's interactions with said platform would serve as input. The rules base could be

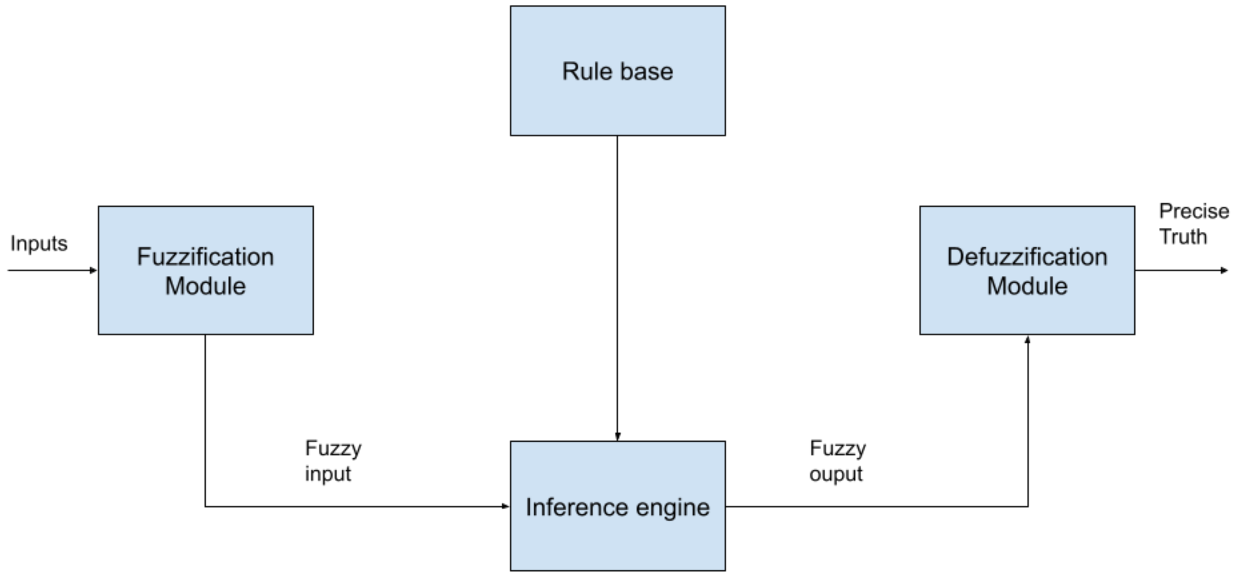

Figure 2.1: Block diagram of a typical fuzzy logic system

determined either by sourcing from experts at the skill in question, or examining data from similar research for possible patterns. The output would be the number indicating a player's proficiency and/or preference.

Thus, the output(s) of FLSs could be used to model a player's proficiency at the skills needed to learn using an e-learning platform. In short, the output(s) of an FLS would be expressed as a player model. A player model generated using an FLS would use a performance player model type as its data structure. An educational game could then make use of such a player model to adapt itself to a player. This could be done by comparing a player's proficiency at each of the required skill(s) needed by the game, and adapting the difficulty and/or frequency of the challenges where the player is meant to apply said skill(s). For example, a player that has been determined to struggle with navigational challenges could be adapted to by reducing the number of and/or reducing the difficulty of navigational challenges given to him/her.

Kavi et al. (2003) echo this by explaining that FLSs can be adopted to model players' preferences in regards to the levels and communication game elements. Hamdaoui, Idrissi, and Bennani (2021) demonstrated this via the creation of an educational game called Woodland. The game utilises an FLS that takes player behaviour data as input and a rule base sourced from experts to predict players preferred style of the presentation of educational content. They tested the effectiveness of their system by getting high school students to take questionnaires regarding their preference in communication styles, getting those same students to play their game, and comparing the output player model to the questionnaires. They explain that a great correlation was found between the player model results and the questionnaire results, indicating that their apporach to player modeling was accurate.

Similarly, Papadimitriou, Chrysafiadi, and Virvou (2019) developed an adaptive educational game named FuzzEG, which teaches the knowledge domain of HTML programming language. It utilises an FLS to estimate players' skill level at the programming language. This is done using player behaviour data and data regarding players' progression as input, and a rule base sourced from experts. By modelling players' skill levels, the number and difficulty of questions posed in quizzes within the game were adapted to each player. They evaluated their game via conducting play testing, and follow up interviews with the players. They noted a positive response from players and increased motivation to keep playing and learning.

2.4.3.2 Decision trees

A decision tree is a flowchart like structure comprising nodes and branches that is commonly used for modeling the states a concept can take and decision making. Each node represents a possible state that the concept being modeled can be in, and each branch represents the probability and/or conditions for each of the states occurring (Dunham, 2006). The probability/conditions for transitioning between states varies greatly depending on the concept that the decision tree is attempting to model. But, these can often be extrapolated from previously collected data and models, and/or supplied by experts in that field.

When considering applying a decision tree to video games, Adhatrao et al. (2013) explain that two methods exists, the probabilistic method and the condition method. Regarding the probabilistic method, each node would represent a state that the game was capable of taking and each branch would represent the actions that a player would have to take and/or the probability that a player would choose that transition. The decision tree would start with a root node where player actions normally begin. Next, a player would traverse from node to node based a combination of the node transitions presented and his/her personal choice. The decision tree ends when a player reaches an end node, which has no further connective branches. The choices and paths a player takes through the decision tree can then be used to update a decision tree's edge probabilities, denoting the most like choices and path that player could take next. A player model generated using the decision trees technique would therefore use a probability player model as its data structure.

Figure 2.2 represents a decision tree, using standard flowchart symbols, where nodes represent all the states a player can transition to within a game. Each branch describes the action a player would have to take to transition to the next node and an estimated probability of a player performing said action. This example goes through the actions required and probabilities of a player picking up a sword, going exploring and potentially killing a monster.

Regarding the condition method, decision trees using this method would be similar in structure to the probabilistic method but they will include decision points to see if conditions have been met. Each node would similarly represent a state that the game was capable of taking. At least one decision point would reside in between nodes to check if condition(s) have been met, which in turn determines the new node the player can traverse to. Branches represent binary true or false values depending on the conditions being fulfilled. The decision tree ends when a player reaches an end node, which has no further connective branches. The player's proficiency and preference towards the necessary skills can then be estimated from the conditions that have and have not been fulfilled. A player model generated using the decision trees technique would therefore use a performance player model as its data structure.

Although there also exists the possibility of making use of condition based decision trees for player modeling as opposed to probabilistic, the latter is reasoned to be the better option. This is due to condition based decision trees being limited by their conditions being statically fixed by the designer and/or learnt from player testing before the game is published. Whereas probabilistic trees have the potential to automatically learn about the player while in game due to its learning being based on evidence data observed from the player.

When applied to educational games, decision tree techniques have proven to be effective in ensuring that players are automatically presented with their preferred options to learn the intended content, leading to improved efficiency of learning. In their case, Lin et al. (2013) demonstrated this by building a base

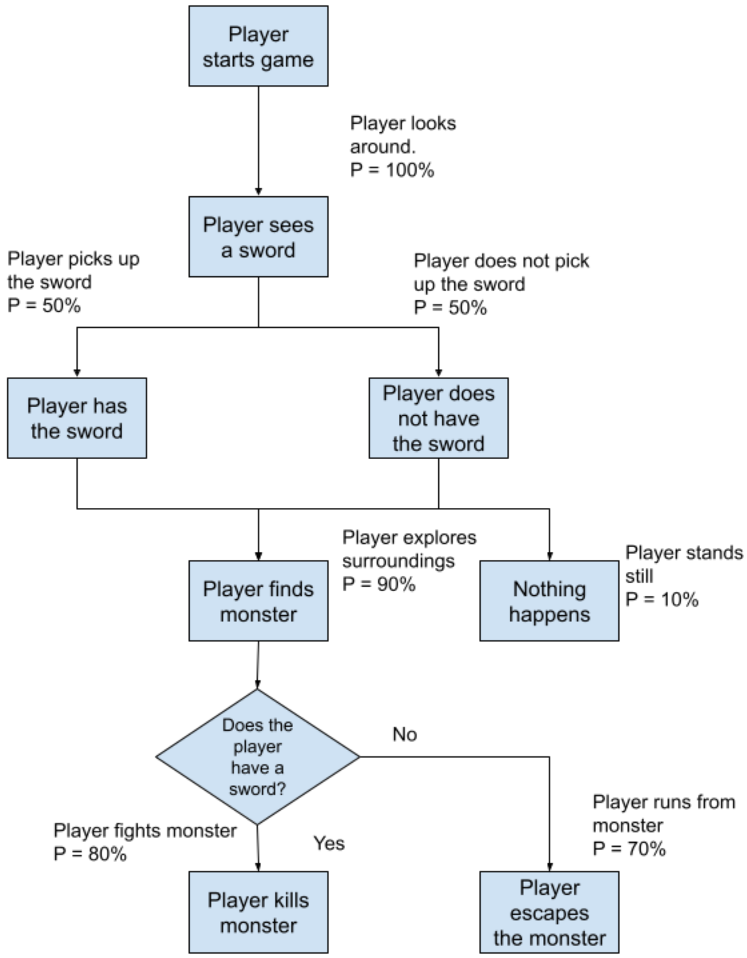

Figure 2.2: An example of a decision tree using flowchart symbols

learning environment with multiple options of content presentation, interviewing and profiling players to create incomplete player models, which were used to determine the optimal options presented to each player. The incomplete player models were created before hand to avoid the possible inaccuracy of player models generated from the beginning of a play session. The learning environment would also keep track of player choices and update the player model at pre-specified points of progress, thereby presenting content to players' using his/her preferred option of presentation. Baradwaj and Pal (2012) also developed an adaptive system meant to optimise learning. This method utilised a decision tree model based on pre-obtained player models to establish the optimal learning paths out of a collection of previously designed paths, which would then be presented to the student.

These two implementations of decision trees proved educationally effective but also suffered from the limitation of requiring player data before the playing of the educational game in order to ensure that the player models became accurate as quickly as possible. This could potentially make a player aware of adaptations being made. However, it is also possible to construct and start with decision trees containing arbitrary/guessed probabilities which will then be updated to more accurate values as player behaviour data provides evidence that probabilities are too high/low. But, this approach in theory could result in a longer amount of time needed for player models to become accurate.

2.4.3.3 Bayesian networks

Gonzalez, Burguillo, and Llamas (2006) explain that a Bayesian network (BN) is a DAG that shows and explains the distribution of probability of events taking place, or condition(s) being met in a way that allows for the modeling of behaviour in intelligent systems. Moreno et al. (2005) explain the BN as a collection of nodes and edges forming a DAG. Nodes represent the concepts being modelled. The concept(s) being modeled are affected by specific variables. Edges represent conditional dependencies between the nodes. Nodes that have no parents, i.e. nodes that are not connected to by edges, represent concepts that are conditionally independent of other concepts. The influence of a parent node on each child node is calculated by a conditional probability table that every node has. These conditional probability tables comprise the probability of the affecting variables of a node and all its parents being true. This in turn can be used to calculate the probability of the node being true. In circumstances where there is no parent node present, the conditional probability table of a node consists of only its own affecting variables.

Almohammadi et al. (2017) detail how a BN could be applied to educational games to calculate the probability that a player will complete the objectives within that game. These objectives are represented by the nodes of the BN. The affecting variables include the actions and choices a player makes. A player model processed using a BN would take the form of a probability player model. However, the actions and choices made are represented in the nodes' conditional probability tables, as opposed to the edges of a DAG. Thus, the probabilities of players completing different objectives can be read from a player model. This could then inform possible adaptations of a learning environment. For example, if it is detected that a player has a low probability of completing his/her current objective, more motivators could be offered.

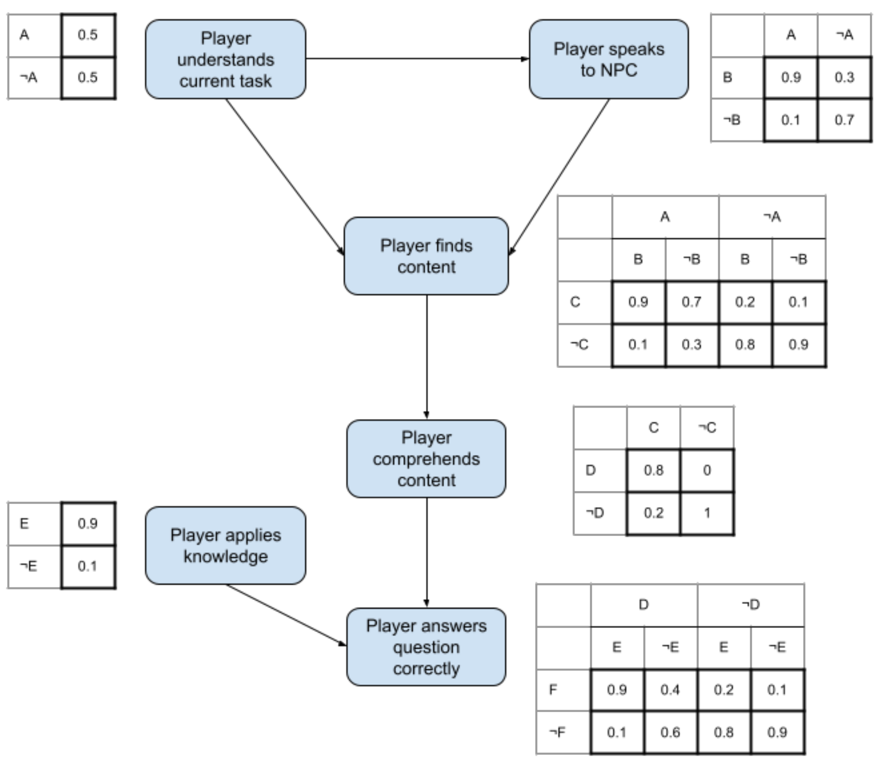

Figure 2.3: An example of a Bayesian network

Figure 2.3 illustrates an example of a BN used to calculate the probabilities of a player accomplishing a string of objectives leading up to answering a question correctly. The nodes include the player understanding his/her current task (A), speaking to an NPC for information (B), the player finding the educational content (C), the player comprehending the educational content (D), the player applying all the knowledge he/she has (E), and the player answering a question correctly (F). Each node possesses a probability table. From the figure, node A directly affects nodes B and C, node C influences D, and the final node F is dependent on nodes E and D.

Creating a player model by using a Bayesian network is considered to be problematic due to the number of affecting variables and conditional dependencies involved with calculating the probabilities of a player completing objective(s) (Ahmad, Basir, and Hassanein, 2004). Having a high number of affecting variables and conditional dependencies would indicate a larger and more complex DAG and conditional probability tables for all nodes. A lower number of variables could be used, however Ahmad, Basir, and Hassanein (ibid.) note that doing so may lead to inaccurate results. As adaptations made by an educational game are informed by a player model, an inaccurate player model could lead to adaptations that are ineffective in aiding a player's learning.

Examples of BNs being used to process player models for adaptivity include Gertner and VanLehno (2000) and Garca et al. (2005) who created the ANDES and OLES adaptive tutoring systems respectively. Both systems succeeded in processing player models, without the need for prior interviews, which adapted the language used and presentation of the desired educational content to players' preferences. A tutoring system differs from an educational game due to functioning like an electronic textbook presenting educational content without the interactivity found in educational games such as game-play, a virtual world, characters, objectives and narratives aimed at entertaining a player.

A more recent example of a BN being utilised to generate player models in an educational context is the work of Ferreira et al. (2016). They developed a hybrid modelling system to predict students' final knowledge level and grade for a university level programming course. They utilised ontologies to describe 20 students' behaviours which served an input to a BN that predicted the probabilities of those students possessing knowledge of each of the 23 topics of the course, along with their final grade (the player model). The BN's final player model outputs were compared to the actual progression and final grades of the same twenty students. The comparison yielded a correlation between the real knowledge level and the predicted values generated by the BN. Although this system was not neither paired with an educational game nor used for any adaptation, Ferreira et al. (ibid.) claim that their system outputs accurate player models that could inform adpativity and content personalisation in e-learning systems.

2.4.3.4 Hidden Markov model

Shih, Koedinger, and Scheines (2010) explain that a hidden Markov model (HMM) is a method of modeling the behavioural patterns of people and/or systems. The process involves defining a collection of unobservable discrete states that a system is capable of taking, with a start state and at least one end state. The current state of a system is assumed based on the observation of specific affecting variables and condition(s). States are connected to one another via a probability matrix. A probability matrix contains the probabilities of observing all the possible values for each of the affecting variables of the states. Probability matrices can therefore be used to determine the probability of transitioning from one state to another. Thus, examining probability matrices can serve as a prediction for the most likely transition(s).

The order of a Markov model is defined as the number of affecting variables contained in the model's probability matrices. Azough (2010) details that low-order Markov models suffer inaccurate predictions due to not accounting for all the relevant variables. On the other hand, the probability matrices of high-order Markov models require more calculation than low-order HMMs due the inclusion of more affecting variables, which entails a more complex calculation of the probabilities of each possible transition.

By applying the high-order HMM behaviour prediction model to students, Huang et al. (2008) explain that it is possible to create a player model to inform the adaptation of the learning environment. The states of the HMM would represent a player's changing knowledge states. When considering an educational scenario where a player is tasked to seek information needed to solve a problem, examples of a player's knowledge states could include: "the player understands that he/she needs to seek information", "the player has found and understood the information", "the player understands how to apply said information" and "the player has applied the information and solved the problem". The affecting variables of such a prediction model comprises the actions that a player performs within the learning environment. Therefore, observation of a player's actions indicate his/her current knowledge state. A player model generated using this method would be a probability player model structure.

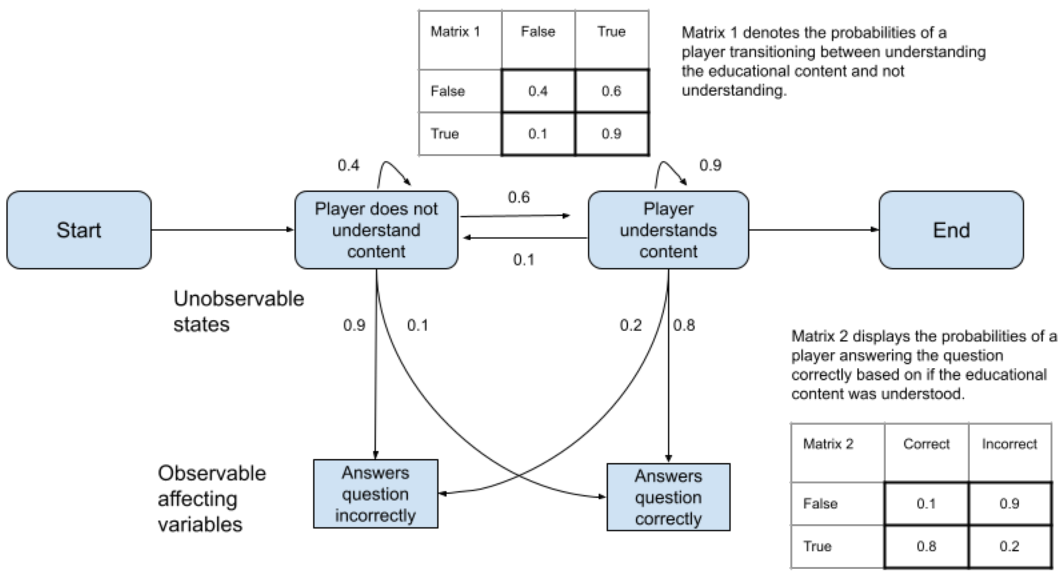

Figure 2.4: An example of a hidden Markov model

Figure 2.4 displays a simple example of an HMM modelling a player's knowledge space. In this example, the unobserved states are defined as a player not understanding the educational content and the player understating the educational content. The observable affecting variable is the result of the player's answer to a question pertaining to the educational content. Matrix 2 details the probability of the player giving the right answer, 80% if the player understands the content and 10% if the player does not. Matrix 1 denotes the probability of the players knowledge space changing. If the player does not understand the content, there is a 60% chance that the explanations given will change that. If the player is assumed to understand the content, there is a 10% chance of this assumption being false. One possible way this player model could inform adaptation could be to offer motivators and/or a different presentation of the educational content if it is surmised that a player does not understand the educational content.

However, the example above is very simple. Modelling a player's knowledge space within an educational game using an HHM is reasoned to be far more complex. This due to the many possible player knowledge states, affecting variables attributed with the intended educational content, and the player's understanding of game rules and mechanics. Huang et al. (2008) successfully utilised an HHM to generate accurate player models that they paired with an it intelligent tutoring system to make the frequency and type of assistance given to player adaptive. Their observable states consisted of different extents of understanding for each of the topics of the educational content. They utilised the percentage of correctly answered questions and the amount of time it took to complete each quiz as observable affecting variables, to indicate the state that players were in.

A more recent example includes the works of Tadayon and Pottie (2020) who successfully used an HMM to predict players' performance in an educational game. Their observable states comprised different levels of mastery for each of the topics of the educational content. They utilised the total number of attempts per level, total number of moves made by a player per level, and the amount of time it took to complete each level as observable affecting variables to indicate the state that players were in. Although their results proved accurate, the player models generated were not utilised to inform adaptations. However they go on to speculate that their implementation could serve as a strong foundation to inform adaptations in the future.

2.4.3.5 Neural networks

A neural network is a series of algorithms aimed at recognising underlying relationships in a set of data through a process that mimics the way the human brain operates (Frias-Martinez et al., 2004). They also explain that neural networks can be used to model and predict human actions and responses. This process involves combining a large number of algorithms (neurons) to work together to process information in tandem. In effect, it is a system which takes in and analyses information akin to the biological nervous systems found in a brain, hence the name. Although Seldon (2022) identifies multiple types of neural networks (feedforward, perceptron, radial basis, recurrent and modular), when considering the context of educational settings and/or games, the works of Tahmasebi and Hezarkhani (2011), Min (2020), Idris, Yusof, and Saad (2009) and Ibrahim and Rusli (2007) indicate that the feedforward neural network is most commonly used. Hence the feedforward network is further delved into.

In a feedforward network, neurons are organised into the input, output and hidden layers (Ciresan et al., 2011). The input layer receives the external data. The output layer produces the desired result. The hidden layer sits between the other two and carries out intermediary processes and/or tasks needed to arrive at the final result. Hidden layers may comprise multiple layers depending on the complexity of the problem. Single layer and unlayered networks structures also exist and may be used. Between two layers, neurons are connected to each other, allowing the output of some neurons to become the input of others. Multiple connection patterns between layers are possible. They can be fully connected, with every neuron in one layer connecting to every neuron in the next layer. They can be pooling, where a group of neurons in one layer connect to a single neuron in the next layer, thereby reducing the number of neurons in that layer. Regardless of the number of layers and the connection patterns, neural networks take the form of directed graphs.

Drigas, Argyri, and Vrettaros (2009) say that neural networks may be applied towards educational games by classifying, characterising and tracking players' progress. This due to their ability to recognise patterns and extract meaning from imprecise or indirectly linked sources of data. For example inputs such as a player's in game interactions and movements could be used to model that player's understanding of their objective. Interactions and movements that have no clear pattern could point to a player not understanding the educational content and/or what to do next.

Neural networks are able to model a player's progress and potential issues he/she might face in a game. Drigas, Argyri, and Vrettaros (2009) explain the process as identifying possible player interactions and data for the input layer, objectives and/or points of progress for the output layer, and linking the inputs and outputs with a probabilistic algorithms in the hidden layer to estimate the probabilities of players accomplishing said objectives and/or points of progress. Hence, a neural network itself would serve as a player model using the form of a probability player model structure. Furthermore, according to Beetham and Sharpe (2013), as the possible interactions and objectives within in each educational game is dependent on developer choice, developers are required to define all possible player actions, points of progress, gameplay based problems and pedagogical problems manually when applying neural networks.

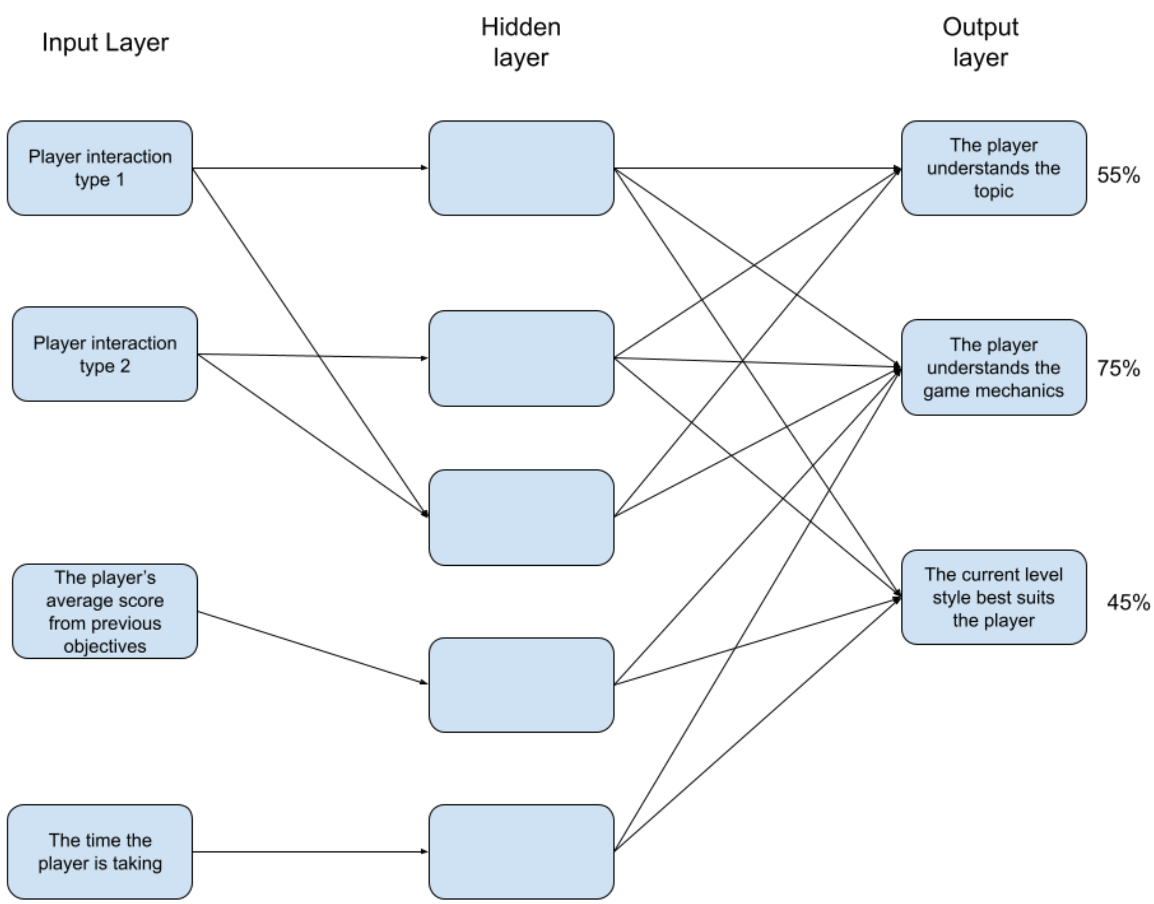

Figure 2.5: An example of a neural network

Figure 2.5 shows a simple neural network that could be used as a player model. This example assumes that there is one educational topic, two types of player interactions and more than one level style to challenge the player. The input layer includes the player's type 1 interactions (input 1), the player's type 2 interactions (input 2), the player's average score for previous objectives (input 3) and the time the player is taking (input

4). The output layer comprises the probability that the player understands the educational content (output 1), the probability that the player understands the game's mechanics (output 2) and the probability that the current type of learning objective best suits the player (output 3). The hidden layer contains four intermediary neurons to process the outputs. These inputs would then be used by the hidden layer nodes to calculate the probability of each of the three outputs being true.

When examining neural networks applied to an educational setting, Fenza, Orciuoli, and Sampson (2017) paired a neural network with an intelligent tutoring system to adapt the order of educational topics presented to players from easiest to most difficult, and to adapt the difficulty of questions asked for each topic. This was accomplished by running a data gathering session where players interacted with teachers. The teachers' opinions were then used to form estimates of each player's knowledge of each topic to generate the initial order of topics to be presented to him/her. From there, the intelligent tutoring system would also monitor the number of questions answered correctly, the number of motivators given for each question and the time it took to answer each question for each topic, to be used as input for the neural network. This was used by the neural network to form new estimates of players' knowledge of each topic and adapt the order of topics if need be.

When being applied to education games, Min (2020) successfully paired a neural network with an educational game he developed, called ENGAGE, in order to model players competency at the educational content. His, approach involved taking a mixture of players' past classroom based performances, grades, and a content based self evaluation survey that each player completed. This served as a reference to compare their system's output to. The game game monitored the number and order of 19 possible interactions within the game that players could make, and the number of attempts players took to complete each task as input to estimate players' competency level. Although the educational game was neither adaptive nor used the generated player models to inform adaptations, he showcases evidence that neural networks possess the capability to accurately model players.

2.4.3.6 Genetic algorithms

Sivanandam (2013) describe a genetic algorithm as search process for a solution akin to that of natural selection in an evolving pool of possible solutions. Each solution is given a fitness rating based on heuristics that are relevant to the target problem. This process enables new solutions to be generated in the pool using new information regarding the problem as it is introduced. Thus, through a continuous supply of new information, generation and searching, solutions may become more optimised and/or better solutions may be found. Evolution in the pool of solutions is carried out using biologically inspired operators such as mutation, crossover and selection (Ronco and Benini, 2013). They explain that the mutation operator is used to generate new solutions with altered aspects from the previous ones so that diversity is maintained. This is so that solutions in the pool do not evolve to be too similar to one another, thereby making the algorithm's results redundant. The crossover operator (also called recombination) is used to combine aspects of two current solutions (parents) to generate new one (offspring). The selection operator is the mechanism that selects solutions for crossovers and mutations. This is done by normalising and accumulating the fitness rating of each of the solutions, randomly generating an R value between 0 and 1, and selecting the solutions whose accumulated normalised fitness rating is greater than or equal to R.

Drigas, Argyri, and Vrettaros (2009) explain that genetic algorithms can be applied to educational games by identifying players' preferences, strengths and weaknesses concerning adaptable components of educational games. This is done by identifying the different possible ways that adaptable component(s)

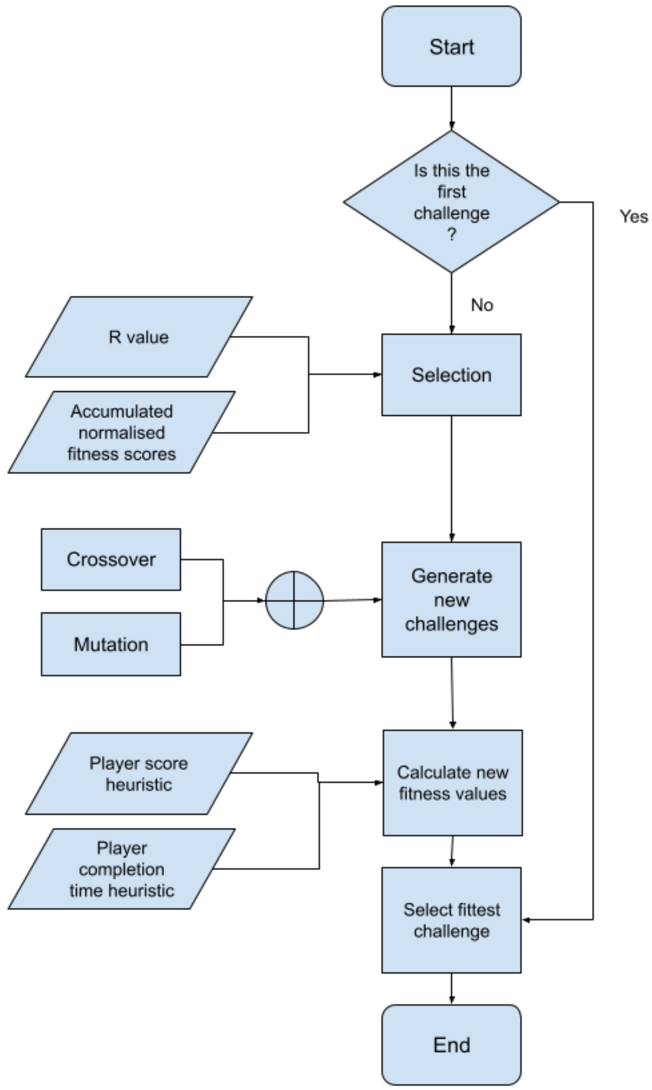

Figure 2.6: An example of a genetic algorithm using flowchart symbols

can evolve (outcomes), identifying relevant heuristics pertaining to players' behaviour, preferences, strengths and weaknesses concerning said adaptable component(s), and using those heuristics to calculate the fitness values of each outcome. Players' preferences in regards to adaptable component(s) would be denoted by the outcome(s) of each adaptable component with the highest fitness ratings. Player models generated using genetic algorithms would take the form of a performance player model.

Figure 2.6 displays an example of a genetic algorithm geared to finding players' preferred type level style (the way in which a game challenges players). For this example, let us assume that there are two outcomes that the level styles evolve can evolve towards, combat and navigation. Thus each new outcome generated could be more geared towards combat, navigation or a mix of the two depending on the operator used for generation. The algorithm is called every time a new level style needs to be selected (when a challenge is to be presented to players). First it is checked if it is the player's first challenge. If so, the fittest level style is chosen. If not, the selection process begins to generate more challenges either by crossover or mutation.

Thus, new challenges are added to the pool with each iteration of the algorithm. The player's preferred type(s) of challenge will be denoted by the fitness scores for each objective.

Badr and Mohammed (2019) showcase the use of a genetic algorithm for player modelling to inform adaptation in an educational game. Their system aims to identify educational topics that players are strong and weak at. Adaptation takes the form of increasing the number of questions (in presented quizzes), and providing assistance for topics that players are weak at and vice versa. This was done by classifying all educational topics into groups (akin to text book chapters they might belong to). Getting players to take an evaluation before playing the game and using players' score to determine the initial fitness values for each group. Quizzes were then generated with either greater or fewer questions, and differing amounts of assistance based on the fitness values of the educational topic being tested, while players played the game.

2.4.4 Methods for player monitoring

When considering Adaptivity Issue 4, which pertains to monitoring for and/or collecting the relevant data to determine when adaptation condition(s) are met, all the examples explored in Section 2.3 utilise a form of player monitoring. Furthermore, since monitoring a player's behaviour within a video game is the only method of capturing player behaviour data, which is one of the methods of generating a player model, methods of player monitoring are explored below. Yannakakis (2012) conducted a review of literature applied to player monitoring for the purpose of creating player models and classified it into two distinct areas:

Player Experience Modelling (PEM). PEM involves monitoring behaviour to infer a player's experience in real time to predict the best adaptation to fulfill the primary goal of the game.

Massive-Scale Game Data Mining (MDM). MDM involves the analysis of statistical player data after a player has completed a play session, such as the number of people playing the game and the number of people who have performed certain actions in the game, to determine trends of player preference. Trends of player preference could refer to an individual player's preference changing over time as they play the game, or the overall player population's preference changing.

The monitoring of player behaviour creates a player behaviour data set. When comparing the two techniques above, PEM is capable of capturing a PBD during a play session as opposed to MDM which requires the play session to be completed. As explained in the previous section, the PBD is processed into the player model, and any adaptations made are informed by the player model. Therefore, adaptations would be dependent on having a player model, which would be dependent on having player behaviour data captured.

As the purpose of this study is to make educational games more adaptive to better teach players, it was reasoned that allowing a player to experience adaptivity in a game as soon as possible should be a priority. The reasoning for this stems from the premise that the quicker adaptations are made, the quicker the different options available can be tried on players. This could allow the more effective option(s) to be identified sooner and presented to players to potentially boost their learning gains.

As MDM requires a player to first complete a game to capture his/her PBD, his/her player model can only be processed after the game is completed. This translates to a lack of adaptivity during a player's first play-through. However, as PEM monitors a player's behaviour in real-time, it also captures a PBD in real-time. Building on this, the player model can also be processed in real-time and adaptations can be made in the game during a player's first play-through.

Due to the greater applicability PEM has to the research question of this study, focus shifts to reviewing the state of the art for this mechanism. Cowley and Charles (2016) explain that there exist three different ways to monitor a player's behaviour using PEM:

Subjective PEM. This is identified as information about the player given by the players themselves. This can be done by either forced data collection using questionnaires or third party observation of the players.

Objective PEM. This refers to information deduced about the player by way of facial expression, speech patterns, posture and level of focus. It takes the string of emotions that a player would experience during game-play into account. For example, Asteriadis, Karpouzis, and Kollias (2008) describe a player who is leaning forward while playing a game as focused and engrossed. However this method can be intrusive at times.

Gameplay PEM. This refers to behavioural data obtained through the interaction between the player and the game. As Conati (2002) explains, video games are able to affect a player's cognitive state which then dictate their preferences and actions. For example, repetition of redundant actions could spell that the player is having difficulty in the game. Hastings, Guha, and K. Stanley (2009) identify gameplay-based PEM as the least computationally costly and least intrusive approach.

As the basis of PEM stems from monitoring a player's interactions with a video game, the types of possible interactions were explored. According to Salen and Zimmerman (2004), there exist four possible types of interactions between a player and a video game:

Cognitive. Refers to the player's psychological, emotional and intellectual state of mind while interacting with the game.

Functional. Includes the player's understanding of the game's mechanics and what is expected of him/her.

Explicit. Pertains to the player's literal interactions, such as pressing a button and moving a joystick, which translates to performing actions and tasks within the game.

Cultural. Encompasses the interactions involving aspects regarding the game that are not formally part of the game such as fan sites, creation of and use of cheats etc.

Each of the types of interactions serve to inform at least one of the types of PEM. Cognitive interactions inform objective PEM due to the creation of player models being based on a player's perceived emotional and psychological state. Functional and cultural interactions inform subjective PEM due to the generation of player models being reliant on information that players provide themselves and the monitoring of behaviour of players who are engaging in aspects regarding the game that are not formally part of the game. Explicit interactions inform game-play based PEM as its basis of generating player models is the monitoring of a player's literal interaction within a video game.

2.5 Summary

This chapter identified five types of pedagogy and how they are employed by educational games. Educational games a whole were also researched which explored three possible teaching mechanisms and creating entertainment value with the use of ludology and narratology. Multiple techniques used to make both entertainment and educational games adaptive were also reviewed for the purpose of extracting a framework for adaptivity in games. The framework includes the addressing of the four adaptivity issues of identifying adaptable components in games, determining the conditions for adaptation, identifying which components should be adapted when each condition is met and monitoring for when said conditions are met. The first issue was addressed with the identification of the 12 game elements in the GES. The second issue was remedied with the derivation of adaption conditions (see Chapter 3). The third issue was addressed with the use of a player model that is processed from monitoring players' behaviour within games. Six techniques that could be used to process players models were explored, along with practical examples of them in use. The fourth issue is solved with player behaviour monitoring techniques, with PEM being the more promising method to monitor a player's behaviour, due to being able to capture a PBD during a player's first play-through.

The next chapter focuses on the design specification of an algorithm that addresses all four of the Adaptivity Issues.

Chapter 3

Algorithm design specification

As explained in Section 2.4, there exist four Adaptivity Issues that need to be addressed in order to make a game adaptable. These include the issues of what to adapt (Adaptivity Issue 1), when to adapt (Adaptivity Issue 2), how to adapt (Adaptivity Issue 3) and how to collect the necessary data (Adaptivity Issue 4). As the purpose of this study was to make educational games adaptable to better improve learning in their players, each of the issues would need to be addressed to accomplish this purpose. This chapter focuses on the design specification of an algorithm that addresses all four of the Adaptivity Issues. Thus this chapter is split into four sections. The first addresses the issue of what to adapt by modeling the elements that are salient and whose adaptation may be of benefit to the education process in educational games. The second addresses the issue of data collection by comparing the advantages and disadvantages of the different techniques of player monitoring, and selecting a suitable technique for the purpose of this study. The third tackles the issue of when to adapt by identifying Adaptation Conditions, which are instances within an educational game where adaptation may be of benefit to players' learning. The fourth aims to address the issue of how to adapt by selecting and detailing the most suitable method of player modelling technique to be utilised by the algorithm.

3.1 Model for adaptable elements

Concerning Adaptivity Issue 1, the modeling of adaptable elements that could be of benefit to learning was split into four sub sections. The first maps the video games elements identified in Section 2.4.1 to the educational scenario stages identified in Section 2.2.1.2, in order to distinguish the elements that affect the educational process. The second identifies adaptability requirements which are used to filter the elements whose adaptability could benefit a player's learning. The third is an expression of the adaptable elements as a formal model across multiple dimensions. The fourth is a survey identifying the options for adaptability for each of the recommended adaptable elements where applicable.

3.1.1 Education process elements

As stated previously, educational games have primary and secondary purposes. The primary purpose of an educational game is to teach players the intended educational content. The secondary purpose of an educational game is to entertain its player as a source of motivation to keep playing and learning.

Section 2.2.1.2 identified and justified the benefits of educational games utilising educational scenarios as their teaching mechanism. A survey of the works of Huo (2019), Ibrahim and Jaafar (2009), Hullett and Mateas (2009) and Niehaus and Riedl (2009) allowed educational scenarios to implicitly be split into four stages. To reiterate, the four stages of educational scenarios consists of task introduction, content acquisition, content comprehension and content application. A player would have to complete all four stages within an educational scenario before it can be assumed that he/she has learned the educational content. Thus, educational scenarios can be considered to put player's through an educational process. Considering the GES explained in Section 2.4.1, each element is reasoned to predominantly affect at least one stage of the education process. The mapping of each element of the GES to the four educational scenario stages are explained below.

Educational Scenario Stage 1 (task introduction) is affected by the tasks, motivators and feedback elements. This is because these elements contribute to introducing the player to the educational scenario and explaining the task to him/her. The tasks element refers to the final objective and/or problem that a player would have to accomplish and/or resolve in order to complete an educational scenario. Thus, it sets the scene to introduce the player to start searching for educational content. The motivators element provides support to the player to help him/her better understand his/her task if he/she was unclear. The feedback element makes the player aware of the effects of his/her interactions within the educational game as he/she is introduced to the educational scenario.

Educational Scenario Stage 2 (content acquisition) is affected by the information, communication, motivators and feedback elements. This is because each of these elements plays a part in defining the educational content, how it is presented to the player and aiding the player with acquiring the educational content. The information element refers to the educational content the player is expected to acquire through playing the educational game. The communication element is embodied by the wording and presentation of the intended educational content. The motivators element aids the player with finding the educational content. The feedback element makes the player aware of the effects of his/her interactions within the educational game as he/she acquires the educational content.

Educational Scenario Stage 3 (content comprehension) is affected by the motivators and feedback elements. Each of these elements plays a part in aiding the player to better understand the educational content he/she has just acquired. The motivators element provides support to the player to help him/her better understand the full extent of the educational content and the way(s) to progress the educational scenario. The purpose of the feedback element is to make the player aware of the effects of his/her interactions within the educational game while he/she attempts to comprehend the educational content.

Educational Scenario Stage 4 (content application) is affected by the level styles, level difficulty, time pressure, motivators and feedback elements. This is due to these elements presenting a challenge to the player to apply the educational content he/she has comprehended, and the assistance he/she may require to do so. The level styles element pertains to the type of challenges(s) a player may face while attempting to complete his/her task. Thus, each task would contain at least one challenge which the player must apply the knowledge to overcome before the task can be considered complete. The level difficulty element refers to the severity of the difficulty that the challenge presents to a player. The time pressure element burdens the player to overcome the challenges presented and complete the task within a set time limit. The motivators element provides support to players experiencing difficulty in applying the educational content. The feedback element makes the player aware of the effects of his/her interactions within the educational game as he/she applies the educational content.

The other elements, narrative, environment, characters and rules, provide entertainment but are not directly part of the education process.

We believe that the GES is a good starting point for a set of elements to include in the design of educational games. However, other game designers could uncover more elements crucial to the education process and/or entertainment value which can then be added to the GES.

3.1.2 Identifying adaptable elements for educational games

In the space of educational games, adaptability is implemented by the manipulation of educational game elements to better teach the player. This section focuses on identifying educational game elements whose adaptability could benefit a player's learning. This would be accomplished by comparing educational game elements to adaptability requirements. The following four adaptability requirements were identified:

Adaptability Requirement 1 (Player learning) *Adaptations of an element must affect the educational process of the game so as to improve player learning.*

Adaptability Requirement 2 (Educational content) *Adaptations must not affect or alter the intended educational content of an educational game.*

Adaptability Requirement 3 (Abstraction) *Adaptations must not break abstraction so that players can continue to feel engaged and motivated to keep progressing.*

Adaptability Requirement 4 (Discreet adaptations) *Adaptations must be possible without the player being aware of the adaptation.*

Adaptability Requirements 1 and 2 were derived directly from the research question put forward in Section 1.1. Adaptability Requirement 3 was taken from Bateman and Boon (2005) and explained in Section 2.2.3. Adaptability Requirement 4 was taken from Chanel et al. (2011) and fully detailed in Section 2.3.1.

The application of these requirements to the 12 elements of the GES, identified the elements whose adaptation would violate at least one requirement. Thus, the elements that could be adapted to better educate the player were filtered. The results are as follows:

Violation of Requirement 1. As the narrative, environment, characters and rules elements affect the game's entertainment value rather than the educational process, as explained in Section 3.1.1 above, they should not be adapted.

Violation of Requirement 2. As altering the tasks or information elements would change the learning objectives or content taught, they should not be adapted.

Violation of Requirement 3. As adapting the rules element in a game would change what a player was previously allowed and not allowed to do, it should not be adapted.

Violation of Requirement 4. As adapting the characters and environment elements during a play session would involve making noticeable changes to the appearance and/or behaviours of NPCs and the game world, neither element should be adapted.

Any element that fulfills all Adaptability Requirements belongs to the *adaptable element set* (AES). The adaptability of any element in the AES is reasoned to have the potential to enrich a player's learning. From this list, the level style, level difficulty, communication, motivators, feedback and time pressure elements fulfilled all requirements and therefore comprise the AES.

This specific instance of the AES containing the six elements filtered above will be referred to as the *recommended adaptable element set* (RAES). The RAES refers to those game elements whose adaptation was identified as most suitable based on the existing research explaining the valuable ways to decompose an educational game. Table 3.1 summarises the elements of the RAES and the effects of their adaptations.

Table 3.1: The elements of the RAES and the effects of their adaptation.

Element	**Effect of adaptation**
Level style	Alters the way in which player's are challenged to apply educational content.
Level difficulty	Alters the severity of the challenges to apply educational content.
Communication	Alters the wording and presentation of information and educational content.
Motivators	Alters the type of assistance and motivation that is offered to players.
Feedback	Alters the type of mechanism used to make players aware of their actions.
Time pressure	Alter the amount of time allotted to players to complete tasks by.

Moving forward, the research contributions in this thesis aim to apply to any element in any AES, but are illustrated by application to the elements in the RAES.

3.1.3 Modelling adaptable game elements

As each element in the AES have different purposes and roles in educational games, they would need to be expressed as a model to distinguish between their differing modes of representation. These elements would be represented across three dimensions. The first is the kind of the element, preference or performance. Preference elements are realised as an unordered set of options that fit differences in players' preferred way(s) of learning, while performance elements take the form of ordered scales that determine different player abilities. The second is the type of adaptivity the element falls under, active or passive. The third is the time that the element should be adapted, at any point during or at predefined points in an educational game.

3.1.3.1 Kind of the element

Building on the work of Westra and Dignum (2010), adaptations in educational games could be of benefit to learning when taking players' individual proficiency at the educational game and content into account. It is reasoned that this would allow for personalised adaptations to aid the learning of players' regardless of differing levels of proficiency. To reiterate the works of Liu, Moon, and Kim (2020), two types of player proficiencies that can be modelled were identified, preference and performance.

The implementations of Bontchev and Georgieva (2018) and Shabihi and Taghiyareh (2017) represent preference elements as unordered sets of options that those elements could employ to teach a player. They reason that players would have differing degrees of preference at learning educational content using each of the available options. Degrees of preference at learning educational content is attributed to the completion of educational scenarios. This is because by definition, an educational scenario can only be completed upon a player going through all four stages of the education process explained in Section 3.1. Therefore, it is theorised that preference at learning using each option of each element would differ from player to player. The method of calculating an option's degree of preference is explained in Section 3.4.1.1. Elements such as level style, communication, motivation and feedback are preference based elements and are thus represented by sets of options. The adaptation of a preference element involves changing the option that is presented to the player to learn with.

According to Jagust, Boticki, and So (2018), performance elements pertain to the severity of challenge posed by educational games in order to teach players. They reason that different players might learn better under different degrees of severity of challenge. They go on to explain that the severity of challenge an educational game poses is tied to *performance metrics*. A performance metric is defined as a measurable aspect of a player's behavior, such as the number of interactions a player could take to complete a scenario. Using the implementations of Sandberg, Maris, and Hoogendoorn (2014), Clark et al. (2016) and Jagust, Boticki, and So (2018) as a basis, performance elements can be represented by two numerical values. The first is a *baseline*. A baseline is defined as the measure of the performance metric that has to be crossed before assistance is offered to a player during a scenario, and remains static. One way of determining a baseline is by taking the average of that performance metric recorded from player testing. The second is a *completion multiplier* (CM), which is variable and quantifies a comparison between a specific player's performance metric and the baseline of that same metric for a scenario. The completion multiplier is used to alter the frequency of assistance that the player is offered. An adaptive educational game would produce different completion multipliers and different amounts of assistance for different players. Conversely, a non adaptive game would have a constant completion multiplier value of 1.0, meaning that assistance is offered to all players at identical intervals. Elements such as level difficulty and time pressure are performance elements. The level difficulty element is tied to the performance metric of the number of player interactions taken to complete a scenario. The time pressure element is tied to the performance metric of the amount of time a player takes to complete a scenario. The adaptation of a performance element involves the recalculations of the CM, thus changing the frequency of assistance to best suit a player's learning abilities.

As the purpose of modelling both of these kinds of elements is to inform adaptations that could aid the learning of players, it was reasoned that including both kinds of elements in player models could allow a greater degree of adaptivity, which in turn could be of greater benefit to learning when compared to utilising only one. Thus is it theorised that at least one of each kind of element should be made adaptive in any implementation.

3.1.3.2 Type of adaptivity

Building on the works of Csikszentmihalyi (1990) and Koster (2004), every element in the AES can be informatively classified into active or passive adaptivity, and then further classified into the sub types of active or passive adpativity. Active adaptivity involves the adaptation of aspects in video games that the player is expected to engage with and/or execute his/herself. Passive adaptivity involves adaptation of the support systems in video games that aid the player with performing the tasks that he/she are expected to.

Considering the RAES, elements representing active adaptivity include the level style, level difficulty, communication and time pressure. The level style element represents interaction style adaptivity as it directly affects the manner in which the player is challenged to apply the educational content. The level difficulty and time pressure elements represent ability adaptivity as they alter the amount of challenge provided by the level style element. The communication element represents learning adaptivity as it revolves around altering the method of presenting educational content and/or instructions in an educational game.

The elements representing passive adaptivity in the RAES are the motivators and feedback elements. The motivators and feedback elements cover motivation, meta-cognitive and meta-reflective adaptivity as they provide hints to players that may be experiencing difficulty, make the player consider the effects of his/her own interactions, and can be used to encourage a player to consider different points of view and/or ways of thinking.

As stated previously, the purpose of combining active and passive adaptivity is so that a player can be educated and entertained simultaneously, thus being motivated to keep learning.

3.1.3.3 Time of adaptation

It was also reasoned that elements within the AES can be classified based on when they should be adapted. Adaptations in theory could take place at any point during a scenario or at specific predefined points during a scenario.

Performance elements can only be adapted at predefined points in an educational scenario. The player needs to be assessed in terms of his/her player metric(s) upon reaching a specific point and compared to the baseline for that point, which the game designer is required to set. Hence a player would need to reach that point in order to supply the needed player metric(s) data before any adaptation can occur. Enacting adaptation prior to this point would produce incorrect results due to lacking a frame of reference in terms of player metric(s). Thus, the level difficulty and time pressure elements should only be adapted at predefined points during a scenario.

Conversely the adaptation of the level style, communication, motivators and feedback elements involves changing the current option in use, and thus can be adapted at any point during an educational game.

Table 3.2 represents a model of the elements within the RAES, indicating each of their categorisations for all three dimensions.

Table 3.2: Model of the RAES.

Element	**Element kind**	**Adaptation type**	**Adaptation time**
Level style	Preference	Active	Any point during scenario
Level difficulty	Performance	Active	Predefined point(s) in scenario
Communication	Preference	Active	Any point during scenario
Motivators	Preference	Passive	Any point during scenario
Feedback	Preference	Passive	Any point during scenario
Time pressure	Performance	Active	Predefined point(s) in scenario

3.1.4 Options for adaptation in the RAES

With a model for elements in place, options of adaptivity for each preference element in the RAES were identified through a survey of several educational games developed and studies conducted by Scarlett (2015), Conati and Zhao (2004), Castellar et al. (2014) and Virvou, Katsionis, and Manos (2005). Table 3.3 presents the results of the survey identifying options for adaptability for the preference elements of the RAES:

Table 3.3: Options for the adaptability of the preference elements in the RAES.

Element	**Example options from literature**
Level style	*1. Memory* - ability to remember and recall information, patterns and procedures within the game. *2. Navigation* - ability to get to and from destinations without confusion or mistakes. *3. Quick thinking* - ability to apply knowledge under time pressure. *4. Problem solving* - ability to use information provided to reason a solution to the presented challenge. *5. Strategic thinking* - ability to use information given to infer further meaning that was not explicitly given.
Communication	*1. Direct* - conveying information in a precise manner using a longer statement like sentence structure with a higher vocabulary without using personal pronouns. *2. Indirect* - breaking up information into smaller units for ease of understanding. Also includes personal pronouns and colloquial/familiar language structured in sentences more akin to conversational dialog.
Motivators	*1. Direct hint* - literally telling the player how to proceed if he/she has not made progress. *2. Verbal hint* - suggesting in the right direction to progress without doing so explicitly. An example would be a hint reiterating the necessary information to remind the player to use it. *3. Visual hint* - using some form of visual change/indicator to point players in the right direction. *4. Encouragement* - motivational hints used to keep the player's morale high.
Feedback	*1. Audio* - some form of noise or cue that acknowledges when the player has performed an interaction. *2. Visual* - some form of visual cue or change to acknowledge a player's interaction. *3. World* - some form of change in the environment or NPC to acknowledge a player's interaction.

Lastly, Table 3.4 indicates the player metrics identified and the performance elements each are tied to.

Table 3.4: Links between player metrics and performance elements in the RAES.

Element	**Player metric**
Level difficulty	The number of player interactions taken to complete a scenario.
Time pressure	The amount of time a player takes to complete a scenario

3.2 Player monitoring technique

Concerning Adaptivity Issue 4, which refers to the issue of data collection, Chapter 2 explained that this issue can be remedied with the use of a player monitoring technique. Section 2.4.4 identified two techniques for player monitoring, PEM and MDM. Of the two, PEM allows for the PBD to be captured during a player's first (and potentially only) play-through of an educational game, whereas MDM does not. Hence, using PEM would also allow for a player model to be processed from a PBD, and for an educational game to be adaptive, during a player's first play session. This fullfills Motivating Requirement 3. Thus, PEM was reasoned to be the more suitable method of player monitoring.

However, of the three types of PEM, several problems with using subjective and objective PEM were identified. Three issues arise when considering the use of subjective PEM. First, it runs the risk of making a player aware of the player monitoring he/she is undergoing due to its intrusive nature of questionnaires and observation. This directly contradicts Adaptability Requirement 4. Second, it was believed that some players might find subjective monitoring off putting to the extent that he/she might not choose to play the educational game all together. This does not fulfill Adaptability Requirement 1 as it defeats the purpose of an adaptive educational game if potential players do not wish to play it. Third, in keeping with Motivating Requirement 2, this study did not want to assume that players would be available and/or willing to undergo surveys or interviews before the play session. Thus, subjective PEM was ommited.

Objective PEM was omitted due to requiring additional equipment such as cameras and facial recognition software to function. This breaks Motivating Requirement 1 and Adaptability Requirement 4. Since gameplay PEM involves monitoring interactions between the player and the educational game, it fulfills Adaptability Requirement 4 as it can be done discreetly. Furthermore, using gameplay PEM requires neither additional equipment nor prior interactions with players to function, thereby fulfilling Motivating Requirements 2 and 1. Hence, gameplay PEM was identified as the only applicable option for player monitoring in this study.

The use of gameplay PEM necessitates that the algorithm be initiated when a player interaction is detected. Concerning the types of player interactions described in Chapter 2, as gameplay PEM is solely being used, only explicit interactions would be taken into account. Explicit interactions are defined as ones a player makes when physically pressing a button on a keyboard, mouse and/or other peripheral.

Additionally, the algorithm should possess the ability to be initiated in the absence of player interaction. This is due to the speculation that there could exist situations where adaptation could benefit a player's learning that do not involve a player's interactions. As this could directly contribute to the algorithm's goal, it was reasoned to account for these situations in the interest of prudence. Thus, the player non-interaction trigger would allow adaptation to occur in situations where the is little to no player interaction. Specific player non-interaction triggers would greatly vary on the educational game(s) utilised and the choice(s) of the developers.

Thus, the algorithm's two triggers were defined as:

Algorithm Trigger 1 (Player interaction) *Any explicit player interaction can trigger the algorithm. This is defined as interactions that the player may perform within an educational game that is physically executed by pressing buttons, touch screens, moving a mouse and/or any other peripheral.*

Algorithm Trigger 2 (Non player interaction) *The algorithm can also be called in instances that do not relate to detecting a player's interactions. Two such triggers were identified for this study. The first is a timer that calls the algorithm at every second. This is due to time being one of the performance metrics* *chosen. Thus the player needs to be timed. The second is when a point of progress has been reached in an educational scenario. This is to set specific times of adaptation for elements that are to be adapted at predefined points.*

3.3 Adaptation conditions

Solving Adaptivity Issue 2, which refers to the issue of when to adapt, entails the identification of in game conditions where the adaptation of each of the elements of the RAES, could be of benefit to players' learning. Moving forward, these conditions will be referred to as adaptation conditions.

Considering learning using educational games in general, three situations where adaptation could aid a player's learning were derived. These situations are called *potential adaptation situations* and comprise:

Potential Adaptation Situation 1 *When a player's behaviour suggests they may not understand some educational content well enough to progress to complete a scenario.*

Potential Adaptation Situation 2 *When a player's behaviour suggests they do not know what to do next in a scenario. Examples of this include not being able to find an NPC and not remembering the route needed to get to a destination.*

Potential Adaptation Situation 3 *When a point in the game is reached at which substantial adaptations can occur to address accumulated evidence of how a player best learns.*

By applying the two algorithm triggers, from Section 3.2 to the three potential adaption situations when applicable, the following adaption conditions were derived:

Adaptation Condition 1 (Interaction completion) *When the number of interactions since this same condition was met has crossed the completion interaction threshold (CIT) for the current scenario. The completion interaction threshold is defined as the number of interactions that the player being monitored is expected to have completed a scenario by, for each level style option being used. A player performing more interactions than the completion interaction threshold might indicate that he/she does not understand the educational content and/or does not understand how to proceed with the scenario. Thus, this Adaptation Condition is checked when:*

1. *Algorithm Trigger 1 and Potential Adaptation Situation 1 are detected.*
2. *Algorithm Trigger 1 and Potential Adaptation Situation 2 are detected.*

The number of interactions since this same condition was met should be reset upon the completion of an educational scenario so that this condition is not met prematurely during the next scenario. Virvou, Katsionis, and Manos (2005) explain 3 possible reasons for a player performing more interactions than the expected baseline without having progressed. First, is that the player might not be adept at facing the style of challenge presented to them. This is represented by the level style element. Second, is the chance that the player is experiencing difficulty with comprehending educational information. This is represented by the communication and motivators element. Third, is that the player might not understand how to proceed in that scenario. This is represented by the motivators element. Therefore the level style, communication and motivators elements should be adapted when this condition is met.

Adaptation Condition 2 (Time completion) *When the time elapsed since this same condition was met has crossed the completion time threshold (CTT) for the current scenario. The completion time threshold is defined as the amount of time the player being monitored is expected to have completed a scenario by, for each level style option being used. A player taking more time than the completion time threshold might indicate that he/she does not understand the content and/or does not understand how to proceed with the scenario. Thus, this Adaptation Condition is checked when:*

1. *Algorithm Trigger 2 and Potential Adaptation Situation 1 are detected.*
2. *Algorithm Trigger 2 and Potential Adaptation Situation 2 are detected.*

The time elapsed since this same condition was met should be reset upon the completion of an educational scenario so that this condition is not met prematurely during the next scenario. Virvou, Katsionis, and Manos (2005) explain 3 possible reasons for a player taking more time than the expected baseline without having progressed. First, is that the player might not be adept at facing the style of challenge presented to them. This is represented by the level style element. Second, is the chance that the player is experiencing difficulty with comprehending educational information. This is represented by the communication and motivators element. Third, is that the player might not understand how to proceed in that scenario. This is represented by the motivators element. Therefore the level style, communication and motivators elements should be adapted when this condition is met.

Adaptation Condition 3 (Interaction redundant) *When too many consecutive redundant interactions are made with the same game object and/or NPC. This may indicate that the player does not understand the effects of his/her interaction. The number of consecutive redundant interactions that can be made before this condition is met is denoted by the redundant interaction threshold (RIT). Bunt, Conati, and Muldner (2004) utilised an RIT of five identical repetitions before assuming confusion. This threshold would also be adopted for this study. Developers may choose different RIT values based on their needs. Thus, this Adaptation Condition is checked when:*

1. *Algorithm Trigger 1 and Potential Adaptation Situation 2 are detected.*

Conati and Zhao (2004) point out that a player performing repeated identical redundant interactions multiple times, may not be aware of the effect(s)/lack of effect(s) of his/her interaction(s). As the effects of a player interactions is tied to the feedback element, it should be adapted once this condition has been met.

Adaptation Condition 4 (Interaction pause) *The pause between interactions is a greater than the expected pause threshold (EPT) for the player's last two interactions. The expected pause threshold is defined as the average time between the completion interaction threshold and the completion time threshold for the current scenario. This might suggest that the player does not understand the content, does not understand how to proceed with the scenario and/or the effects on his/her interaction. Thus, this Adaptation Condition is checked when:*

1. *Algorithm Trigger 1 and Potential Adaptation Situation 2 are detected.*

Conati and Zhao (ibid.) explain that a player may pause between interactions for longer than the expected amount of time in two instances. First, is that the player is unsure how to proceed with the scenario. This is represented by the motivators element. Second, is that the player is not aware of the effect of his/her previous interactions. This is represented by the feedback element. Therefore the feedback and motivators elements should be adapted when this condition is met.

Adaptation Condition 5 (Current scenario completed) *When the educational scenario currently being played is completed. Based on the player's performance in terms of the chosen performance metrics, the player may either require more or less adaptation for future scenarios. Thus, this Adaptation Condition is checked when:*

1. *Algorithm Trigger 2 and Potential Adaptation Situation 3 are detected.*

It was postulated in Section 3.1.3.3 that certain elements should only be adapted at predefined points during a scenario. These include the level difficulty and time pressure elements. As a result, these two elements should be adapted when this condition is met

Algorithm trigger 1 and Potential Adaptation Situation 3 can not coincide as it is the game engine that keeps track of the points of progress achieved. Therefore, this can only be triggered by the game engine, which falls under Algorithm Trigger 2. If at least one condition is met, a rule based is used to determine the course for adaptation. If no condition is met, the algorithm terminates.

Table 3.5 summarises the five adaptation conditions, their related triggers and the elements that should be adapted when each condition is met.

Table 3.5: Summary of adaption conditions, their related triggers and related elements.

Condition	**Condition name**	**Related trigger**	**Related elements**
1	Interaction completion	Player interaction	Level Style, Communication, Motivators
2	Time completion	Non player interaction	Level Style, Communication, Motivators
3	Interaction redundant	Player interaction	Feedback
4	Interaction pause	Player interaction	Motivators, Feedback
5	Current scenario completed	Non player interaction	Level difficulty, Time pressure

3.4 Player modelling technique

Adaptivity Issue 3, pertaining to how to adapt, concerns the determining of the element(s) to adapt and what they should be adapted to when each of the Adaptation Condition(s) are met. As explained in Section 2.4.3, this issue can be remedied using player modelling. Multiple techniques suitable for player modelling were identified. These include fuzzy logic, decision trees, Bayesian networks, hidden Markov models, neural networks and genetic algorithms.

However, the goal of this study was to create an overall system that could be used to make any educational game adaptive to benefit the learning of as many players as possible, as opposed to a review of the player modelling techniques available. Therefore, it was reasoned that applying rule based logic to each element of the AES could allow for player models to be updated whilst fulfilling all motivating requirements. Thus, in theory, researchers with different sets of motivating requirements could utilise the algorithm with different methods of player modelling while maintaining the rest of its functionality. The player model formulae and rule base used for this study are explained below.

3.4.1 Player model formulae

This subsection explains the formulae used to update the player model. To reiterate the work of Liu, Moon, and Kim (2020), there exist two ways to model players' proficiency, preference and performance. The elements of the AES were thus sorted into one of these two categories in Section 3.1.3.1. Different formulae were used for preference and performance elements due to their different roles in adaption.

3.4.1.1 Formulae for preference elements

Preference elements contain multiple options to execute their purpose. Since a player's goal is to learn the educational content by completing educational scenarios, the formulae of preference elements involve expressing a player's preference rate at completing educational scenarios while using each of the options of the elements. Using the implementations of Bontchev and Georgieva (2018) and Shabihi and Taghiyareh (2017) as a basis, preference rates for each option are expressed as the ratio of the number of scenarios completed while using that option to the number of times that option has been used. The preference rate of each option is determined via the formula:

$$Preference.Rate[option] = \frac{Scenarios.Completed.Using[option]}{Number.Of.Uses[option]} \tag{3.1}$$

Considering the RAES, the formula above can be applied to calculating the preference rate for the following options of each of the preference elements:

Level style: Memory, quick thinking, navigation, problem solving and strategic thinking.

Communication: Direct and indirect.

Motivators: Verbal hint, visual hint, direct hint and encouragement.

Feedback: Audio, visual, and world

3.4.1.2 Formulae for performance elements

With the implementations of Sandberg, Maris, and Hoogendoorn (2014), Clark et al. (2016) and Jagust, Boticki, and So (2018) serving as a basis, the aim of performance elements are to measure how well a player is succeeding at completing an educational scenario. This leads to determining the degree of adaptation he/she might need via the performance metrics linked to the elements. Since a player's goal is to learn the educational content by completing educational scenarios, the formulae for performance elements involve expressing how well a player performed at completing educational scenarios using two variables. These have been adapted from the three aforementioned implementations. The first known as a *completion multiplier* (CM) which compares a player's measured performance metric (P) to a baseline (B) of that same metric. A player's measured performance metric (P) refers to the value of a performance metric that a player took to complete a scenario. A baseline (B) is a measure of a performance metric that an averagely competent player is expected to have completed a scenario by. One way of determining a baseline is by taking the average of that performance metric recorded from player testing. A CM is calculated as the aggregate average of the ratio of B, P and the previous CM. The aggregate average is taken to account for potential improvements in a player's skill at the educational game and/or educational content. This is expressed via the formula:

$$CM = \frac{Previous.CM + \frac{B}{P}}{2} \tag{3.2}$$

Where the first CM is always 1.0.

Players whose performance is below the baseline would have a lower CM value, suggesting a higher likelihood of requiring more frequent adaptation and vice versa.

The second variable is called the *completion threshold* (CT) which sets a threshold indicating when adaption should occur. When the CT of a performance metric is crossed, it implies that a player may require adaptation to progress in the educational scenario. The CT is calculated by multiplying the CM with the baseline for that scenario via the formula below:

$$CT = B \times CM \tag{3.3}$$

The CT is inversely proportional to the player's performance. Thus under-performing players receive more adaptations and vice versa.

Two performance elements were identified in the RAES, which include the level difficulty and time pressure elements, which use the number of player interactions taken to complete a scenario and the time spent completing a scenario as their performance metrics respectively. Therefore, applying the formulae above to these two elements yielded the following results:

Level difficulty: The CM for this element is referred to as the *completion interaction multiplier* (CIM). The player's measured performance metric (P) is the number of interactions it takes for a player to complete a scenario (CI). The baseline (B) is the *baseline number of interactions* (BI) a player is expected to take to complete that same scenario. The formula of the CIM is expressed as:

$$CIM = \frac{Previous.CIM + \frac{BI}{CI}}{2} \tag{3.4}$$

Where the initial value of CIM is always 1.0

The CT for this element is the *completion interaction threshold* (CIT). The CIT is the absolute number of interactions a player can make before the algorithm is called to attempt adaptivity. This is calculated by multiplying the CIM with the BI for that scenario via the formula:

$$CIT = BI \times CIM \tag{3.5}$$

Time Pressure: The CM for this element is referred to as the *completion time multiplier* (CTM). The player's measured performance metric (P) is the amount of time it takes for a player to complete a scenario (CT). The baseline (B) is the *baseline amount of time* (BT) a player is expected to take to complete that same scenario. The formula of the CTM is expressed as:

$$CTM = \frac{Previous.CTM + \frac{BT}{CT}}{2} \tag{3.6}$$

Where the initial value of CTM is always 1.0

The CT for this element is the *completion time threshold* (CTT). The CTT is the absolute amount of time that can elapse before the algorithm is called to attempt adaptivity. This is calculated by multiplying the CTM with the BT for that scenario via the formula:

$$CTT = BT \times CTM \tag{3.7}$$

3.4.2 Rule base

The rule base describes the course of the adaptation of AES to take when one or more of the adaptation condition(s) have been met. These rules are used to determine which element(s) should undergo adaptation and the option to switch to. These rules only apply to preference elements. This is due to the adaptation of preference elements involving changing the option of the element to use which require rules to determine which option to switch to. Conversely, adaptation of performance elements involve altering the frequency of adaptation by recalculating the numerical values of completion multipliers (CM) and completion thresholds (CT). Since CMs and CTs are numerical values with specified formulae, they can be recalculated when Adaptation Condition 5 is met, and do not require rules to adapt.

The rules for preference elements are affected by five Boolean variables. These will be referred to as *rule variables* (RV) and include:

RV 1 *If it is the player's first educational scenario. The default value is TRUE.*
Shabihi and Taghiyareh (2017) explain that this variable needs to be tracked due to the fact that a player's preferred option can not be identified without said player having completed at least one educational scenario. The algorithm needs to identify and keep track of the options of preference elements that players successfully complete educational scenarios with in order to establish any form of preference towards said options. Hence at least one educational scenario must be completed to serve a starting point for players' preferences. So when a preference element needs to be adapted during the first educational scenario, Shabihi and Taghiyareh (ibid.) selected the next option available from their pool of options.

RV 2 *If there exist element specific rules. The default value is FALSE.*
See below for the element specific rules identified.

RV 3 *If any element specific rules have been met. The default value is NULL.*
See below for the element specific rules identified.

RV 4 *If the preferred option is currently being use or has been used for the current educational scenario. The default value is FALSE.*
Bontchev and Georgieva (2018) kept track of this in their implementation as they reasoned that there could exist situations and/or content where a player's preferred options might either not be suited or be failing for other reasons. Hence, they argue that different options should be tried if players preferred options have already been utilised and not produced any progress and/or success.

RV 5 *If all options have been used at least once. The default value is FALSE.*
Building on the concept that there could exist situations where a player is unable to progress or complete an educational scenario despite his/her preferred options being utilised, it was reasoned that utilising options that have not been utilised might be of benefit. This is reasoned to make the adaptation process as inclusive and prudent as possible since utilising every option would allow for preference rates of each option to be modeled, thereby increasing the accuracy of the player model.

Each preference element should have its own instances of the five RVs which should be checked when the appropriate adaptation condition(s) have been met (see Table 3.5 to determine which conditions affect which elements).

Furthermore, certain elements may also be bound by element specific rules which have to be met before they can be adapted. Two element specific rules for the RAES were identified and are explained below:

Element specific rule 1 (Level style rule) *Castellar et al. (2014) indicate a connection between the level style and communication style, by showing that players who do not understand the information and instructions being presented to them suffer from a higher probability of not being able to complete the scenario regardless of their adeptness of the challenge style. To account for this, two methods have been identified:*

1. *If it is the player's first scenario, all communication styles will need to have been tried before the level style element can be adapted.*
   Determined using the condition: If (RV1 is True) AND (RV5 for the communication element is True)

2. *If it is not the player's first scenario, the player's preferred communication options(s) must have been used for the current scenario before adaptions to the level style can be made.*
   Determined using the condition: If (RV1 is False) AND (RV4 for the communication element is True)

These are to ensure that the manner of challenge is the problematic element as opposed to the instructions/information. Therefore the level style element is dependent on the communication element.

Element specific rule 2 (Communication rule) *The player must be given motivators before the communication element can be adapted. Scarlett (2015) suggests a link between the motivators and communication element. It was shown that players who do not benefit from assistance provided by motivators have a higher probability of failure due to the style in which the information was conveyed rather than the content of the information itself. Therefore, motivators must be given and have no effect on the player's progress before changing the communication style to ensure that the communication style is the issue. This is done in two ways:*

1. *If it is the player's first scenario, all motivator styles will need to be tried before adapting the communication element.*
   Determined using the condition: If (RV1 is True) AND (RV5 for the motivators element is True)

2. *If it is not the player's first scenario, the player must have received motivators using his/her preferred option before the communication element can be adapated.*
   Determined using the condition: If (RV1 is False) AND (RV4 for the motivators element is True)

These are done to ensure that the player's learning could benefit from a change in the way information is presented to him/her. Therefore the communication element is dependent on the motivators element.

Two internal dependencies exist amongst the RVs. First, RV 4 is dependent on RV 1 being false. This is due to the player needing to have completed at least one educational scenario to establish any preferred options. Second, RV 3 is dependent on RV 2 being true. This is because element adaption conditions must exist before said conditions can be checked. Table 3.6 summarises the rules determining the course of adaption for each preference element depending on the truth value of the five RVs:

Table 3.6: Lookup table for preference element rules.

RV 1	RV 2	RV 3	RV 4	RV 5	Course of adaption
T	T	T	null	T	Switch to the next option available
T	T	T	null	F	Switch to the next option that has not been used
T	T	F	null	T	Do nothing
T	T	F	null	F	Do nothing
T	F	null	null	T	Switch to the next option available
T	F	null	null	F	Switch to the next option that has not been used
F	T	T	T	T	Switch to the next option available
F	T	T	T	F	Switch to the next option that has not been used
F	T	T	F	T	Switch to the preferred option
F	T	T	F	F	Switch to the preferred option
F	T	F	T	T	Do nothing
F	T	F	T	F	Do nothing
F	T	F	F	T	Do nothing
F	T	F	F	F	Do nothing
F	F	null	T	T	Switch to the next option available
F	F	null	T	F	Switch to the next option that has not been used
F	F	null	F	T	Switch to the preferred option
F	F	null	F	F	Switch to the preferred option

When adaptation conditions are met, this rule base should be applied to each of the affected elements.

3.5 Summary

Solutions to each of the four Adaptivity Issues were addressed by this Chapter. Regarding Adaptivity Issue 1, each relevant element from the GES were mapped to one of the four stages of educational scenarios, explained in Section 2.2.1.2. Using a combination of this study's research question and the literature, four adaptability requirements were derived. Any element whose adaptation fulfills these four requirements is deemed to be adaptable and is place in the adaptable elements set (AES). By applying the four requirements to the general element set (GES), six elements remained, comprising the AES. This specific instance of AES is referred to as the recommended adaptable elements set (RAES) and is reasoned to contain elements whose adaptation is believed to be most suitable for the benefit of a player's learning. The elements of the AES were modelled along three dimensions. The first is the kind of element, whether it is realised as an unordered set of options (preference element) or if it takes the form of ordered numerical values (performance elements). Second, is the type of adaptivity being executed by the element. Third, is the time at which the element should be adapted. Lastly, a survey of several educational games and studies identified the possible options for the adaptation of preference elements in the RAES.

Concerning Adaptivity Issue 2, gameplay PEM was chosen as the method of player monitoring to allow for adaptation to occur during a player's first time through. Two the two algorithm triggers of a player interaction and non-player interaction were also identified. Adaptivity Issue 3 was addressed by deriving five adaptation conditions that indicate when adaptation of each of the elements in the RAES might be of benefit to the player's learning. These conditions were derived from combining the identified algorithm triggers with three potential adpatation situations. Lastly, Adaptivity Issue 4 was addressed using a rule based logic system as the player modelling technique. This system used was split into the formulae for both preference and performance elements used to update the player model, and the rules used to determine the best course for adaptation. The next chapter delves into the design of an algorithm that implements the design specification explained in this chapter.

Chapter 4

Algorithm implementation

The previous chapter presented a design specification that identified solutions to each of the four Adaptivity Issues. Moving forward, an algorithm that fulfills said design specification would need to be implemented. It is reasoned that such an algorithm could adapt any element in the AES when it is determined that adaptation could benefit a player's learning, thereby accomplishing the goal of this study. The implementation of such an algorithm is presented in this chapter. This is split into four sections. The first presents the high level explanation for the functionality and structure of the algorithm. The second explains the inputs and outputs used by the algorithm comprising the player model (PM), player behaviour dataset (PBD), *adaptation statistic dataset* (ASD), *scenario benchmark dataset* (SBD), the *running state dataset* (RSD) and the *adaptation command* (AC). Full details on the ASD, SBD, RSD and AC are presented in Sections 4.2.3, 4.2.4, 4.3.5 and 4.3.6 respectively. The third expresses the five rule variables from Section 3.4.2 using the data structures explained in Section 4.2. The fourth presents the pseudo-code for the entire algorithm along with low level explanations.

4.1 High-level structure of algorithm

The goal of this algorithm would be to make adaptations that could improve the learning of players. This could be accomplished by ensuring that the algorithm adheres to the design specification from Chapter 3. From reviewing the adaptive entertainment and educational games in Section 2.3 along with the solutions to the four Adaptivity Issues discussed in Chapter 3, it was reasoned that the algorithm would need to perform different tasks to accomplish its goal. These tasks were classified into the following three distinct stages:

Algorithm Stage 1 (Player monitoring) *Monitoring a player's behaviour while he/she plays an educational game allows for a PBD to be updated. The PBD is one of the data sets needed to update a player model, which is needed to inform adaptations. The output for this stage is the updated PBD.*

Algorithm Stage 2 (Player modelling) *The purpose of this stage is to update the player model (PM). The player model represents a player's preference and/or performance for each of the elements in the AES. The input data needed include the PBD, statistics of previous adaptations made (ASD), static benchmark data specific to each educational scenario (SBD) and data pertaining to the educational game's running state (RSD). The output is the updated player model.*

Algorithm Stage 3 (Executing adaptation) *The PM, PBD, ASD, SBD and RSD serve as input and are checked against conditions to determine the element(s) whose adaptation could be beneficial to the players learning, if any. If adaptation(s) are deemed beneficial, a rule base is used to determine the option(s) to switch to for said element(s). An adaptation commands(s) (AC) to the game engine to execute any adaptation(s) is then generated. The outputs of this stage are the adaptation command(s).*

Figure 4.1 shows the high level functionality and data flow of the algorithm for each of the three stages. Algorithm Stage 1 starts when either a player interaction call or a game engine call is made. If the trigger was a player interaction, the algorithm will check whether it is a *relevant interaction*. A relevant interaction is defined as the type of interaction that needs to be performed to advance the current scenario and must be manually defined by the game designer for each scenario. If so, it will continue on to update the player model. If not it will terminate. This is so that interactions irrelevant to progressing the scenario but are necessary to play the game are not counted. For example, in a game which allows a player to move freely around a virtual world while tasking him/her to pick a specific object amongst numerous objects, the player will need to move around the game world in order to search for and pick up that object. Thus, the interactions of searching and picking up objects are considered relevant, but the interactions of moving around and changing where the playable character is looking, are not. Different scenarios and/or different educational games might require different types of interactions to progress. Therefore, relevant interactions might differ from scenario to scenario and from game to game. As a result of this, developers would need to define relevant interactions for each scenario in each video game.

If the algorithm was triggered by a game engine call or a relevant interaction, Algorithm Stage 2 is started. The PBD, ASD, SBD and RSD are used as input to update the player model. The adaptation statistics dataset (ASD) contains data regarding all adaptations for all elements in the RAES that have been made for a specific player. The scenario benchmark dataset (SBD) comprises static benchmark data pertaining to the completion of each educational scenario that must be defined by the game designer and may differ from scenario to scenario. The running state dataset (RSD) contains information pertaining to the state that the educational game is currently in and the options of elements in the RAES currently being used. The updated player model serves as the output for this stage.

Algorithm Stage 3 then starts by checking the adaptation conditions using the updated PM, PBD, ASD, SBD and RSD. If one or more of the adaptation conditions is met, the algorithm determines the course of adaptation to be made for each of the affected elements using the rule base. Adaptation command(s) (AC) are then generated. An AC is defined as a command to the game engine that will execute any of the adaptation(s) deemed beneficial by the algorithm. The output of this stage consists of adaptation command(s). The algorithm then terminates. If no condition has been met, the algorithm terminates.

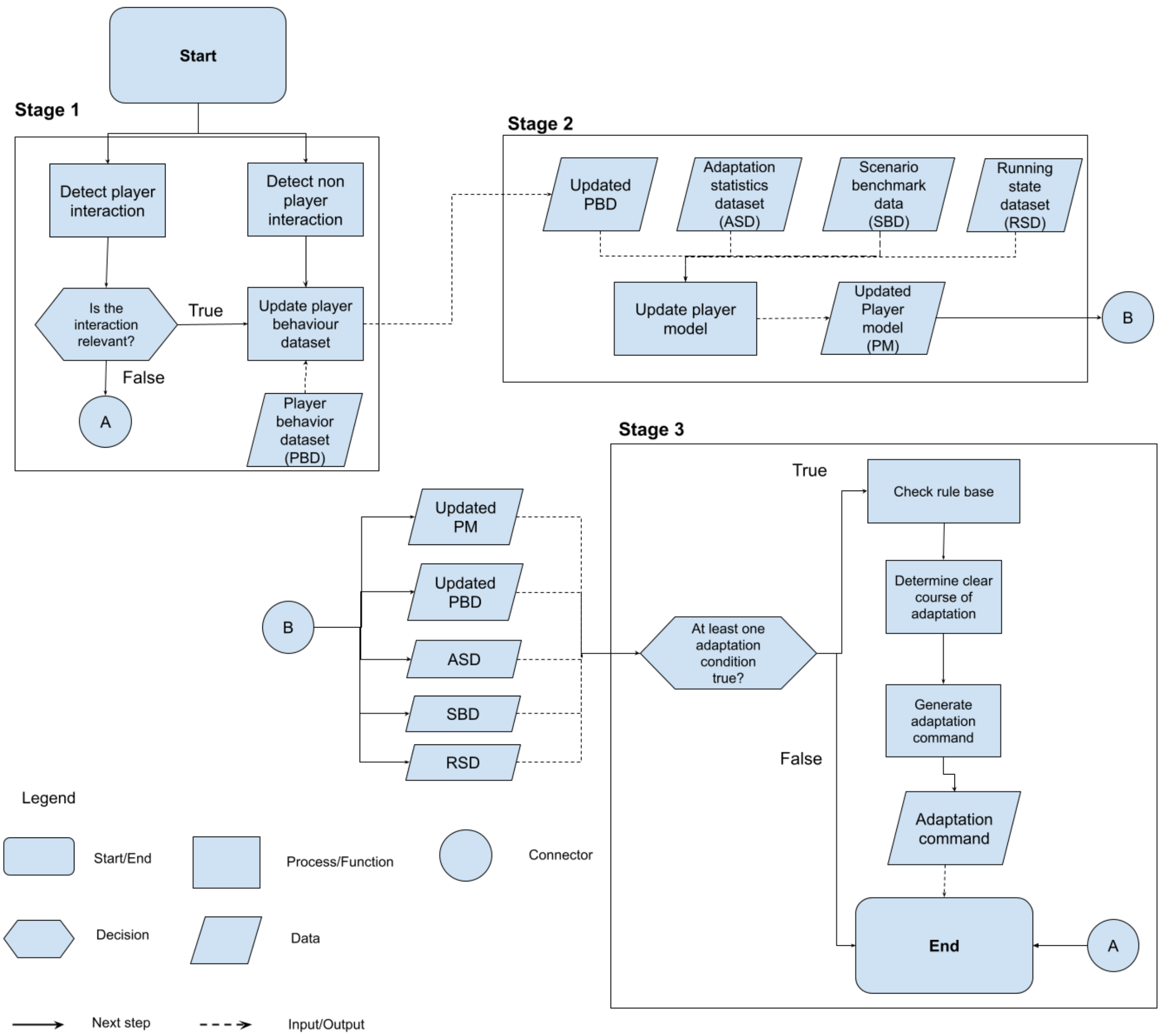

Figure 4.1: High level structure of the algorithm

4.2 Inputs and output

This section details the algorithm's inputs and outputs which include the player model (PM), player behaviour dataset (PBD), adaptation statistics dataset (ASD), scenario benchmark dataset (SBD), running state dataset (RSD) and adaptation commands (AC).

4.2.1 Player model

The purpose of the player model is to determine the element(s) in the AES whose adaptations might be beneficial to learning at that specific moment when at least one Adaptation Condition has been met. Considering the goal of this study is to make educational games adaptive in a manner that improves players' learning, the player model is reasoned to be a salient component. The player model refers to a set of data representing the player's ability at completing educational scenarios while each of the differing options of the AES are active. The player model comprises high level categories, each of which represents an element chosen from the AES.

Concerning preference type elements, using the implementations of Bontchev and Georgieva (2018) and Shabihi and Taghiyareh (2017) as a basis, each high level category contains two variable maps. The first is a variable map of the player's preference rate of using each option of each element of the AES to complete educational scenarios. This takes the form the form of (option, preference rate). The second is a set containing the preferred option(s).

Concerning performance elements, with the implementations of Sandberg, Maris, and Hoogendoorn (2014), Clark et al. (2016) and Jagust, Boticki, and So (2018) serving as a basis, each high level category contains two variable maps. The first compares a player's performance to a combination of the SBD and that same player's previous performances for each scenario. The second contains thresholds for each scenario which indicate when adaptation may next be beneficial to a player's learning. The data shown in Table 4.1 is therefore used by the algorithm to deduce if adaptation would aid the player with learning. In future sections, variables from the player model will be referred to via its label.

Table 4.1: The structure of the player model.

Category	**Label**	**Description of variable**
Preference elements	PM.1A	Map of each option and the player's preference rate of using that option to complete scenarios.
	PM.1B	Set of the options(s) whose preference rate equals the maximum value of the first variable map.
Performance elements	PM.2A	Map of completion multipliers (CM) for each scenario.
	PM.2B	Map of completion thresholds (CT) for each level style option being used for each scenario.

Data concerning players' behaviour, previous adaptations made, the educational scenario currently being played and the state of the educational game would be required to calculate the player model described above. The necessary data was sorted into the PBD, ASD, SBD and the RSD respectively. Thus the data fields in each of these four data sets were derived by identifying the game and player data needed to calculate the values in the player model. Furthermore, once the adaptation that could be beneficial to learning has been identified, said adaptation would need to be executed. The AC was established to perform such functionality.

Each of these data sets are detailed below.

4.2.2 Player behaviour dataset

As explained previously, monitoring a player's behaviour within an educational game produces a player behaviour data set. The PBD consists of data, pertaining to a players' behaviour, that is needed to update the player model. The PBD contains records of the algorithm triggers that have been detected. These include player interactions (Algorithm Trigger 1) or non-player interactions (Algorithm Trigger 2), for both timing the player and checking if a scenario has been completed. A PBD is created upon a player's very first play session. Furthermore, the PBD is updatable in order for that same player to be monitored throughout multiple play sessions, which could occur across multiple machines provided that the PBD can be shared across those machines. Although the algorithm can still function using a new PBD for a returning player, that player model would not be as accurate which could lead to adaptations that have no benefit on learning. Therefore, it is ideal that each player uses his/her up to date PBD during any play session.

Table 4.2 shows the PBD containing the minimum required data for the player model to be updated on the algorithm triggers identified. The PBD could require additional data if more triggers are identified. In future sections, data from the PBD will be referred to via its label.

Table 4.2: The structure of the player behaviour dataset.

Label	**Description of variable**
PBD.1	Total relevant interactions performed.
PBD.2	Number of relevant interactions performed for current scenario.
PBD.3	Sequence of the the IDs of the most recent relevant interactions, with a sequence length corresponding to the chosen RIT.
PBD.4	Total time spent playing.
PBD.5	Time spent for current scenario.
PBD.6	Expected pause threshold. Calculated using the formula: $$PBD.6 = \frac{CIT}{CTT}$$
PBD.7	Time of last interaction.
PBD.8	Total number of scenarios completed.

4.2.3 Adaptation statistics dataset

In addition to the PBD, the adaptation statistic dataset contains data regarding previous adaptations made that is required to update the player model and determine whether Adaptation Condition 1 and/or Adaptation Condition 2 is met. This comprises high level categories, each of which represents an element chosen from the AES.

Concerning preference type elements, each high level category contains two variable maps. The first counts the number of times each option for each element has been used. The second contains the number of scenarios completed while using each of the options for each of the elements. These are used in the generation of the player model.

Concerning the performance elements, each high level category contains two variables. The first keeps track of each of the chosen player metrics since the completion threshold attributed to that metric was last crossed. This is used to determine when the algorithm should attempt adaptation. The second keeps count of each of the player metrics that were used to complete each educational scenario. This is used to update the player model.

Table 4.3 shows the general structure of the ASD. All variables will be referred to via their label in future sections.

Table 4.3: The structure of the adaptation statistics dataset.

Category	**Label**	**Description of variable**
Preference elements	ASD.1A	Map of the number of times each option has been used.
	ASD.1B	Map of the number of scenarios completed when each of the options was the one being used.
Performance elements	ASD.2A	Count of the player metric since the completion threshold attributed to that metric was last crossed.
	ASD.2B	Map measuring the player metric taken to complete each scenario.

4.2.4 Scenario benchmark dataset

In addition to the PBD and the ASD, the algorithm requires the scenario benchmark dataset to update the player model. The SBD contains static benchmarks for each educational scenario made and must be manually defined by a developer. Table 4.4 displays the minimum required SBD for the player model to be updated. This comprises the following data:

Table 4.4: The structure of learning benchmark data.

Label	**Description of variable**
SBD.1	Map of relevant interactions for each scenario.
SBD.2	Map of the *baseline number of interactions* (BI) for each level style option being used to complete each scenario. The baseline number of interactions is defined as the number of interactions an averagely competent player can be expected to complete a scenario.
SBD.3	Map of the *baseline times* (BT) for each level style option being used to complete each scenario. The baseline times are defined as the amount of time an averagely competent player can be expected to take to complete a scenario.

In future sections, scenario benchmark data will be referred to via its label. When considering the values of SBD.2 and SBD.3 for each scenario, ideally each baseline should be taken as the average number of interactions and time taken to complete the scenario. This would require play testing of the educational game with adaptivity enabled using randomly selected members of the game's intended audience as the participants.

4.2.5 Running state dataset

During gameplay, an educational game would have a *running state*. A running state is defined as a dataset containing information pertaining to the state that the educational game is currently in and the options of elements in the RAES currently being used. As an educational game has a running state that adaptations would alter, data pertaining to that running state is also a relevant to to Algorithm Stages 2 and 3. This is referred to as the running state dataset (RSD).

The RSD comprises two high level categories. The first category contains variables that track data universal to educational games which include the educational scenario currently being played and the in-game time. The second category contains variables indicating the option currently in use of the preference elements. Only preference elements are included in the RSD due to their adaptation involving the changing of options being used. Thus, information pertaining to the option(s) in use prior to adaptation are needed to generate adaptation commands to change said option(s). Table 4.5 displays the general structure for the RSD.

Table 4.5: The structure of the running state dataset.

Category	**Label**	**Description of variable**
Universal	RSD.1	Current scenario.
	RSD.2	Current game time.
Preference elements	RSD.3	Current option being used for each preference element.

4.2.6 Adaptation command

Lastly, is the adaptation command to the game engine. The adaptation command (AC) is defined as a command to the game engine to carry out any adaptations that have been instructed by the algorithm.

When the algorithm calculates that at least one of the elements in the RAES requires adaptation and what to adaptions to make, it is the game engine that would execute the change(s) in said element(s). Therefore, command(s) to the game engine to make said changes would have to be generated. Each AC would adapt one element. Thus multiple ACs would need to be generated when more than one element requires adaptation.

The algorithm will output these command(s) to the game engine which would then switch the option currently being used to the option instructed by the algorithm. Although commands might differ from engine to engine, the desired functionality would remain the same. The functionality that a command must enact is:

1. Switch the desired element(s) option from its current option to the instructed option.
2. If the switching of options requires new game assets and scripts to be functional:
   (a) Load in assets and scripts attributed to the new option.
   (b) Unload assets and scripts attributed to the previous option.

Concerning step two of the AC, the changing of options for certain elements might require new game assets and scripts to be functional. For example, switching the level style from the memory option to the navigation option might require a new area in the game with obstacles which would need to be loaded in. In this case, the game engine should load in the new area and obstacles, and unload the questions and question use interface that was used for the memory option.

4.3 Rule base mapping

Regarding the rule base for player modelling established in Section 3.4.2, each of the rule variables (RVs) would need to be mapped to their corresponding data structures explained in Section 4.2 so that their values may be checked and changed accordingly. Table 4.6 details each of the five RVs labels, their description, default values and the expression for checking their values.

Table 4.6: The mapping of rule variables to data from the input and output data sets.

Label	**Description of variable**	**Default value**	**Expression**
RV1	If it is the player's first educational scenario.	TRUE	If (PBD.8 <1)
RV2	If there exist element specific rules	FALSE	Must be set manually by designer
RV3	If any element specific rules have been met.	NULL	Must be set manually by designer
RV4	If the preferred option is currently being use or has been used for the current educational scenario.	FALSE	If (PM.1B contains RSD.3)
RV5	If all options have been used at least once	FALSE	If (ASD.1A contains a zero value)

4.4 Pseudocode

With the design, inputs, outputs and the applied rule base mapping explained, below is the pseudo-code and low level descriptions for the algorithm.

4.4.1 Main algorithm

Algorithm 1 is responsible for monitoring the player's behaviour, checking the adaptation conditions and calling the element specific algorithms depending on the conditions that have been met. The element specific algorithms called correspond with the adaption conditions shown in Table 3.5. Hence it executes Algorithm Stage 1 and Algorithm Stage 3. Algorithm 1 uses the PBD, ASD, SBD and RSD as input. Outputs include the PBD and ASD.

Algorithm 1: Main Algorithm

```
     Input          : PBD, ASD, SBD, RSD
     Output         : PBD, ASD
     Local Variables: playerInteraction
(1)  if Trigger is playerInteraction then
(2)      if SBD.1[RSD.1] contains playerInteraction then
(3)          PBD.1++
(4)          PBD.2++
(5)          for int i = 0, i <RIT, i++ do
(6)              PBD.3[i+1]= PBD.3[i]
(7)          PBD.3[0] ← playerInteraction
(8)          PBD.7 = RSD.2
(9)          for All Adaptation Conditions that are linked to Algorithm Trigger 1 do
(10)             if An Adaptation Condition is met then
(11)                 for Every element affected by the Adaptation condition do
(12)                     if The element is preference then
(13)                         PreferenceElementAdaptation(ASD, PM, RSD)
(14)                     else
(15)                         PerformanceElementAdaptation(ASD, PM, RSD)
(16) else
(17) PBD.5++
(18) for All Adaptation Conditions that are linked to Algorithm Trigger 2 do
(19)     if An Adaptation Condition is met then
(20)         if The Adaptation Condition is Adaptation Condition  5 then
(21)             PerformanceElementsDataUpdate(PBD, ASD, PM, RSD)
(22)         for Every element affected by the Adaptation condition do
(23)             if The element is preference then
(24)                 PreferenceElementAdaptation(ASD, PM, RSD)
(25)             else
(26)                 PerformanceElementAdaptation(ASD, PM, RSD)
(27) Return (PBD, ASD)
```

The line by line explanation for Algorithm 1 explanation is as follows:

Lines 1 and 16. Checks if the algorithm was trigger by a player interaction or a game engine call.

Line 2. Checks if the player interaction that triggered the algorithm was a relevant one.

Lines 3 to 4. Increments the total number of relevant interactions performed and the relevant interactions performed for the current scenario

Lines 5 to 7. Saves the player's last interactions corresponding with a chosen RIT.

Line 8. Saves the time of the players last interaction.

Lines 9 and 10. Checks if any adaptation conditions associated with Algorithm Trigger 1 are met. These include Adaptation Conditions 1, 3 and 4.

Lines 11. Identifies any element that is affected by the adaptation condition(s) met following Table 3.5.

Lines 12 to 15. Checks if the element(s) to be adapted are preference or performance and calls to appropriate algorithms to enact the adaptation.

Line 17. Increments the time spent on the current scenario.

Lines 18 and 19. Checks if any adaptation conditions associated with Algorithm Trigger 2 are met. These include Adaptation Conditions 2 and 5.

Lines 20 to 21. Checks if Adaptation Condition 5 is met, and calls algorithms 4 if it is to update the data needed for the adaptation of performance elements.

Lines 22 to 26. Checks if the element(s) to be adapted are preference or performance and calls to appropriate algorithms to enact the adaptation.

Line 27. Returns the PBD and ASD.

4.4.2 Preference element algorithm

Algorithm 2 aims to update the preference elements high level category of the player model and execute the adaptation of each preference element using the rule base. Thus, it enacts Algorithm Stage 2 and Algorithm Stage 3. It uses the ASD, PM and RSD as input, and outputs the ASD, RSD and an AC.

Each preference element should have it's own algorithm replicating the functionality displayed in the pseudo code below.

Algorithm 2: PreferenceElementAdaptation

```
    Input          : ASD, PM, RSD
    Output         : ASD, RSD, AC
    Local Variables: newOption, options[] = selected options for the element
(1)  if RV 1 is TRUE then
(2)      if RV 2 is TRUE then
(3)          if RV 3 is TRUE then
(4)              newOption = SelectNextOptionIn(options[])
(5)      else
(6)          newOption = SelectNextOptionIn(options[])
(7)  else
(8)  if RV 2 is TRUE then
(9)      if RV 3 is TRUE then
(10)         PreferenceElementRatio(ASD)
(11)         if RV 4 is TRUE then
(12)             if RV 5 is FALSE then
(13)                 newOption = OptionWithZeroValue(ASD.1A)
(14)             else
(15)                 newOption = SelectNextOptionIn(options[])
(16)         else
(17)             newOption = SelectNextOptionIn(PM.1B)
(18) else
(19)     PreferenceElementRatio(ASD)
(20)     if RV 4 is TRUE then
(21)         if RV 5 is FALSE then
(22)             newOption = OptionWithZeroValue(ASD.1A)
(23)         else
(24)             newOption = SelectNextOptionIn(options[])
(25)     else
(26)         newOption = SelectNextOptionIn(PM.1B)
(27) GameEngine.ChangeElementOption(newOption)
(28) ASD.1A[RSD.3]++
(29) Return (ASD, RSD, AC)
```

The line by line explanation for Algorithm 2 explanation is as follows:

Lines 1 and 7. Checks if the educational scenario being played is the player's first one.

Lines 2 and 3. Checks if there exist any element specific adaptation conditions and if they have been met.

Lines 4 and 6. Sets the option to be used as the next option on the list.

Lines 8 and 9. Provided it is not the player's first educational scenario, Checks if there exist any element specific adaptation conditions and if they have been met.

Line 10. Calls Algorithm 5 to update the preference rates of the options for the element in question.

Line 11 and 17. Checks if the preferred option is currently being used or has already been used for the current scenario and sets the new option to be used as the preferred option if the check is false.

Lines 12 to 15. Checks if all options for the element have been used least once. It sets the new option to be used as an unused option if the check is true and the next option on the list if the check is false.

Lines 19 to 26. Replicates the functionality of Lines 10 to 17 given that there exist no element specific adaptation conditions.

Line 27. Generates an adaptation command to change the option of the preference element to what ever option was determined.

Line 28. Increments the counter for the number of times the new option has been used.

Line 29. Returns the ASD, RSD and AC.

4.4.3 Performance element algorithm

Algorithm 3 aims to update the performance elements high level category of the player model and execute the adaptation of each performance element. Thus, it enacts Algorithm Stage 2 and Algorithm Stage 3. It uses the ASD, SBD and RSD as input, and outputs the PM.

Each performance element should have it's own method replicating the functionality displayed in the pseudo code below.

Algorithm 3: PerformanceElementAdaptation

```
Input           : ASD, RSD, SBD
Output          : PM
Local Variables:
(1) PM.2A[RSD.1] = (PM.2A[RSD.1-1] + (SBD[RSD.3][RSD.1-1] / PBD.2B[RSD.1-1])) / 2
(2) PM.2B[RSD.3][RSD.1] = PM.2A[RSD.1] ×SBD[RSD.3][RSD.1]
(3) Return (PM)
```

The line by line explanation for Algorithm 3 is as follows:

Line 1. Calculates the CM for the new scenario.

Line 2. Calculates the CT for the new scenario.

Line 3. Returns the PM.

4.4.4 Support algorithms

The following are support algorithms used by Algorithms 2 and 3 for their own internal functionality.

4.4.4.1 Performance element data update algorithm

When a scenario is completed, data in the PBD, ASD and RSD need to be updated in order to execute the adaptation of the performance elements. Algorithm 4 fulfills this purpose as shown below.

Algorithm 4: PerformanceElementsDataUpdate

```
     Input            : PBD, ASD, PM, RSD
     Output           : PBD, ASD, RSD
     Local Variables:
(1)  ASD.4A = PBD.1
(2)  ASD.4A = PBD.4
(3)  ASD.2B = PBD.2
(4)  ASD.2B = PBD.5
(5)  PBD.8++
(6)  RSD.1++
(7)  PBD.2 = 0
(8)  PBD.6 = CIT/ CTT
(9)  PBD.5 = 0
(10) for All preference elements do
(11)     ASD.1B[RSD.3]++
(12) Return (PBD, ASD, RSD)
```

The line by line explanation for Algorithm 4 explanation is as follows:

Line 1. Sets the player interaction count since adaptation condition 1 was last met to the player's total interaction count to reset the condition for the next scenario.

Line 2. Sets the time count since adaptation condition 2 was last met to the player's total play time to reset the condition for the next scenario.

Line 3. Saves the number of interactions the player took to complete the scenario.

Line 4. Saves the amount of time the player spent to complete the scenario.

Line 5. Increments the total number of scenarios the player has completed.

Line 6. Increments the current scenario being played

Line 7. Resets the player relevant interaction count for the current scenario to zero.

Line 8. Calculates the player's average time between interactions using the formula in 4.2.

Line 9. Resets the time that the player has spent on the current scenario to 0.

Line 10 and 11. Increments the counter for the number of scenarios completed using the options currently in use for all preference elements.

Line 12. Returns the PBD, ASD and RSD.

4.4.4.2 Preference element ratio algorithm

The purpose of algorithm 5 is to update the preference rates of all options and determine the player's preferred option for each preference element. It executes Algorithm Stage 2. Each preference element should have its own method to update its preference ratios. Pseudo code displaying the general structure to calculate the preference rations for any preference element is displayed below.

Algorithm 5: PreferenceElementRatio

```
    Input          : ASD
    Output         : PM
    Local Variables: maxValue, options[] = selected options for element
(1) for All options o in options[] do
(2)     PM.1A[o] = ASD.1B[o] / ASD.1A[o]
(3) maxValue = Maximum(PM.1A)
(4) for each option in options[] do
(5)     PM.1B = OptionCorrespondingWith(maxValue)
(6) Return (PM)
```

The line by line explanation for Algorithms 5 is as follows:

Lines 1 to 2. Updates the preference rates for all options for the element in question and save it in the PM.

Line 3. Save the maximum value of the preference rates in a local variable.

Lines 4 to 5. Checks the the preference rates of any option that matches the maximum value and saves the option(s) in the PM.

Line 6. Returns the PM.

4.5 Summary

To summarise, the purpose of the algorithm is to make adaptations in an educational game that improve a player's learning. This is broken down into the three stages of monitoring a player's behaviour while playing an educational game, generating that player's player model, and executing the adaptation of the instructed elements. The five input data structures were explained as the player behaviour dataset, adaptation statistics dataset, scenario benchmarks dataset, the player model and the running state dataset. The output data structures include the updated player model and adaptation commands. The five rule variables in the rule based were given mapping using the data from the input and output data structures so that they may have their values checked and changed. Algorithms executing the three stages were defined in pseudo code and given low level explanations. With the implementation of the algorithm complete, the following chapter will explore evaluation methods.

Chapter 5

System testing

With the design of the algorithm specified in the previous Chapter, the next contribution involves evaluating the algorithm's ability to positively affect players' learning gains. This would be best determined by running a user evaluation pairing the algorithm with an educational game. However, it was reasoned that system testing the algorithm before running a user evaluation was prudent. This was to ensure that the algorithm behaved and produced outputs as intended, given the correct conditions. The system testing was done via simulation so that the algorithm's function could be tested without the need to identify educational content and develop educational scenarios, which require significant time and resources. This chapter focuses on explaining the system testing of the algorithm prior to the user evaluation explained in Chapter 6. This includes defining a clear goal and hypothesis, establishing a testing procedure, making high level decisions for the testing, generating the needed data, conclusions drawn from the results and the decisions carried forth to the user evaluation.

5.1 System testing by simulation

Researchers such Garousi and Pfahl (2016) and Lazić and Velašević (2004) define system testing by simulation as the process of designing a computerised model of a system (or process), and conducting experiments with this model using simulated inputs and/or stimuli. This is done with the purpose of either of verifying that the system behaves as intended when given the expected inputs, or evaluating various strategies for the operation of the system. Garousi and Pfahl (2016) go on to explain that, observing and analysing the system's behaviour using different inputs can provide insights into the designs of processes, architectures, or product lines before significant time and cost have been invested. Thus, system testing by simulation can be of great benefit in support of a full scale evaluation.

A wide selection of methods and analysis techniques for system testing by simulation have existed for decades and have been documented by researchers such as Montgomery (1997). Lazić and Velašević (2004) extracted a general framework from said compilation of techniques, which they condensed into the following steps:

1. Define the goals of the experiment.
2. Choose a suitable testing procedure for the system being tested, which could include identifying and classifying the inputs and expected outputs of the model, the length of each run, the number of runs

to be performed, etc.

3. Choose a suitable probability model for the behaviour of the simulation model.
4. Select a suitable method(s) to analyse and draw conclusion from the results.

Looking back to the literature (from 2004 to 2016), the definition of system testing by simulation has remained the same. This indicates that the overall goal and purpose of system testing by simulation in general has remained constant. Thus, it is reasoned that this framework could still hold relevance despite its age. As a result, the system testing conducted below followed this general framework.

5.2 Testing goals and hypothesis

Concerning step 1, the overall goal of the system testing was to confirm that the algorithm functioned as intended. Correct function of the algorithm would be denoted by the algorithm's output player models being closer to reference player models than false player models. A reference player model is defined as a completely accurate player model of a player or simulated player. A false player model is defined as a completely inaccurate player model of a player or simulated player. Thus, the following hypothesis was established for the purpose of the algorithm's system testing:

H0 = The algorithm's output player models will not be closer to the reference player models than the false player model.

H1 = The algorithm's output player models will be closer to the reference player models than the false player model.

The method used for system testing is explained below.

5.3 Testing procedure

When considering step 2 of the framework (the testing procedure), Algorithm Stage 1 begins with monitoring players and saving their behavioral data in PBDs. As the only method of system testing available was by simulation, simulated players took the place of actual players. A simulated player is defined as an NPC whose behaviour is scripted. Due to this, the behavioural data of each simulated player would have to be generated. This was done by generating *Simulated player scripts* (SPSs) that would denote the behaviour of each simulated player that the algorithm would then monitor and adapt to. Drawing on the goal of the user testing, the use of simulated players meant that their reference player models and false player models would also have to be generated. Reference player models and false player models are generated as pairs. See Sections 5.5.1, 5.5.2 and 5.5.3 for full details on the generation of the reference player models, false player models and SPSs.

With the SPSs and reference player models generated, the testing procedure involved getting a simulated player to behave in accordance to the SPS when placed in simulated scenarios. The algorithm monitored it, updated its player model and executed adaptations throughout the test. A simulated educational scenario is defined as a scenario that a simulated player will have completed by the number of interactions and time based on values of its SPS. When simulated players completed all the simulated scenarios given, the algorithm outputted updated player models for each simulated player. Concerning the analysis of the results (step 4 of the framework), the variables in output player models were compared to their counter parts in the reference player models and false player models to check which they matched closer to, which would determine the algorithm's success at accomplishing its goal. Thus, the system testing procedure was broken up into the following steps:

1. Generate reference player models for each simulated player.
2. Generate false player models for each simulated player.
3. Generate SPSs for each simulated players.
4. Have each simulated player play through simulated scenarios using its SPS to dictate its behaviour while the algorithm monitors it, updates its player model and makes adaptations. Check for mismatched elements at the start of the play session and between every scenario completed. Mismatched elements refer to when the option of an element currently in use does not match the preferred option from the reference player model.
5. Compare the variables of the output player model of each simulated player with their their counter parts in its reference player model and false player models.

5.4 Decisions for testing

Along with the procedure above, five high level decisions had to be made to carry out the system testing. First, a test sample size of 20 simulated players was chosen. As a result, 20 SPSs, reference player models, and false player models were generated. Second, it was decided that each simulated player be put through five simulated educational scenarios.

Third, as explained in Section 4.2.4, the algorithm requires scenario benchmark data (SBD) to function. The specific SBD data needed for each of the five simulated scenarios, included the baseline number of interactions taken to complete each scenario (SBD.2) and the baseline time taken to complete each scenario (SBD.3). For the sake of uniformity, SBD.2 were given values of 100 interactions and SBD.3 were given values of 100 seconds for all simulated scenarios. This is illustrated by Table 5.1.

Table 5.1: Baseline number of interactions and time to complete each scenario

Baseline	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**	**Scenario 5**
SBD.2	100	100	100	100	100
SBD.3	100	100	100	100	100

Fourth, since Minecraft was chosen as the platform for the user evaluation, see Section 6.3.1 for full details, it was also deemed a suitable platform to carry out the system testing. This was because Minecraft already incorporated the tools for player monitoring and an NPC system that could express simulated players as NPCs and dictate their behaviour using SPSs.

Fifth, the Minecraft game world that the NPCs were to interact in would need to start with an option of each of the four preference elements being active, along with an ordered list denoting the next options to adapt to when suitable. The following shows the set default ordered lists of options for each element, with the option in bold being the starting one:

Level style - [**memory**, navigation, quick thinking, problem solving, strategic thinking]

Communication style - [**formal**, informal]

Motivator style - [**verbal hint**, direct hint, visual hint, encouragement]

Feedback style - [**audio**, visual, world]

5.5 Data generation

This section explores the probability model and methods used to generate the reference player models, false player models and SPSs in accordance with step 3 of framework for system testing by simulation.

5.5.1 Reference player model

A reference player model is a completely accurate player model of a player or simulated player. When applying the player model specification from Section 4.2.1 to the RAES, reference player models took the form of a table denoting players' preferred options (PM.1B) for the four preference elements. The preferred option for each element was generated randomly for each simulated player. The options for each element included:

Level style. Memory, navigation, quick thinking, problem solving or strategic thinking.

Communication. Formal or informal.

Motivator. Direct hint, visual hints, verbal hints or encouragement.

Feedback. Audio, visual or world.

The PM.1A variables for the preference elements and the performance elements had to be omitted from the reference player models. These were omitted as generating values for them could cause two problems. First, is that following the generated values would dictate the values in the SPS. It was reasoned that generating values for PM.1A would essentially dictate the values contained in the SPS. This would defeat the purpose of system testing as any desired output can always be obtained if input is manufactured specifically to obtain that output. Second, assuming that these values and the SPS were generated independently, there existed the possibility that these values in the output and reference player models could never match, even if the algorithm was functioning correctly. This is because the values in the output player model would be calculated from the SPS, while the values in the reference player model would be generated randomly.

Therefore, only preference elements were included. Each simulated player's reference player model has a randomly generated preferred option for each element. Table 5.2 shows an example of a reference player model.

Table 5.2: Example of a reference player model.

Element	**Preferred option**
Level style	Quick thinking
Communication	Informal
Motivators	Visual hint
Feedback	World

5.5.2 False player model

A false player model is a completely inaccurate player model of a player or simulated player. It uses the same structure as a reference player model. However, it contains the options of each element that a player does not prefer. Thus, generating a false player model involves including every option for each element that is not in its corresponding reference player model. Table 5.3 shows an example false player model generated using Table 5.2

Table 5.3: Example of a false player model

Element	**Preferred option**
Level style	Memory, Navigation, Problem solving, Strategic thinking
Communication	Formal
Motivators	Verbal hint, Direct hint, Encouragement
Feedback	Audio, World.

5.5.3 Simulated player scripts generation

The purpose of an SPS is to serve as a script to make simulated players behave like actual players. The behaviour of players that are relevant to the algorithm are denoted by the two player metrics identified. Due to these player metrics applying to simulated players in the same way as actual players, the two variables generated for the SPS mirrored the two player metrics used. These include:

SPS.1. The number of interactions taken to complete each scenario.

SPS.2. The amount of time taken to complete each scenario.

When considering methods to generate values for the SPSs, although using completely random values for each variable in the SPS was an option, there was a chance that it could result in SPSs that do not reflect the behaviour of an actual player. This issue was remedied by using a formula which includes two additional variables to generate values for the SPS that more accurately mirror those of players.

The first variable is the *player proficiency multiplier*. Its purpose is to account for the varying number of interactions and time different players could take to complete scenarios. The player proficiency multiplier is an inversely proportional multiplier denoting the percentage of the number of player interactions and time taken to complete a scenario compared to the baselines SBD.2 and SBD.3 respectively. Vittori et al. (2000) explain that individuals with a high proficiency have a more directed approach to learning which enables better and quicker understanding of the educational content. Conversely, those with low proficiency have an undirected approach and experience the exact opposite effect, having greater difficulty and take more time to learn. Those with average proficiency fall somewhere in between. Applying this concept to the two player metrics represented by SPS.1 and SPS.2, the higher the player proficiency, the lower the player multiplier and the lower the player metric taken to complete a scenario and vice versa. Thus, the following classifications of player proficiency multipliers were taken from Vittori et al. (ibid.):

High proficiency. Players take between 50% to 80% the baseline number of interactions and time to complete a scenario. This translates to the player proficiency multiplier having a range of 0.5 to 0.8.

Average proficiency. Players take between 80% to 140% the baseline number of interactions and time to complete a scenario. This translates to the player proficiency multiplier having a range of 0.8 to 1.4.

Low proficiency. Players take between 140% to 200% the baseline number of interactions and time to complete a scenario. This translates to the player proficiency multiplier having a range of 1.4 to 2.0.

Each simulated player had its own set of player proficiency multipliers for interactions and time for each of the five scenarios. This totaled to 10 player proficiency multipliers per simulated player, five multipliers for interactions and five for time. The values of each multiplier were randomly generated between the ranges above, based on the classification of player proficiency. When deciding the number of simulated players in each category to generate SPS data for, the normal distribution model was used. Hence 15% of simulated players were of low proficiency, 60% would be of average proficiency and the remaining 15% would be of high proficiency. Given the sample size of 20 simulated players, this translated to 3 low proficiency players, 14 average proficiency players and 3 high proficiency players.

The second variable quantifies the effect of the option(s) of preference elements being used not matching the simulated player's preferred options, has on the number of interactions and time taken to complete a scenario. Blickle (1996) conducted several studies that showed that students attempting to learn in a manner that they were unaccustomed to, could take between 20% to 50% longer to learn content compared to students who were more accustomed to that manner. Extrapolating this finding to situations where the options of elements are not the preferred ones, it was decided that each mismatched element would cause an increase in the number of player interactions and time taken to complete a scenario by 20% to 50% of baselines SBD.2 and SBD.3 respectively. This increase would be quantified by the *penalty multiplier*, which can take a value from 0.2 to 0.5.

Thus, each simulated player has its own set of player penalty multipliers for interactions and time for each of the four preference elements. This totaled to eight penalty multipliers per simulated player, four multipliers for interactions and four for time. A random value between the specified range was generated for each of the corresponding multipliers for each of the mismatched elements. The penalty multiplier has a value of 0 when the options of the preference elements currently in use match the simulated player's preferred options. The four penalty multipliers for interactions and time were added together to form the total penalty multipliers for interactions and time respectively. This was done to cumulatively stack their effects on number interactions and time taken to complete a scenario.

The effects of both types of multipliers were reasoned to stack on one another due to each quantifying changes to the number of interactions and time taken to complete a scenario. Thus, they would be added together in the final calculation of SPS.1 and SPS.2. This is displayed in the following formulae:

$$SPS.1 = SBD.2 \times (player.proficiency.multiplier.interactions + total.penalty.multiplier.interactions) \tag{5.1}$$

$$SPS.2 = SBD.3 \times (player.proficiency.multiplier.time + total.penalty.multiplier.time) \tag{5.2}$$

An example of utilising these formulae on an averagely competent player playing a scenario with two mismatched elements could yield:

$$\text{SPS.1} = (100)\ \times((0.95) + (0.24 + 0.3 + 0 + 0)$$

$$\text{SPS.2} = (100)\ \times((1.04) + (0.41 + 0.2 + 0 + 0)$$

Tables 5.4 and 5.5 show an example of a set of proficiency multipliers and penalty multipliers respectively, that are used to calculate the values in the SPS.

Table 5.4: Example set of player proficiency multipliers for each scenario generated for a simulated player.

Proficiency multiplier	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**	**Scenario 5**
For interactions	1.12	1.10	0.98	1.05	1.02
For time	1.23	0.91	0.88	1.14	1.09

Table 5.5: Example set of penalty multipliers for each element taken at the start of a play session where two elements are mismatched.

Penalty multiplier	**Level style**	**Communication**	**Motivators**	**Feedback**
For interactions	0	0.29	0.24	0
For time	0	0.36	0.25	0

Table 5.6 displays an example SPS calculated using formulae 5.1 and 5.2 with the data from Tables 5.1, 5.4 and 5.5.

Table 5.6: Example SPS for a simulated player

SPS variable	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**	**Scenario 5**
SPS.1	165	110	98	105	102
SPS.2	184	91	88	114	109

5.6 Results and conclusions

With Minecraft's NPC system used to simulate players who were monitored by the algorithm, player models with the structure explained in Section 4.2.1 for each simulated player were outputted. However, only variables PM.1B would need to be compared to the reference and false player models to determine the accuracy of the algorithm's output, as the reference player models only included a player's preferred options and not the preference rates for each option (PM.1A). This section explores the results of the simulated evaluation and its effects on the rest of the study.

5.6.1 Results

When comparing the algorithm's output player models to the reference player models and false player models, the algorithm's outputs were 65% closer to the reference player models on average. This was due to the variables in the output player models matching the variables in the reference player models with 82.5% accuracy and the variables in the false player models with 17.5% accuracy. This is taken from the average of 100% accuracy with below average proficiency players, 92.8% with average proficiency players and 0% with above average proficiency players when compared to the reference player models. Thus, hypothesis H0 can be rejected. A comparison of algorithm output accuracy and player proficiency is shown in Figure 5.1.

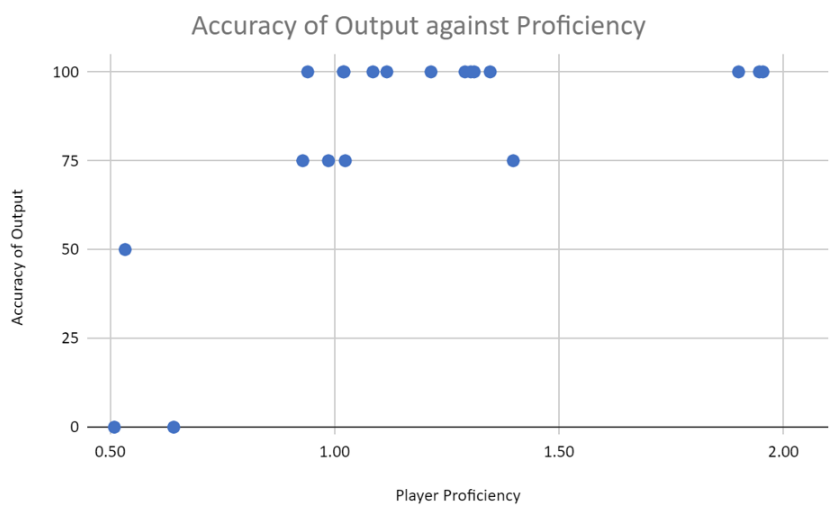

Figure 5.1: Shows the accuracy of the algorithm's output compared to the player proficiency

Concerning the below average proficiency players, Figure 5.2 shows that the algorithm had generated completely accurate player models by the end of the first or second scenario. This was attributed to the high number of interactions and time taken by such a player, thus having the algorithm execute more adaptations. The more adaptations the algorithm executed, the higher the accuracy of its output player model. The algorithm executed an average of 16 adaptations per simulated player for this group.

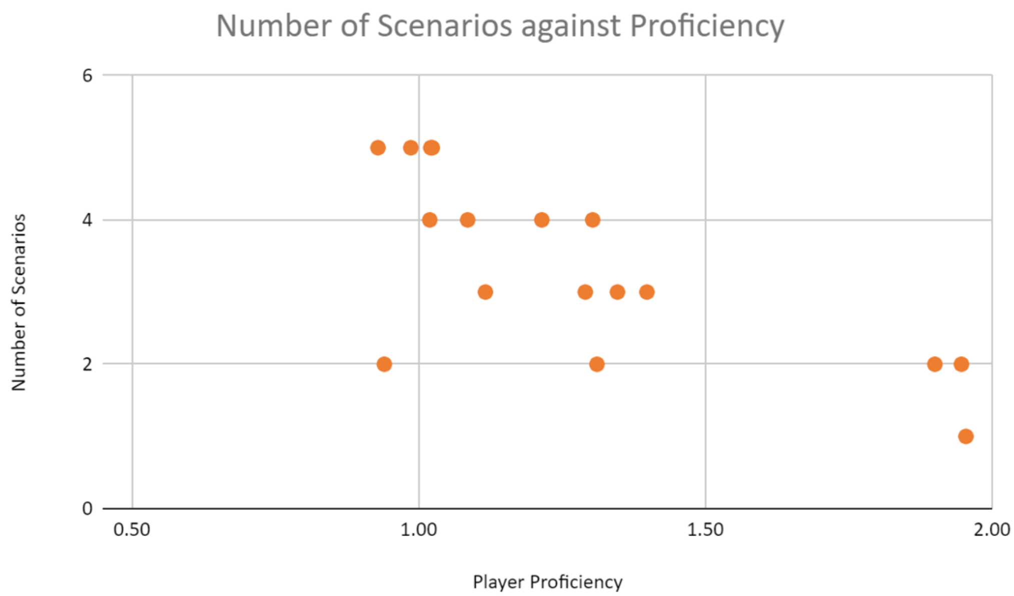

Figure 5.2: Shows the number of scenarios completed before the algorithm's output player model matched the reference player model.

Looking at the averagely proficiency players, the algorithm's output player models matched most reference player models by the end of the fourth objective. However, there were four mistakes in this group. Three of these mistakes were mismatched feedback styles and the fourth was a mismatched level style. The level style mistake was due to that simulated player's number of interactions and time taken to complete scenarios causing fewer adaptations to be executed by the algorithm. Thus, the algorithm did not execute enough adaptations to correctly identify that player's preferred level style. Regarding the feedback style mistakes, it was found that the Minecraft NPCs performed their interactions at uniform time intervals, thus the Adaptability Condition 4 was rarely met. It was theorised that the time between interactions of an actual player would not be as uniform and thus Adaptability Condition 4 would be met at a higher rate. On average, the algorithm executed 9.3 adaptations per player for the averagely proficient group. Hence, the overall lower effectiveness is indicative of the adaptations made by the algorithm.

This is further shown by the above average proficiency players who did not have the algorithm execute a single adaptation. Due to the players of this group being able to progress despite mismatched elements, the algorithm did not have the opportunity to make adaptations and updated accurate player models. Therefore, the algorithm is least effective with players who are highly skilled at learning and/or already familiar with content matter. The algorithm executed 0 adaptations for all players in this group.

The accuracy of the algorithm's output player models were shown to be directly correlated with the number of adaptations executed. Figure 5.3 shows that the number of adaptations executed is directly correlated to the proficiency of a player. Therefore, it can be surmised that the lower the proficiency of a player at the educational content and/or game, the more accurate the output player model will be. Lastly, as the algorithm's output achieved an accuracy of 82.5%, the algorithm possesses the ability to output player models closer to the reference player models than the false player model.

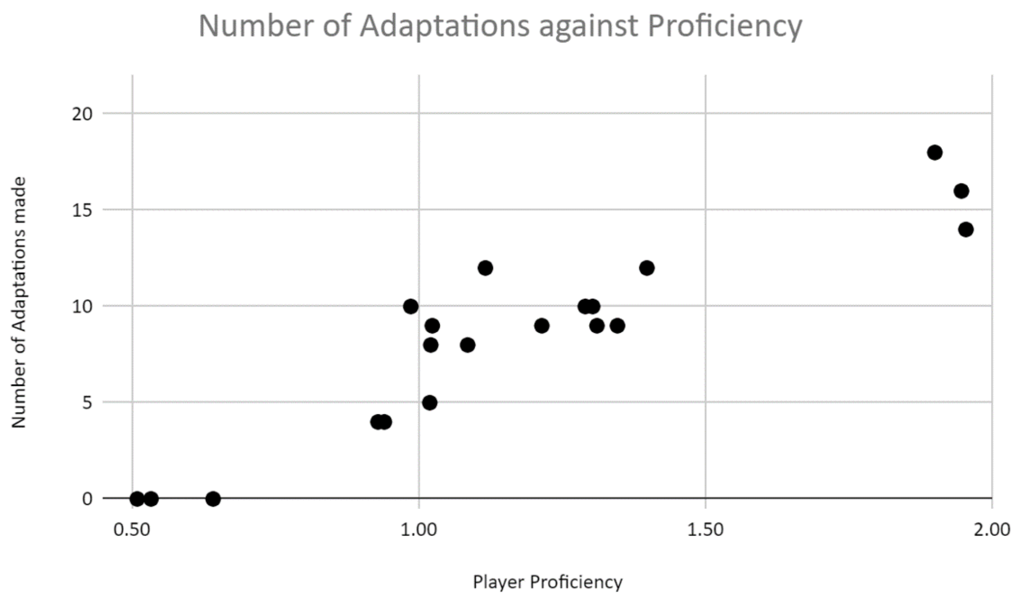

Figure 5.3: Shows the correlation between player proficiency and the number of adaptations made

5.6.2 Conclusions

Looking at the results, it appears that with the above average proficient players, the algorithm's output player models become more accurate over more executed adaptations, which take place over more scenarios. Figure 5.2 illustrates this by comparing the number of scenarios needed to achieve the correct output player model compared to the proficiency of the player. It shows that the majority of averagely proficient players need to play through between three and five objectives before the algorithm outputted accurate player models.

Therefore, assuming a random user test group for the user evaluation, comprising mainly averagely proficient players, educational game content developed for user evaluations should prioritise the number of educational scenarios over the number of adaptable elements to maximise the number of executed adaptations. It is theorised that a minimum of five educational scenarios should be targeted.

In conclusion, as the algorithm's output is 82.5% accurate (92.8% accurate when omitting the above average proficiency players), it can be assumed that the base principle of the algorithm's function is sound. The following chapter will explore and detail the user evaluation.

Chapter 6

User evaluation

This chapter focuses on evaluating the effectiveness of the algorithm at enhancing the learning gains of users when playing an education game. The evaluation of the algorithm involved studying a group of students at the appropriate educational stage (13-16 years old) as they played a Minecraft-based educational game teaching critical thinking skills, with and without adaptation. The evaluation process includes defining clear goals and hypotheses, selecting appropriate educational content, adopting the optimal teaching strategy, reviewing and identifying the most effective experimental design, defining a measure of success, the evaluation procedure and drawing conclusions from the results.

6.1 Purpose and aim

The goal of the evaluation was proving that the algorithm enhances the learning gains in users when playing an educational game as opposed to playing the same game without the algorithm. Therefore, this user evaluation was designed to test the following hypothesis:

H_0 = There is no statistically significant difference in learning gains in users who receive adaptive aid and those who do not, while playing an educational game.

H_1 = There is a statistically significant difference in learning gains in users who receive adaptive aid and those who do not, while playing an educational game.

This evaluation was aimed at students aged 13 to 16. This is because the educational content and measures of effectiveness identified, comprised different versions for different ranges of student age. Since the goal of the evaluation was to determine the effectiveness of the algorithm at improving learning gains with any educational content regardless of a player's age, only one age range was necessary. Thus, the student age range of 13 to 16 was selected along with the corresponding educational content and measure of effectiveness for that range.

Furthermore, due to the limited time-frame, resources and scope of this study, certain concessions had to be made with regards to the evaluation methodology. These included choices regarding the teaching strategy used by the educational game and which of the six elements in the RAES had adaptive content developed for them.

Although these choices were reasoned to still produce valid results (see section 6.2.2 and 6.3.2 for justification), it was decided that this evaluation be treated as a preliminary one, with the potential to lead on to a more extensive experiment. This future experiment could utilise a different teaching approach and/or include adaptive content for more game elements when time and resources are available, thus shedding more light on the validity of the algorithm. The following section will explain and justify the evaluation methodology used.

6.2 Evaluation methodology

The evaluation methodology was broken down into the educational content, teaching strategy, experimental design, the measure of success and the recruitment procedure. This section explains the choice and justification for each of the evaluation methodology subsections.

6.2.1 Educational content

As stated in Chapter 1, the selected topic for educational content is critical thinking due to it being a transferable skill, especially when applied to problem solving and making sound conclusions. Critical thinking is therefore a desirable skill to teach to all age groups. This section involves detailing and selecting the subset of the critical thinking syllabus that would serve as the educational content for this evaluation.

Lipman (1988) states that students using critical thinking practice self-correction, objectivity, consistency and reflection upon their thoughts. This translates to students being able to justify their thinking using evidence. He explains that teaching critical thinking is accomplished by asking appropriate questions, defining information, challenging the validity of statements or beliefs, judging the credibility of sources, and drawing conclusions via deduction. Similarly, Banning (2006), defines critical thinking as a process involving identifying, analysing, appraising and reflecting on information in order to form conclusions. Ennis (1989) mirrors this definition and further details that those versed in critical thinking display 12 behaviours that are absent in those who are not. These include:

1. Seeking a clear definition of the topic/problem.
2. Breaking down more complex topics/problems into simpler sub-components.
3. Seeking precise information to inform decisions.
4. Using credible sources for information.
5. Ensuring that the information remains relevant to the topic/problem.
6. Searching for valid reasons to credit or discredit the information.
7. Identifying and defining all possible positions at the time.
8. Taking the entire situation into account before deciding on a position.
9. Being capable of taking a position or changing a position due to evidence.
10. Being open-minded to new and/or previously unknown information and view points.
11. Assessing new information while keeping the original topic/problem in mind.
12. Exhibiting sensitivity to others' feelings and depth of knowledge when discussing topics/problems.

Examining these definitions shows overlap in the three steps of identifying a problem, finding and analysing credible information relevant to solve the problem, and coming to a conclusion based on information found. Ideally, content for all these sections would be developed. However, as the purpose of this study was not the development of educational content, it was reasoned that developing content for the second step would suffice for determining the effectiveness of the algorithm.

Butterworth and Thwaites (2013) delve into further detail with identifying and determining the credibility of information. They further subdivide the topic into the identification of information and the analysis of the credibility of information. Concerning the identification of information, they explain that there are four ways of categorising information:

1. Direct information. This pertains to objective facts that can be calculated, derived or directly observed.

2. Testimony. This is information received from a person either verbally or written.

3. Circumstantial evidence. This refers to information of all kinds that is indirectly relevant to the problem, which is used for inference and thus is incapable of completely proving a conclusion.

4. Corroborative information. This is external evidence that might further support or weaken inferences made using circumstantial evidence

Both Butterworth and Thwaites (ibid.) and Thomson (2001) split analysing the credibility of information into the following five steps:

1. Trusted source. This refers to checking that the source of the information is a trusted one. Trusted sources pertain to people, books and/or websites that can be viewed as having knowledge and expertise relevant to the problem.

2. Neutrality. This involves assessing the information given for neutrality to determine its trustworthiness. Facts are easier to assess for neutrality due to the binary nature of being true or false. Assessing opinions for neutrality is more ambiguous due to biases individuals may posses. Recognising and considering biases lowers the neutrality and trustworthiness of opinions.

3. Vested interest. This pertains to doubting the source's neutrality due to potential gain that the source could experience from holding a biased point of view. For example, when considering a criminal trial, family members are not permitted to serve as members of the jury as they may have a vested interest to acquit their relative, irrespective of innocence or guilt.

4. Relevance of the evidence. The is ensuring that any evidence considered has relevance to the problem. While evidence from a trusted and neutral source might be reliable, it may not apply or be useful when concerning situations that has nothing to do with.

5. Corroborative evidence. This involves assessing any corroborative evidence found for credibility using the former four steps above. Non-credible pieces of corroborative information should not be used to strengthen the credibility of other information.

To summarise, the identification and credibility assessment of information served as the educational content for the evaluation of the algorithm. This topic can be split into the identification of information and analysing the credibility of information. Concerning the identification of information, four types of information were identified. Regarding the analysis of information for credibility, five steps were identified.

Therefore, for the purpose of developing educational content for the educational game in this evaluation, learning scenarios incorporating the four types of information and the five steps of credibility analysis were made. Please see Sections 6.3.3.1 and 9.1 for full specifications of educational scenarios.

6.2.2 Teaching strategy

Two strategies for teaching critical thinking were identified, programmatic and instructional. Bartlett and Cox (2002) explain the programmatic approach as the blending of critical thinking into the background with the student's focus on a separate subject. They implemented this approach via a 12 month study involving 26 nursing students who had critical thinking combined with their nursing syllabus. Scott, Markert, and Dunn (1998) conducted a similar study over 3 years involving 68 medical students. Both studies showed a significant improvement in scores using standardised critical thinking tests in the majority of participants.

The instructional approach teaches critical thinking as a stand alone subject. A survey of multiple studies revealed different methods of implementation of the instructional approach. These include problem based learning (Hesterberg, 2005), computer-assisted instruction (Yang, 2008), concept mapping (Wheeler and S. Collins, 2003), scenario-based exercises (Sandor et al., 1998) and the question approach (Thompson, 2009). Concerning the effectiveness of instructional approaches, Wheeler and S. Collins (2003) adopted the concept mapping instructional approach to teach critical thinking to nursing students over a 16 week period. They achieved positive results by posting significant growth in critical thinking skills in their participants. However the positive effects were not as pronounced as the programmatic approach utilised by Bartlett and Cox (2002) and Scott, Markert, and Dunn (1998). Another survey of instructional approaches conducted by Behar-Horenstein and Lian (2011) revealed that no single instructional method proved effective in every case or more effective than the other methods.

When considering which of the two teaching approaches to utilise, the programmatic approach yielded more significant improvement learning gains when compared to the instructional. However, the programmatic approach requires a longer duration of study as well as focus towards the combination and teaching of a separate subject. Doing so would require significant time (a year in the case of the works of Bartlett and Cox (2002)) and resources.

Although not as pronounced as the programmatic approach, the instructional approach has also proven to produce significant growth in learning gains. As the goal of this evaluation was to prove that the algorithm significantly enhances the learning gains in users when playing an educational game, the instructional approach was reasoned to be an appropriate option. Additionally, utilising the instructional approach also requires significantly less time and resources. Lastly, it should be noted that Campbell and J. Stanley (1963) describe utilising the instructional approach as "worth doing where nothing better can be done".

While it would be ideal to utilise the programmatic approach, the limited time-frame, resources and scope of this study made doing so less viable. Given the three reasons above and the notion that this evaluation was treated as a preliminary one with room for future expansion, it was reasoned that the instructional approach would be adequate at producing significant learning gains, be within the time and resource scope of this study, and be worth doing overall. Hence, the instructional approach was utilised for this evaluation. More specifically a combination of computer assisted instruction, problem based learning and scenario based exercises was devised as educational games have the capability to seamlessly blend all three.

Moving forward, educational scenarios covering all aspects of the identification and credibility assessment of information. These scenarios were modelled after the examples found in textbooks designed to teach analytical and critical thinking by Butterworth and Thwaites (2013), Ennis and Millman (1989) and Thomson

(2001). It was theorised that students playing through all the scenarios would see an increase in their ability to identify the different types of information and judge their credibility.

6.2.3 Experimental design

With the educational content and teaching strategy decided on, the next step was to review and adopt an appropriate experimental design. Campbell and J. Stanley (1963) detail three approaches to user experiment design: the pre-experimental design, the quasi-experimental design and the true experimental design. All designs involve administering a pretest, intervention and post-test to the participants, with the interventions effectiveness determined via comparison of the post and pre-intervention test scores. However, they differ in following ways.

The pre-experimental design involves a one-shot intervention but excludes a control group and the random assignment of participants. A one-shot intervention is defined as an intervention and measurement of effectiveness that only occurs once per participant. The lack of a randomly assigned control group increases the difficulty in determining the effectiveness of the intervention, making this method vulnerable to internal criticisms of validity. It should be noted however, that Campbell and J. Stanley (ibid.) describe this method as “worth doing where nothing better can be done”.

Quasi-experimental designs include a time-series intervention experiment and a non-random non-equivalent control group. A time-series intervention is defined as one where each participant undergoes multiple interventions over an extended period of time with the effectiveness of each intervention measured and tracked. This method is less vulnerable to validity concerns due a lack of a non random control group. True experimental designs include one of the types of interventions, more often time-series than one-shot, and a completely randomly assigned control and experimental group. Although this method nullifies the internal invalidity of lacking a control group, it is the most time and resource heavy approach and is therefore not always implementable by researchers.

When examining the three designs, the lack of a randomly assigned control and experimental group serves as the strongest of criticism of validity. As the experimental treatment in question is the activation of the adaptive algorithm, randomly assigning participants to a control and experimental group was relatively straightforward. All participants played educational scenarios in Minecraft with the control group receiving no adaptive assistance and the experimental group receiving full adaptive assistance. Boyadjian-Samawi (2006) notes that time-series interventions lasting 5 months and longer produce significant gains of critical thinking skills in up to 70% of participants. However, resource, logistical and public health concerns rendered the one-shot intervention the only tactic available at the time of this study. Furthermore, it was theorised that focusing on a subset of critical thinking as opposed to the entire syllabus would allow for significant learning gains to be detected in a shorter intervention period.

To summarise, the evaluation method entailed combining a one-shot intervention and randomly assigned control and experimental groups, with the experimental treatment being the activation of the adaptive algorithm.

6.2.4 Measure of effectiveness

The next aspect of the evaluation methodology involves selecting an appropriate measure to determine the effectiveness of the intervention. As the effectiveness of the intervention involved identifying differences in learning gains, it was reasoned that standardised tests would be the most effective measure. This is due to the reliability and validity of standardised tests at determining the knowledge a person possesses on a topic, as they were developed and evaluated by experts in their respective field. Therefore, the effectiveness of the intervention was measured by having each participant take a pre and post intervention test, and comparing the difference in scores for statistical significance.

Behar-Horenstein and Lian (2011) identified three standardised test methods for critical thinking. These include the Watson-Glaser Critical Thinking Appraisal (WGCTA), the California Critical Thinking Skills Test (CCTST) and the Cornell Critical Thinking Tests (CCTT). After conducting a review of 42 studies that used at least one of the three test methods, they found that all studies using CCTT detected statistically significant differences during the post test or between the experimental and the control groups. The results of WGCTA and CCTST were mixed. But, they were unable to draw concrete conclusions from comparing the three measures due to a significantly lower number of studies using CCTT.

However, as the focus of this study was to determine the effectiveness of the algorithm, and not the effectiveness of each testing method, the most statistically effective test method was chosen to lessen the probability of failure due to ineffective measurement. Thus, a subset of CCTT was chosen as the measure of learning gains. This is due to the educational content being a subset of the syllabus CCTT was designed to test for.

CCTT is split into the X and Z form. Ennis, Millman, and Tomko (2005) explain that Form X is for students aged 8 to 16, while Form Z is for advanced/gifted secondary school students, undergraduates, graduate students and adults. Since the target users of this evaluation were secondary school students ages 13 to 16 and no reliable method of determining each student's proficiency at the educational content before the pre-test, only Form X was used. Form X is a 50 minute multiple choice test with 72 questions split evenly between the four topics of assumptions, induction, deduction and credibility. It employs a scoring penalty for incorrect answers to prevent students guessing the right answer. The credibility section of CCTT was used as it mapped directly to the chosen educational content.

Therefore the Form X test developed by Ennis and Millman (1989) could be condensed to 18 questions exclusively testing the credibility topic to be completed under 15 minutes. This adapted Form X test has a maximum score of 18 and a minimum score of -18, a range of 37 marks in total, and served as the pre and post intervention tests for the participants.

Although the option to employ qualitative measures of learning gains such as interviews was available, it was decided against for two reasons. Firstly, it was a concern that the inclusion of an interview to the pre-test, video game session and the post-test methodology would add unnecessary burden and stress given the younger age of participants (13 to 16 years). Secondly, as a subset of the critical thinking syllabus was assessed, it was reasoned that the quantitative measure would suffice.

To summarise, adapted CCTT X Form tests served as the pre and post intervention tests for the participants. The difference between the post and pre-test scores of each participant was taken as the measure of learning gains.

6.2.5 Recruitment procedure and ethical issues

In theory, the recruitment procedure for a user evaluation regarding software would be to make members of the targeted audience aware of the evaluation, establish a line of contact with them, explain the purpose and expectation of the evaluation, obtain their consent to participate and then run the evaluation session. However, the targeted audience of this evaluation being students aged between 13 to 16 presented several ethical issues due to them being younger than 18 years old.

Firstly, as the participants were classified as a vulnerable group (being minors), it was deemed that approaching them directly to establish interest in participating in the evaluation was inappropriate. This was remedied using the following three methods. The extra curriculum departments of two schools near myself (west London) were emailed information regarding this evaluation which they then forwarded to prospective parents, who could then contact me directly. Parents that I personally knew who had children within the desired age limit were also approached directly by myself. Lastly, a request to spread the existence of this evaluation through word of mouth was issued to the parents that I came in contact with.

Secondly, the participants were unable to legally consent to participating in this evaluation. Therefore, consent was sought from the parents and/or legal guardians of the participants. Furthermore, the aim and methodology of the evaluation was explained to the parents first, to alleviate any potential worries and/or grievances, before explaining the same information to the participants.

Thirdly, building on the fact that the participants were classified as vulnerable, it was deemed inappropriate for them to be left alone with a stranger (myself) while participating in the evaluation. Thus, it was a requirement that all evaluation sessions be conducted with at least one parent and/or guardian present throughout. This requirement was agreed upon by all parties before proceeding and followed for every evaluation session conducted.

Fourthly, as this evaluation was conducted during the COVID-19 pandemic in 2021, further stipulations concerning health and safety were observed. This included conducting the evaluation sessions in public open-air areas such as parks and cafes with outdoor seating, and myself, the participants and their parents wearing face masks at all times.

To summarise the overall recruitment procedure used, the parents and/or legal guardians of potential participants were contacted. This was accomplished using a mixture of mass emails forwarded by extra curriculum departments of two schools in west London to parents, personal contacts and word of mouth. All lines of communication were established between myself and the parents instead of the participants. The purpose of the evaluation along with a brief outline of what was expected from participants was explained to parents first before relaying the same information to participants. When both parties were agreeable to the evaluation, consent was obtained from parents via a consent form. Participants were also given consent forms for the purpose of making them feel more included in the procedure. The evaluation sessions were then conducted in public open-air locations with at least one parent present at all times. Ethical approval for this procedure was applied for and granted by the King's College ethical department and was adhered to throughout.

6.3 Evaluation platform and content

This section focuses on the platform (game) paired with the algorithm for the evaluation, the adaptive content developed and the educational scenarios that were adapted to teach the chosen content.

6.3.1 Evaluation platform

As explained in Chapter 2, educational games have the primary purpose of educating and the secondary propose of entertaining their players. Therefore the educational game chosen to be paired with the algorithm must be capable of achieving both purposes. When considering the primary purpose of educating players, the functionality of the algorithm dictates that the educational game chosen to be paired with the algorithm fulfills the following compatibility requirements:

Compatibility Requirement 1 (Player modelling) *The educational game must posses the ability to create and update player models in order to determine the course of adaptation(s) to take that could improve players' learning when any of the adaptation conditions are met. The educational game must possess the ability to run custom/modified code along with its original processes. As long as a game possesses this functionality any form of player modeling may be utilised.*

Compatibility Requirement 2 (Educational scenarios) *The educational game must possess the ability to execute educational scenarios to teach the educational content. Being the selected teaching mechanism for this study, the structured design of educational scenarios is necessary for the function of adapting elements that are to be adapted at predefined points and for the checking of Adaptation Condition 5.*

Compatibility Requirement 3 (System clock) *The educational game must possess the ability to time the player. As explained in Chapters 3 and 4, timing the player is one of the performance metrics chosen for this study and is integral to Adaptation conditions 2 and 3.*

Although the algorithm was designed to be incorporated with educational games, it was noted that there could exist entertainment games that meet or at least possess the capability to meet all three Compatibility Requirements. Hence moving forward, it was reasoned that entertainment games that could be modified to fulfill the Compatibility Requirements, could also be considered to be suitable games to pair the algorithm with and by extension serve as the evaluation platform for this study. The inclusion of entertainment games as candidates for the evaluation platform meant that there existed an overwhelmingly vast pool of games that could have been used. Thus, the additional factor of a game having a high entertainment value was included in the selection process. This approach was taken in an effort to maximise the number and/or willingness of participants to take part in this evaluation.

As established in Chapter 2, a game most effectively educates players when it is also entertaining them. However, drawing up requirements for entertaining players was deemed to be more difficult. This is due to the concept of finding a game entertaining to be highly subjective amongst individual players. But, it was reasoned that games that were popular with their targeted audience must possess some degree of entertainment value to have achieved their popularity in the first place. Therefore, it was decided that the best approach to selecting a game to serve as the evaluation platform would be to select a popular entertainment game that could also fulfill all three Compatibility requirements. Sites such as Newzoo (2021) identified the most popular games suitable for ages 16 and below as the following (in descending order):

1. **Minecraft**. An open world, exploration, crafting and adventure game made by Mojang (2011) that uses PCG to generate its game world and objects and may be played online and offline.

2. **Fortnite**. An online open world game centred around player combating one another and the environment either competitively or cooperatively, developed by Epic (2017).

3. **Roblox**. An online game platform and game creation system developed by Roblox (2006) that allows players to program games and play games created by other players.

4. **The Sims 4**. A social simulation game where players control the lives of NPCs in a fictional world.

All four games met and/or had the potential to meet the three Compatibility requirements. But, addressing Compatibility Requirement 2 with Fortnite was deemed to be disproportionately challenging as opposed to the other games. This was due to nature of Fortnite being geared towards combat, which would require a complete redesign to alter. A complete redesign was considered wasteful allocation of time a resources due to the presence of other popular games that could have Compatibility Requirement 2 more easily incorporated.

Between Minecraft, Roblox and The Sims 4, Minecraft was identified as the most suitable platform for the evaluation for the following three reasons. Firstly, Minecraft already possessed educational functionality through its expansion "Minecraft educational edition", thereby fulfilling Compatibility Requirement 2 to a greater degree than the other three. Secondly, between all the popular games identified, it stood as the most popular one. This meant that Minecraft could potentially attract the most number of participants when compared to the competition. This fulfilled the purpose of selecting an entertaining entertainment game in the first place. Thirdly, the personal familiarity of the researcher (myself) with playing and creating custom content for the Minecraft platform was reasoned to allow for more time and resources to be put towards developing adaptive content and educational scenarios as opposed to the compatibility requirements. This would allow the emphasis of the evaluation to remain on the research goal of determining effectiveness of the algorithm and adaptive content on improving players' learning, as opposed to software challenges of making games compatible with the algorithm. Thus, Minecraft was used as the evaluation platform.

However, two custom modifications were made to the base Minecraft game to make it fulfill Compatibility Requirement 2. Firstly, the feature to have conversations and dialogues with NPCs was added as a medium to deliver information and create educational scenarios as seen in Figure 6.1. Thus, educational scenarios and their relevant NPCs could be given a backstory and purpose, thereby incorporating an enacted narrative structure to each educational scenario based on the principles of narratology.

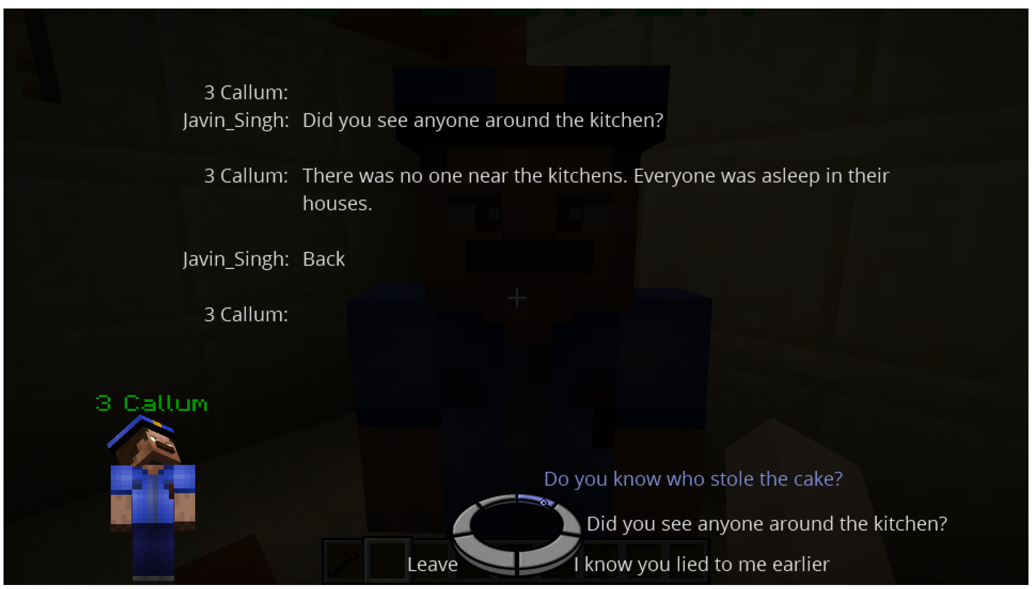

Figure 6.1: A screen shot of the game showing dialog with NPCs feature.

Secondly, the ability to add specific objectives with clear paths to accomplishing was added. Figure 6.2 illustrates this. This was done for both the function of the algorithm and to ensure that players learn the intended content by attempting to complete the objectives. The combination of these functionalities allowed for educational scenarios with storied NPCs and clear objectives to be created.

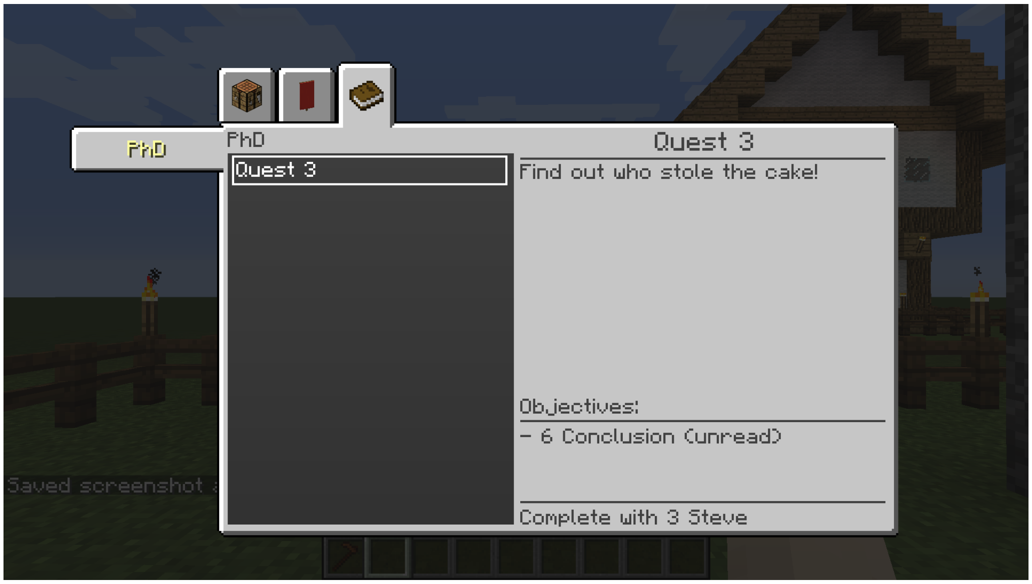

Figure 6.2: A screen shot of the game showing an objective given to a player.

6.3.2 Adaptive content

As explained in Chapter 5, the accuracy of the algorithm's output player models increases with more adaptations made over more educational scenarios played. The algorithm was able to output player models with 82.5% accuracy when simulated players completed five educational scenarios. As accuracy of player models is theorised to cause adaptations that have a more pronounced positive impact on learning gains, it was reasoned that at least five educational scenarios should be made for user evaluation. By the end of the development period, six educational scenarios were developed.

Of the six elements in the RAES, four were given adaptive content. The level style and communication elements were omitted. The level style was limited to the problem solving style due to the nature of the educational content chosen. This is because using techniques involving quick thinking, memorisation, navigation and planning are not credible ways to teach the concepts of credibility and accuracy of information based on the scenarios described by Butterworth and Thwaites (2013), Ennis and Millman (1989) and Thomson (2001).

Secondly, the communication element was omitted due to the preliminary nature of this evaluation, and time and resource limitations (explained in Section 6.1). The time and resources required to convert all the lines of dialog present in the game were not available. The omission of the level style and communication elements meant that any conclusions drawn from this evaluation would be regarding a partial model of the algorithm.

However, the remaining four elements that were included represented a balance between preference and performance elements, active and passive adaptability, and elements that can be adapted at any point during a scenario and predefined points during a scenario. Thus, adaptation of the motivators, feedback, level difficulty and time pressure elements were reasoned to have the potential to have a positive effect on players' learning and provide suitable evidence for this preliminary evaluation. Furthermore, it was reasoned that the effects of a full model of the algorithm would possibly be more pronounced versions of the effects of a partial model. Hence, the evaluation of this partial model could provide insight and basis for the potential effects of the full model on players' learning.

Adaptive content for the following four elements were developed in accordance with the survey explained in Chapter 3:

1. Feedback - players can learn about the effects or the lack of effects of their actions on the game world through feedback. These effects could be an indication to a player if he/she is on the right track to learning the educational content and progressing a scenario. Feedback can be given in three methods; audio, visual and world.

    (a) Audio - any NPC that a player would interact with would make a grunting sound as an acknowledgement of the interaction. Furthermore, NPCs would also make a deeper and more irritated grunt if the player makes the identical interaction five times without progressing the scenario to signify discontent. This method was arrived at in an attempt to mirror the sounds that people often make to further express themselves while having conversations. The irritated grunt was devised as a method to mirror an auditory expression of irritation that a person might make when having the same dialog repeated to him/her multiple times during a conversation.

    (b) Visual - NPCs bob their head up and down when players interact with them with a neutral facial expression. NPCs' facial expression would change from a neutral to an upset one if the player makes the identical interaction five times without progressing the scenario. In the same vein as the audio feedback style, it was reasoned that an NPC bobbing his/her head when interacted with would mirror the head movements that people often make while having conversations. The facial expression changing to an upset one was devised as a natural response most people have when being bothered repeatedly.

    (c) World - when a player interacts with an NPC, the illumination level around that NPC will increase to emulate the effect of highlighting that NPC. If the player makes the identical interaction five times without progressing the scenario, it will start to rain and thunder to signify that the player is doing something wrong. Highlighting the NPCs and game objects being interacted with was taken from entertainment games such as Resident Evil 4 (Capcom, 2005), Metro exodus (4AGames, 2019) and Crystal island (Wang et al., 2017) that use the same mechanism to indicate what player are and/or could be interacting with. The effect of rain and thunder were used as they traditionally symbolise something negative and/or going wrong in media such as games, movies and television series.

2. Level difficulty - refers to how challenging the player's current objective is.

    (a) Referring back to Section 2.4.1, there exist two methods to adapt the level difficulty element. First, is adapting the amount of aid given to the player during a challenge. Second, is adapting the severity of the challenge(s). As the level difficulty element is one that is adapted at predefined points during a scenario, it was reasoned that some player's might struggle to reach said predefined points and thus never experience the potential benefit of this element's adaptivity. The second method of adapting the level difficulty element was believed to have the potential to exacerbate this issue. As not being able to learn and progress would defeat the purpose of the algorithm and education games in general, the first method of implementation was taken instead. Hence, the number of interactions a player could commit before being offered assistance was adapted between each scenario based on the formula detailed in Chapter 4.

3. Motivators - individual players will gravitate towards different forms of encouragement and support. This is split into direct hint, visual hints, audio hints and encouragement.
   (a) Direct hint - the player is given a written message explaining either the NPC to interact with or the area to go to to progress the scenario.
   (b) Verbal hint - the player will receive a message describing the mindset they should adopt in order to figure out the next step to progress.
   (c) Visual hint - the path leading to the area and/or NPC that they player needs to go to/interact with will be highlighted red for 20 seconds.
   (d) Encouragement - the player is given a motivational quote and encouraged to keep trying.
4. Time pressure - refers to the extent to which the player is pressured to complete his/her objective within a specific time limit.
   (a) It was reasoned that giving players a time limit that would result in the failure of an educational scenario if crossed would be counter productive to the primary purpose of educating players. This is due to the idea that players who crossing the time limit are struggling to learn and/or apply the education content. Hence, the failure and potential restarting of an education scenario might frustrate and lower motivation in players who are already experiencing difficulty to try again. Thus in a similar vein to the method used to make the level difficulty element, it was decided that the amount of time a player can spend on a scenario before being offered assistance would be adapted between each scenario based on the formula detailed in Chapter 4.

Below are details of the educational scenarios developed and the content that served as source material.

6.3.3 Educational scenarios

Each educational scenario developed took place in a custom built area within a Minecraft world. Each area comprised varied building structures, unique NPCs, problems, narratives and educational content. Figure 6.3 showcases an aerial view of all six areas that each educational scenario took place in. These scenarios followed the structure of the four educational stages described in Section 2.2.1.2.

Figure 6.3: A screen shot of the game showing all six areas that each educational scenario took place in.

The problems, narratives and educational content of all scenarios developed were adapted from the sections in Ennis and Millman (1989), Butterworth and Thwaites (2013), and Thomson (2001), that were aimed at teaching assessment of credibility of evidence. These sections describe example scenarios from the point of view of a character who encounters an issue and interacts with other characters to learn the solution. These comprise Educational Scenario Stages 2, 3 and 4 for the educational scenarios developed.

A modification was made to adapt the scenarios from the literature to function more like educational scenarios in a game. This involved the creation of a helper NPC for each scenario who would fulfill the role of Educational Scenario Stage 1. This included presenting the player with a problem, explaining that the player's task is to resolve that problem, and providing motivation to complete said task. The player must then interact with the other NPCs in the area to learn the educational content to resolve the issue. This was done to provide a mechanism to explain the educational content and mistakes that the player could make using the helper NPC as the medium.

Concerning the values of the baseline number of interactions (SBD.2) and baseline times (SBD.3) from the scenario benchmark dataset; although it was explained previously these values could be set with the use of taking the average of play testers, play testing was not possible. This was due to the COVID-19 restrictions at the time of this evaluation. Thus the following alternate method was used. Using the design specification of one of the scenarios, the developer estimated the number of interactions and amount of time an averagely competent person aged between 13 to 16 years could take to complete that scenario. The estimation was done using the developer's eight years of experience playing Minecraft and 24 years of experience playing educational games and entertainment games in general. Next the developer played through that scenario himself. The estimates of the number of interactions and time taken to complete that scenario were roughly half that of the developers performance. Building on this, it was reasoned that both baselines for each scenario could be set by doubling the values that the developer took to complete each scenario himself.

Below is the specification of one of six educational scenarios developed. This includes an explanation of the educational content, a high level description of the events in the scenario, a figure showing the sequence of events and dialog options, all the possible dialog options, and a map of the in game area the scenario takes place in and the placement of NPCs. The labels in the figures refer to the labels of each dialog option detailed in their respective tables. Note that all human NPCs were given names in the game. All references to NPCs in all lines of dialog were replaced with the respective name of the NPC in the final build of the game. The specifications of the other five educational scenarios can be found in Section 9.1.

6.3.3.1 Scenario 1

Scenario one is aimed at teaching players to search for trusted sources when attempting to look for information. This scenario is based on a scenario described by Ennis and Millman (1989), in which a man is ill and decides to approach others in hopes of finding a cure. He is given the wrong advice by two people before coming to the realisation that he is not considering the profession of the people that he is asking. He reasons that a doctor would be able to tell/give him the cure. He proceeds to find and obtain the correct information to cure himself from a doctor.

This scenario was recreated with one modification. The player has four possible NPCs to query instead of three. This was due to the source material for the other five scenarios (detailed in Section 9.1) utilising four characters as opposed to three. It was reasoned that this increase in NPCs would both provide more uniformity throughout the scenarios and would not affect the structure and intended educational content of the scenario.

The player is able to query NPCs that take on the roles of a beekeeper, farmer, butcher and doctor. Players can distinguish the role of an NPC by taking note of their attire and reading the sign located next to each NPC which explicitly states his role. The player can only conclude this scenario by talking to the doctor NPC who will correctly give him the cure. The doctor NPC was also placed furthest away from the player's start point with the intent that the player asks the more closely placed NPCs before reaching the doctor. This was done to allow for more in-game conversation explaining how each NPC could either be giving right or wrong information based on their profession.

Figure 6.4 details the sequence of events and the dialog options available to the player for scenario 1. The labels in the figure refer to the label of each dialog option detailed in Table 6.1. Table 6.1 displays all dialog options and the dependencies of each option signifying the order in which dialog options become available. Figure 6.5 is a map showing the area scenario 1 occurs in and placements of all NPCs. Figure 6.6 displays the general layout of the playable area. Figure 6.7 shows the interior of one of the buildings. Figure 6.8 displays an example of the player engaging in dialog with one of the NPCs.

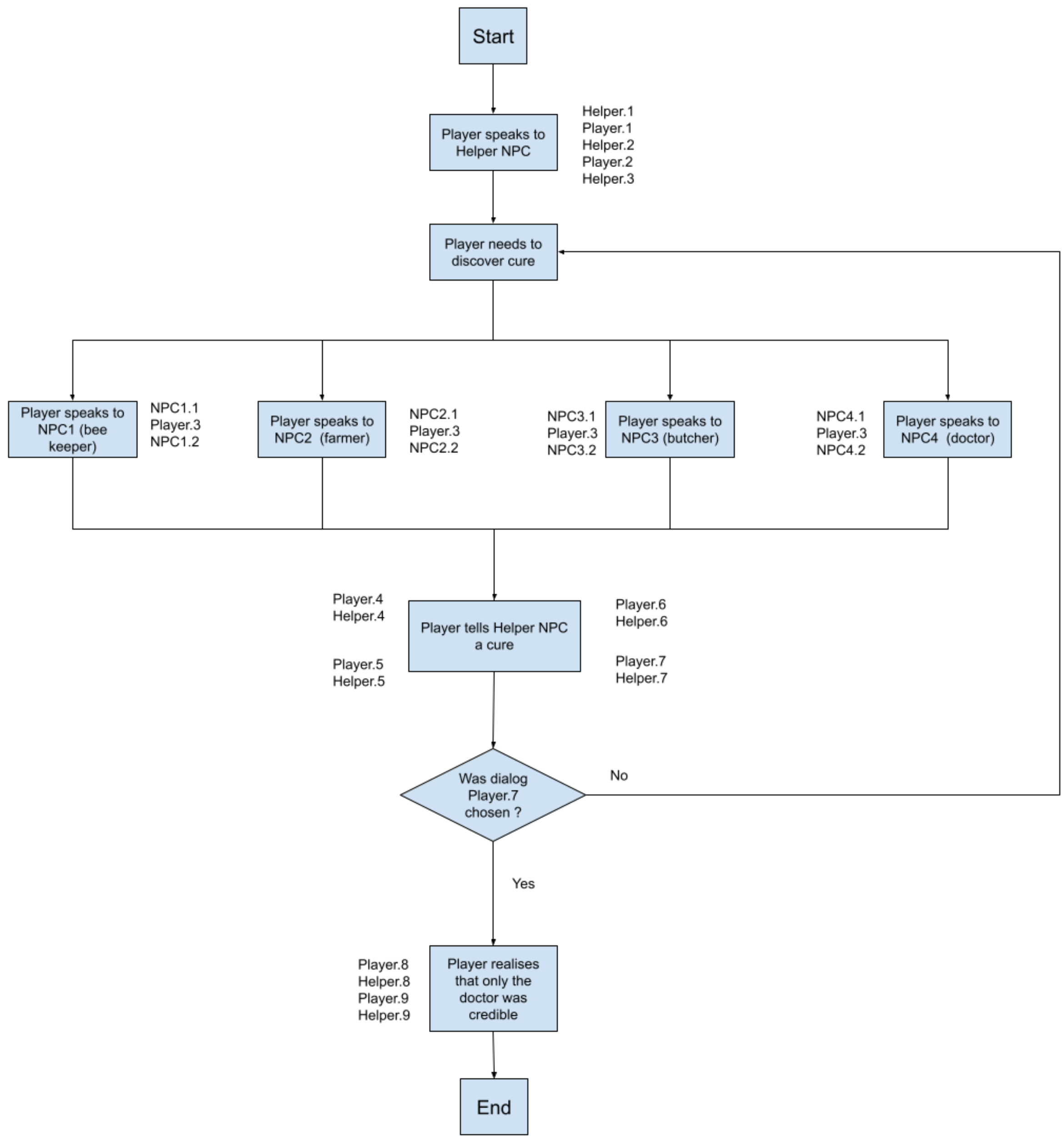

Figure 6.4: A flowchart showing the sequence of events and the dialog options for scenario 1

Table 6.1: The specification of all dialog options and their dependencies for scenario 1.

Label	**Dependencies**	**Full dialog**
Player.1	Helper.1	What do you need?
Player.2	Helper.2	Sure, I'll help you find the cure.
Player.3	Helper.3	I need to know the cure for eating a rotten potato.
Player.4	NPC1.2	Drink some honey, you'll fell better.

Player.5	NPC2.2	I heard eating carrots will help.
Player.6	NPC3.2	Have you tried some rabbit stew?
Player.7	NPC4.2	Eating a dandelion is the cure.
Player.8	Helper.7	It was the doctor.
Player.9	Helper.8	Next time, I'll go straight to a doctor then.
Helper.1		Hi there! I need your help!
Helper.2	Player.1	I've eaten a rotten potato and now I'm sick. Can you help me find the cure?
Helper.3	Player.2	Ask the villagers around here if they know the cure.
Helper.4	Player.4	Hmmm, I'm still sick...
Helper.5	Player.5	That didn't seem to work...
Helper.6	Player.6	I've tried that and it failed.
Helper.7	Player.7	Hey! I suddenly feel a lot better! Who told you about this cure?
Helper.8	Player.8	Ah, since doctors have studied medicine, they would be the best people to ask about sickness and cures.
Helper.9	Player.9	Sounds good. Thanks for all your help, I feel much better now.
NPC1.1	Helper.3	How can I help you?
NPC1.2	Player.3	Drinking honey will cure it.
NPC2.1	Helper.3	What do you need?
NPC2.2	Player.3	Eating carrots always helps with sickness.
NPC3.1	Helper.3	Hello.
NPC3.2	Player.3	Rabbit stew is the cure.
NPC4.1	Helper.3	Do you need help?
NPC4.2	Player.3	Eating a dandelion will cure it.

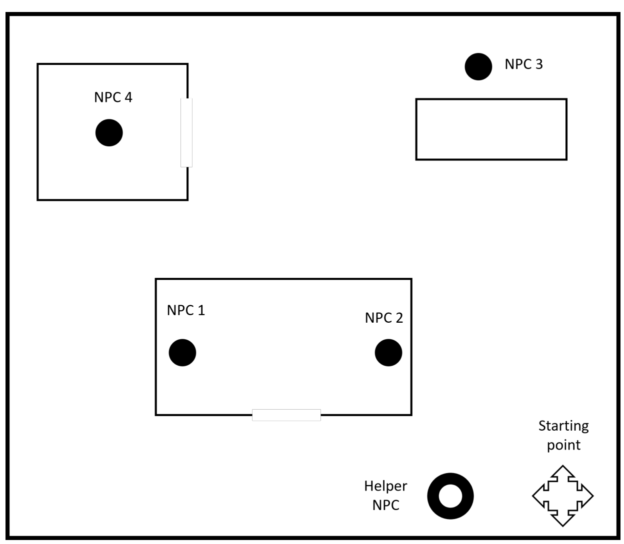

Figure 6.5: A map showing the area and placements of NPCs for scenario 1

Figure 6.6: A screen shot showing the general layout of the playable area.

Figure 6.7: A screen shot of the interior of one of the buildings.

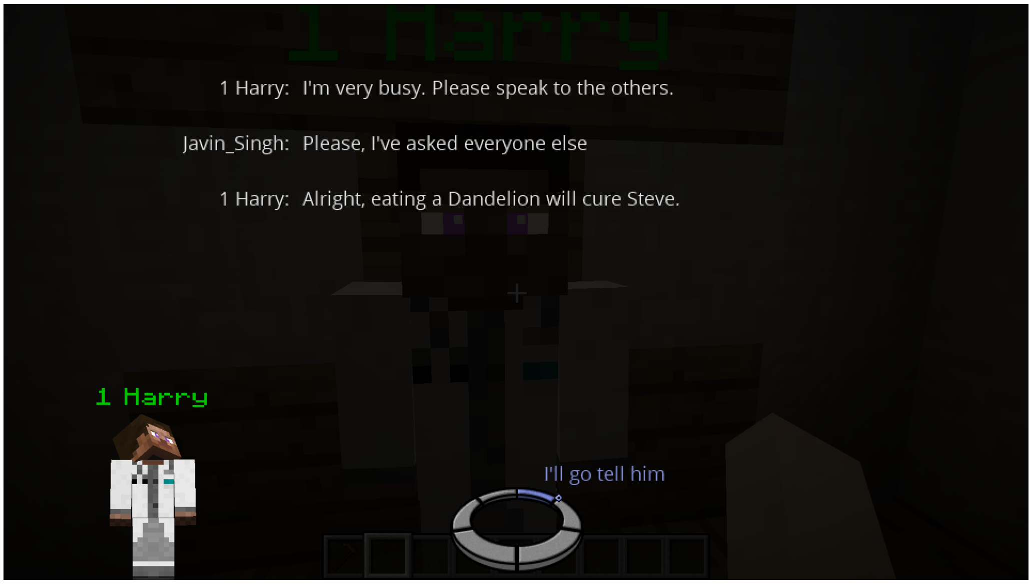

Figure 6.8: A screen shot of the player engaging in dialog with one of the NPCs.

With all software components developed, the following section explores carrying out the user evaluation and the results yielded.

6.4 Results

The evaluation was carried out on 23 participants, all of whom completed all six scenarios in the game. The participants were split into the control group, who played the educational game without the adaptation algorithm, and the experimental group, who played the game with the adaptation algorithm. The control group comprised 11 participants and the experimental group comprised 12 participants. As stated before the aim of this evaluation is to determine the effectiveness of the algorithm at enhancing the learning gains of users when playing an education game. Evaluating this aim will be split into four tests. The first tests for a statistically significant difference in learning gains of all players to establish that the educational game created could produce learning gains irrespective of the activation of the algorithm. The second tests the control group for significant learning gains to prove that the educational game developed has the ability to educate without adaptability. The third tests the experimental group for significant learning gains to prove the educational game's ability to educate with adaptability. The fourth compares both groups for a statistically significant difference between learning gains as a measure of the algorithm's effect on learning gains.

6.4.1 Analysis methodology

As each of the aforementioned tests involve comparing two data sets for significant differences, the T-test was selected as the analysis methodology. Fadem (2008) further explains that the T-test for two data sets can be divided into the paired and unpaired test. The purpose of the paired T-test is to measure a single group at different points in time. Due to this, the paired T-test was used for the first three tests using the equation:

$$t = \frac{m}{s/\sqrt{n}} \tag{6.1}$$

Where: t = t-statistic, m = mean of the group, s = standard deviation of the group, n = sample size

The unpaired T-test is used to compare the means of two different data sets for statically significant differences. Thus, it is more appropriate to be used for the fourth test via the equation:

$$t = \frac{m_A - m_B}{\sqrt{\frac{S_2}{n_A} + \frac{S_2}{n_B}}} \tag{6.2}$$

Where: m_A and m_B are the means of two different samples, n_A and n_B are the sample sizes, S_2 is an estimator of the common variance of two samples using the equation:

$$S_2 = \frac{\sum(x - m_A)^2 + \sum(x - m_B)^2}{n_A + n_B - 2} \tag{6.3}$$

As the aim of this study is to determine if the algorithm's inclusion makes the effectiveness of educational games strictly higher, only one tailed T-tests were used. Furthermore the standard value of p = 0.01 was selected for all the four tests to ensure the lowest possibility of the results being achieved by pure chance.

6.4.2 Overall learning gains

The purpose of the first test is to establish the developed educational game's ability to educate players regardless of the presence of the adaptation algorithm. This is done by comparing the post and pre-game test scores for all players. Hence it tests the following hypothesis:

H_{0A} = There is no statistically significant difference between the post and pre-game test scores for all participants.

H_{1A} = There is a statistically significant difference between the post and pre-game test scores for all participants.

Table 6.2 shows the pre-game test scores, post-game test scores and the difference between the two for all 23 participants recorded.

Table 6.2: Results for all participants.

Participant	**Pre-test score**	**Post-test score**	**Delta**	**Algorithm Activated**
1	0	2	2	False
2	0	12	12	True
3	8	8	0	False
4	-2	8	10	False
5	6	14	8	True
6	-6	0	6	False
7	6	10	4	False
8	-6	4	10	True
9	7	12	5	False

10	-4	8	12	True
11	0	10	10	True
12	5	5	0	False
13	2	8	6	True
14	-6	6	12	True
15	8	12	4	False
16	6	8	2	True
17	6	14	8	True
18	6	11	5	False
19	0	6	6	True
20	6	12	6	False
21	-5	8	13	True
22	-2	4	6	False
23	4	10	6	True

Using the formulas for running the paired T-test described above, a p-value of 1.30E-08 was calculated. As the p-value is smaller than the alpha p-value of 0.01, H_{0A} can be rejected. Therefore the educational game developed is capable of producing significant learning gains.

6.4.3 Control group learning gains

The purpose of the second test is to establish that the developed educational game has ability to educate players without the adaptation algorithm. This is done by comparing the post and pre-game test scores for the control group players. Hence it tests the following hypothesis:

H_{0B} = There is no statistically significant difference between the post and pre-game test scores for the control group participants.

H_{1B} = There is a statistically significant difference between the post and pre-game test scores for the control group participants.

Table 6.3 shows the pre-game test scores, post-game test scores and the difference between the two for all 11 participants in the control group.

Table 6.3: Results for control group.

Participant	**Pre-test score**	**Post-test score**	**Delta**
1	0	2	2
3	8	8	0
4	-2	8	10
6	-6	0	6
7	6	10	4
9	7	12	5
12	5	5	0

15	8	12	4
18	6	11	5
20	6	12	6
22	-2	4	6

Using the formulas for running the paired T-test described above, a p-value of 2.78E-04 was calculated. As the p-value is smaller than the alpha p-value of 0.01, H_{0B} can be rejected. Therefore the educational game developed is capable of producing significant learning gains without the aid of the adaptation algorithm.

6.4.4 Experimental group learning gains

The third test is meant to establish that the developed educational game has ability to educate players with the adaptation algorithm activated. This is done by comparing the post and pre-game test scores for the experimental group players. Hence it tests the following hypothesis:

H_{0C} = There is no statistically significant difference between the post and pre-game test scores for the experimental group participants.

H_{1C} = There is a statistically significant difference between the post and pre-game test scores for the experimental group participants.

Table 6.4 shows the pre-game test scores, post-game test scores and the difference between the two for all 12 participants in the experimental group.

Table 6.4: Results for experimental group.

Participant	**Pre-test score**	**Post-test score**	**Delta**
2	0	12	12
5	6	14	8
8	-6	4	10
10	-4	8	12
11	0	10	10
13	2	8	6
14	-6	6	12
16	6	8	2
17	6	14	8
19	0	6	6
21	-5	8	13
23	4	10	6

Using the formulas for running the paired T-test described above, a p-value of 9.48E-07 was calculated. As the p-value is smaller than the alpha p-value of 0.01, H_{0C} can be rejected. Therefore the combination of the educational game developed and the adaptation algorithm is capable of producing significant learning gains.

6.4.5 Control group against experimental group learning gains

With the developed educational game's ability to educate players regardless of aid from the adaptation element confirmed, the fourth and final test attempts to determine if the differences in learning gains from having the adaptation algorithm activated are statistically significant. This is done by comparing the differences between the post and pre-game test scores of the control group and the experimental group. Hence, the following hypothesis is tested:

H_{0D} = There is no statistically significant difference in learning gains between the the control group and the experimental group.

H_{1D} = There is a statistically significant difference in learning gains between the the control group and the experimental group.

Table 6.5 shows the difference in scores between the post and pre-game tests for all the 11 participants in the control group and the 12 participants in the experimental group.

Table 6.5: Delta in scores for the control and the experimental group.

Control group delta scores	**Experimental group delta scores**
2	12
0	8
10	10
6	12
4	10
5	6
0	12
4	2
5	8
6	6
6	13
	6

Using the formulas for running the unpaired T-test described above, a p-value of 1.45E-03 was calculated. As the p-value is smaller than the alpha p-value of 0.01, H_{0D} can be rejected. Thus, the inclusion of the adaption algorithm produces statistically significant higher learning gains in players when compared to the same educational game without the adaptation algorithm.

6.4.6 Discussion of the results

Although positive, the results concerning learning gains in players yielded by the user evaluation proved to be unusually more effective when compared to similar instructional approaches (Behar-Horenstein and Lian, 2011) and programmatic approaches (Scott, Markert, and Dunn, 1998). These unusual results were reasoned to be due noise that may have been caused by the following three reasons.

Firstly, it is possible that players who are more adept at learning from a behaviorist, cognitive and Socratic pedagogical approach (which are the pedagogical models more heavily utilised by educational scenarios) were randomly sorted into the experimental group and thus were able to achieve a higher degree of learning gains. It is reasoned that for future, more thorough evaluations, this issue could be avoided by sorting participants based on the pedagogical approach(s) they are most adept at. Researchers such as Ortega (2020) and Schmidt and Watanabe (2001) explain that an individual's adeptness towards different pedagogical approaches can be ascertained by surveying and/or interviewing him/her. The individual's teacher(s) can also be surveyed and/or interviewed where appropriate. By doing this, the level of adeptness at different pedagogical approaches could be made more balanced throughout both groups.

Secondly, it is possible that the players whose learning could benefit from adaptivity the most were randomly sorted into the experimental group and therefore inflate the measured learning gains. Unfortunately, no method to determine which players could benefit from adaptivity the most prior to taking the evaluation could be identified. However, the studies explained in Section 6.2.2 made use of participant sample sizes of 30 or more, and had significantly less noise in their results. Hence, it is similarly theorised that aiming for a larger sample size in future experiments might be able to reduce the overall noise and produce more accurate results.

Thirdly, as the pre and post game tests exclusively comprised multiple choice answer questions, there exists a possibility of the correct answer being guessed without the comprehension and/or application of the educational content. Building from this it is possible that players in the experimental group guessed the correct answers during their post game tests thereby artificially increasing their measured learning gains. In a similar vein, it is also possible that players in the control group wrongly guessed answers in their post game tests thereby artificially decreasing their measure learning gains. It reasoned that this issue could be remedied by modifying both tests to require participants to correctly explain/justify their answers before being awarded a mark. The participants' explanations/justifications could be compared to the test's own model explanations to determine correctness. This could serve as a way to eliminate participants rightly and wrongly guessing answers through sheer chance.

6.5 Conclusions

Although all four tests performed yielded statistically significant positive results, H_0 could not be rejected due to the following four reasons. Firstly, the evaluation conducted was a preliminary one due to time and resource constraints. The unusually positive results from this preliminary evaluation suggest a high degree of noise and/or artefacts. Hence, a more thorough evaluation would have to be conducted to shed further light on these preliminary results.

Secondly, the evaluation did not include adaptive content for the entire RAES model. Hence, any conclusions that could be drawn from this evaluation, positive or negative, would be in reference to a partial model. Content for the full model would need to be developed and evaluated before any concrete conclusions regarding the effectiveness of the algorithm can be drawn.

Thirdly, as explained in previous sections, an instructional approach was use to teach the educational content as opposed to a programmatic approach. While not as accurate as a programmatic approach, an instructional approach is still described as worth doing when resource and time constraints do not allow for a programmatic approach to be used. Hence more accurate results regarding the effectiveness of the adaptation algorithm could be brought forward by utilising a programmatic approach to teaching in future user evaluations.

Fourthly, the studies involving instructional approaches to teaching made used of participant samples sizes of 30 or more. Thus, aiming for a similar or higher number of participants in future evaluations may lead to more accurate results concerning the algorithm's effectiveness.

Despite these issues and limitations however, three positive findings were drawn from the evaluation. Firstly, even though adaptive content for four out of the six elements were developed, the effect of those four elements presented the potential to be of benefit to players' learning. It was reasoned that this is because an equal measure of preference and performance elements, and active and passive adaptability was incorporated into the educational game. Active adaptability pertaining to the player's ability at the educational content combined with the passive adaptability for support and self-reflection for players experiencing difficulty is believed to have created a flow experience. Achieving the flow experience, explained in Chapter 2, allowed players to learn at a pace he/she was comfortable with while simultaneously being entertained. Due to this, it is believed that creating adaptive content for the other two elements may further increased the positive effects on learning gains of the algorithm due to higher degrees of active adaptability.

Secondly, the utilisation of Minecraft proved to be effective in handling the entertainment value of an educational game due to its popularity and refined game mechanics. This created a smooth game-play experience where players were more motivated to complete the game and learn the educational content while doing so. Thus, using Minecraft allowed for more focus to be directed towards educational content, the adaptation algorithm and the adaptive content. While developing a custom educational game from the ground up is possible, it would require significant time, effort and resources to ensure effective entertainment value and is therefore not recommended.

Thirdly, it is believed that the use of educational scenarios could have had a positive impact on the educational effectiveness of the developed game. Using educational scenarios allowed for the educational content to be masked behind a compelling problem that the player was presented with. This allowed players to be entertained while the learning of the content took place in the background. Therefore, utilising educational scenarios to teach content in educational games appears to contribute towards the creation of an effective educational game.

With the user evaluation complete, the next chapter focuses on devising a methodology to utilise the adaption algorithm with other educational games along with an evaluation of said methodology.

Chapter 7

Methodology for Utilisation

To accomplish the goal of making educational games adaptive to improve the learning of their players, both a technical solution and a methodology to apply said solution would be needed. As the adaptation algorithm itself is the technical solution, this chapter focuses on the methodology for utilising the adaptation algorithm with educational games and the evaluation of said methodology. The utilisation methodology is intended to instruct game designers, who want to make educational games adaptive in order to improve learning in players, to incorporate the adaptation algorithm with said educational games. Furthermore, the effectiveness of the utilisation methodology's instructions to incorporate the adaptation algorithm with educational games was also evaluated. The techniques and results of said evaluation are presented as well.

7.1 Utilisation methodology

The purpose of this utilisation methodology is to provide game designers with instructions to incorporate the adaption algorithm with educational games to make them adaptive. When examining educational game design methodologies from literature such as the ones put forth by DeLope, Arcos, et al. (2017) and DeLope, Medina-Medina, et al. (2017), educational game design is split into the five chronological stages of startup, design, production, testing and post production. However, the utilisation methodology is meant to be used to incorporate the algorithm with existing educational games, as opposed to having to create educational games from scratch. Hence, the utilisation methodology would differ in structure from the two game design methodologies mentioned above. Despite the difference in starting points, it was reasoned that adopting the practice of splitting the utilisation methodology into chronological stages, similar to that of the game design methodologies, would provide greater clarity.

Utilising the algorithm is split into five steps. First, is to ensure that the educational game chosen to be paired with the algorithm possesses the necessary functionality to be compatible. Second, is the design of educational scenarios. Third, is to make sure that the algorithm is triggered at the correct instances. Fourth, is coding the algorithm using the pseudo-code in Section 4.4. Fifth, is to develop element specific content for each of the elements chosen to be made adaptive using the specified methodologies. The methodologies were applied to the RAES to provide an illustration of the result. Ensuring that each of the five stages of the utilisation methodology are fulfilled will allow for the algorithm to function with any educational game. Figure 7.1 shows a visual representation of the five steps of the utilisation methodology and serves as a tool for developers to document their progress and decisions.

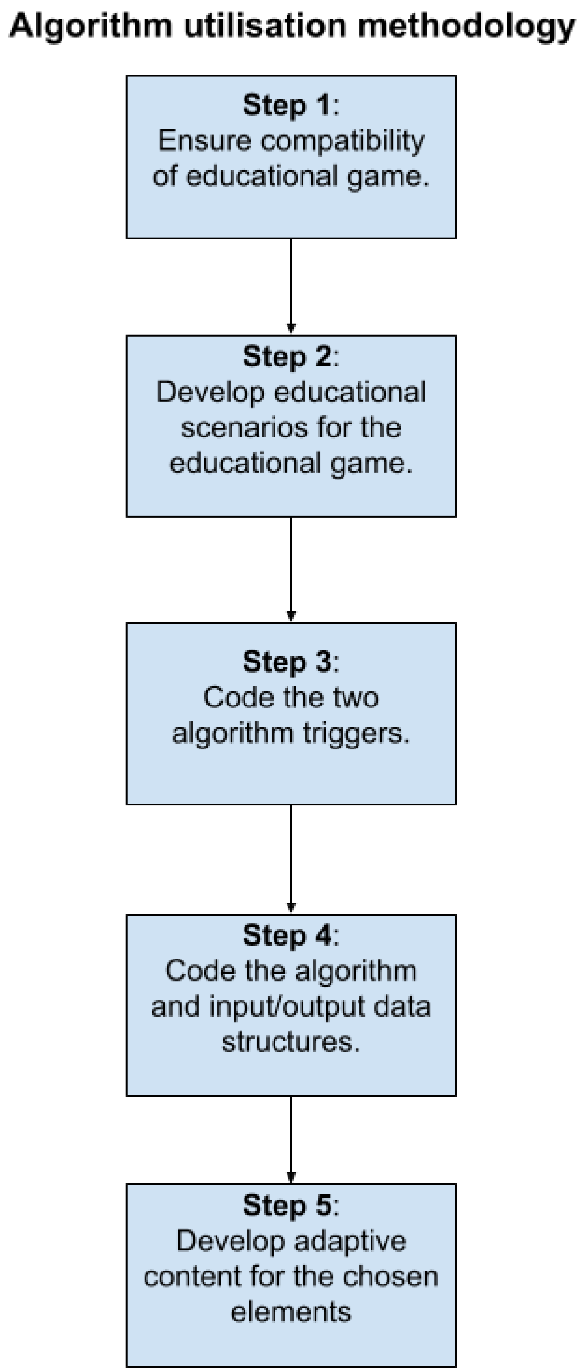

Figure 7.1: A diagram showing the five steps of the utilisation methodology.

7.1.1 Educational game compatibility

Concerning step one, the utilisation of the algorithm is dependent on being paired with a compatible educational game. A compatible educational game is one that fulfills the three Compatibility Requirements (introduced in Section 6.3.1) of player modelling, educational scenarios and a system clock. It should be noted that it is possible for non compatible educational games to be made compatible by manually implementing these three functionalities, such as what was done with Minecraft for the user evaluation (Chapter 6). The following presents the functionalities required to fulfill the three Compatibility Requirements that can either be check for, or implemented in potential educational games:

Player modelling (Compatibility Requirement 1). The educational game must possess the ability to run custom/modified code along with its original processes. As long as a game possesses this functionality, the algorithm detailed in Section 4.4 should be able to be recreated in any programming language and incorporated with said game. However, it is possible for an educational game to not function if it is detected that its source code has been altered and/or if new/custom code is detected. Selecting an educational game that does not posses this functionality would render any player modelling code and the game itself unusable.

Educational scenarios (Compatibility Requirement 2). The educational game must possess the ability to execute educational scenarios to teach the educational content. Functionalities needed to accomplish this include:

1. The tasks element from the GES. Educational scenarios by design challenge players to accomplish objective(s) by learning and applying educational content. Therefore, the game must be able to track players' progression and identify when objective(s) have been reached. Depending on the game of choice, it will be up to game designer(s) to determine suitable method(s) to implement this functionality due to the differing source codes and/or engines.
2. Method(s) of presenting educational content. The game must posses at least one mechanism to present the intended educational content to its players. For example, the version of Minecraft used this study made use of a custom made text based dialog system with NPCs to convey the educational content. Other examples include a fully voiced dialog system, cut-scenes and game objects that contain the content in text form that players can refer to. It is at game designers' discretion to pick and choose which mechanism(s) are most suitable for their own game.
3. Method(s) to make players apply educational content. This is directly tied to the level styles element and the five options identified in Section 3.1.4. The game must posses the ability to carry out at least one option to challenge players to apply the intended educational content. For example, the version of Minecraft used this study consisted of contextual problems presented by NPCs where players would have to speak to other NPCs to gather information and apply the educational content to deduce the solutions, all done within the custom made text based dialog system. Please see Section 7.1.4.1 for general descriptions of how each level style option may challenge players to apply educational content. It is at game designers' discretion to pick and choose which level style option(s) are most suitable for their own game as each option would require different functionalities, assets and possibly different method of implementation depending on the game of choice.

System clock (Compatibility Requirement 3). Although the majority of both educational and entertainment games have possessed this functionality for decades, it is possible that some games might not. However, the operating system of the platform that games run on would have a system clock. Thus, a game that does not have its own system clock could call on the operating system's clock to fulfill this functionality.

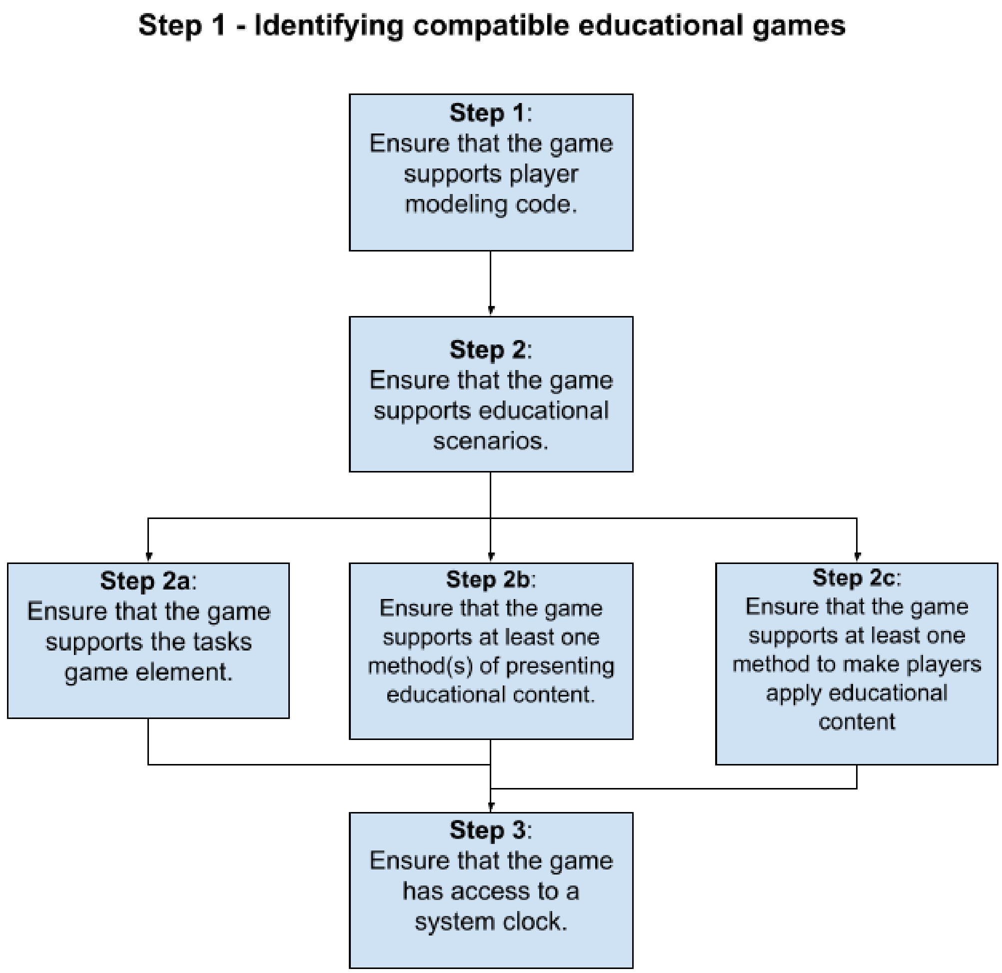

Figure 7.2: A diagram showing the steps to ensure that an educational game is compatible.

Figure 7.2 acts a visual representation of the steps and a checklist for documenting progress and decisions made when ensuring the compatibility of an educational game. Thus developers must ensure that his/her chosen educational game possesses these three functionalities to be compatible with the algorithm.

7.1.2 Designing educational scenarios

Regarding step two, it was previously explained that the inclusion and execution of educational scenarios is essential to the function of the algorithm. Although the concept and design of educational scenarios is open ended due to being based on the creativity and choices of game designers, guide points towards designing scenarios in the vein as the ones made for the user evaluation were established. This was split into six steps of considerations regarding the intended educational content before designing scenarios, coming up with task(s) for scenarios, and considerations regarding each of the four stages of educational scenarios. It is recommend that stage 4 (Content application) be considered directly after deciding on the intended educational content. This is due to stage 4 following directly from the educational content. Therefore, design choices made for stage 4 can directly affect design choices made for the first three stages. As such, the following six step **educational scenario creation methodology** was devised:

1. **Educational content**. By their nature, educational scenarios must contain educational content that they are meant to teach. Although the subject chosen is at the complete discretion of game designers, it is suggested that each educational scenario aim to teach one topic or subtopic of the chosen subject. This is to potentially avoid over loading players with information. For example, the subject chosen for the user evaluation of this study was analytical and critical thinking. Multiple sources of information for this subject comprised multiple topics and sub topics. The topic chosen was the accuracy and credibility of evidence, which consisted of multiple sub topics (see Section 6.2.1 for full details). Each of the six educational scenarios developed aimed to teach one sub topic.

2. **Content application (Educational Scenario Stage 4)**. Consider the subject and topic of the educational content chosen and think of the level style(s) that best suits said content. Taking the user evaluation as an example, when attempting to teach ascertaining the accuracy and credibility of information, the most appropriate level style was identified as the problem solving option. This is because using techniques involving quick thinking, memorisation, navigation and planning are not credible ways to teach the chosen content based on the source material. Therefore, the premise chosen was an NPC posing a problem with the other NPCs presenting differing information that could lead to solutions. Only one NPC would provide the accurate and/or credible information. Players would be required to acquire all the possible information and work out which was accurate and/or credible, thereby finding the solution.

3. **Task(s)**. As Section 2.2.1.2 pointed out, educational scenarios must have some manner of task(s) for the player to accomplish. Thus, consider at least one task per scenario to provide the player with a goal to work towards and apply the educational content to accomplish. The nature of each task will vary based on the level style option(s) chosen, which in turn is based on the educational content chosen. As stated above the educational scenarios from the user evaluation utilised the tasks of an NPC providing a problem and/or question for the player to solve. Furthermore, each task would need at least one predefined point of progress to be specified by developer(s). This is so that the performance elements that have been selected can be adapted to players. The most straight forward approach is to have a task's predefined point of progress at the very end when the task has been accomplished. Developers may set intermediary points of progress within one task if they so choose.

4. **Task introduction (Educational Scenario Stage 1)**. Consider the level style option(s) chosen for the scenario and think with regards to the elements in the GES that could stoke interest to complete and/or make the scenario's task compelling. Examples include an interesting narrative, compelling NPCs that players would want to help and a mysterious environment to explore.

5. **Content acquisition (Educational Scenario Stage 2)**. Consider the subject and topic of the educational content chosen and think of methods of presentation that could best suit said content. For example, this study utilised a text based dialog system with NPCs to convey the educational content. However, it could be argued that for subjects whose content requires a visual component, such as the drawing of graphs in Mathematics, some form of visual presentation like a cut-scene or interactive diagram could be more appropriate. It will be up to the creativity and discretion of game designers to decide the best manner at which to present the educational content for players to acquire.

6. **Content comprehension (Educational Scenario Stage 3)**. Consider the subject, topic and chosen level style options of the scenario and identify ways that the player could be helped and/or guided when experiencing difficulties with the content. This is linked to the motivators element. It will be up to game designers to decide on the motivator options to use and how best to implement them. For instance, the version of Minecraft used for the user evaluation used text based messages and visual cues to aid the player in understanding why some of the proposed solution are wrong.

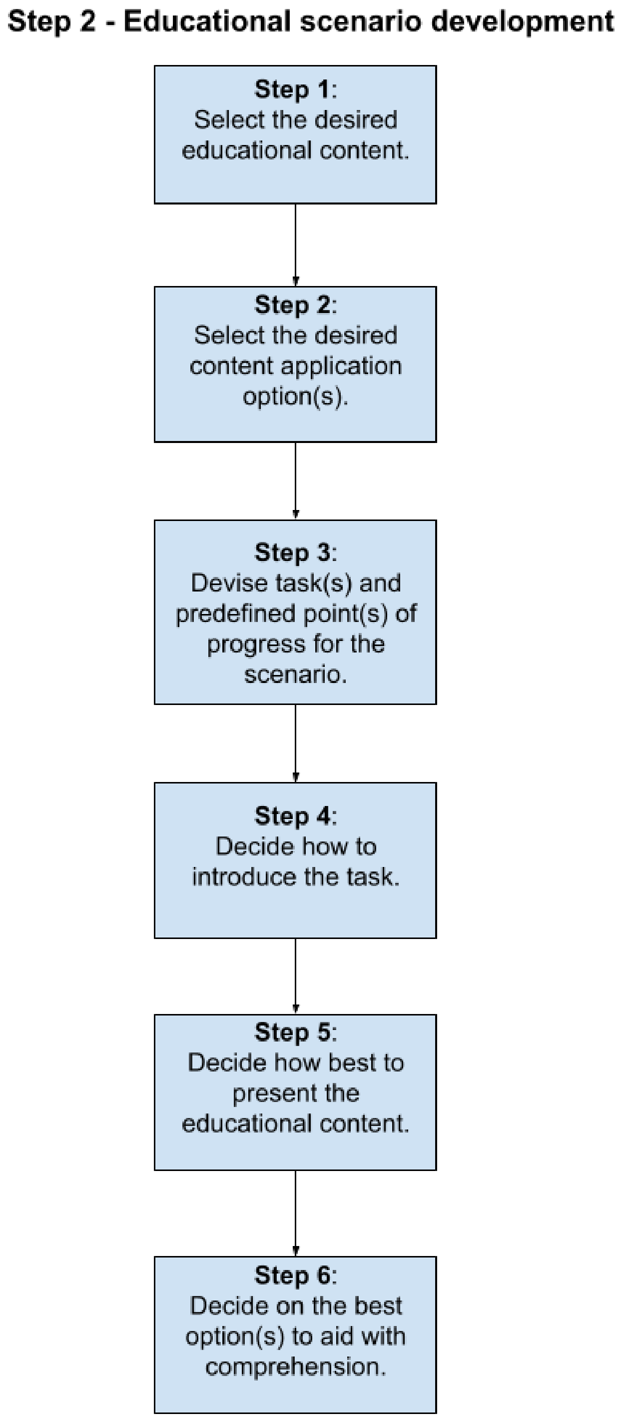

Figure 7.3: A diagram showing the six steps to develop an educational scenario.

Figure 7.3 acts a visual representation and a checklist for documenting considerations, progress and decisions made when developing an education scenario. Following the steps in this methodology should allow game designers to create educational scenarios for any educational content.

7.1.3 Using the triggers

For step three, as explained in Chapter 3, the algorithm is triggered by either a player interaction or a non player interaction. Each trigger is linked to and causes different adaptation conditions to be checked. As different educational games employ different game engines, the implementation of the triggers might differ from game to game. However, the steps to utilising each trigger remains constant. The methodologies explaining the steps for the utilisation of each trigger are presented below.

7.1.3.1 Player interaction triggers

The **player interaction trigger methodology** is used to implement the player interaction trigger and involves two steps:

1. **Player interactions**. Triggering the algorithm every time an explicit player interaction is detected. An explicit player interaction is define as in game action a player could execute by physically performing an action with a peripheral. Examples include moving a mouse to look around, swiping a touch pad to pick an item up and pressing a button to jump. Every educational and entertainment game have a finite number of in game actions that are mapped to physical peripherals managed by their game engine. To implement step one:
    (a) Identify every type of in game action possible by physically interacting with peripherals in the game of choice.
    (b) Set the algorithm to be triggered upon the execution of all in game actions from step 1A.
2. **Relevant interactions**. Identifying and setting the relevant interactions for each scenario. Relevant player interactions are defined as ones that progress an educational scenario. Relevant interactions are required to cause the checking of Adaptation Conditions 1, 3 and 4 by the algorithm. To implement step two:
    (a) Identify every type of player interaction possible via the game's finite number of in game actions.
    (b) Examine each educational scenario and determine which type of player interaction(s) from step 2A cause it to progress.
    (c) Add the types of player interactions from step 2B to the list of relevant actions for that scenario (SBD.1 variable in the scenario benchmark data set).

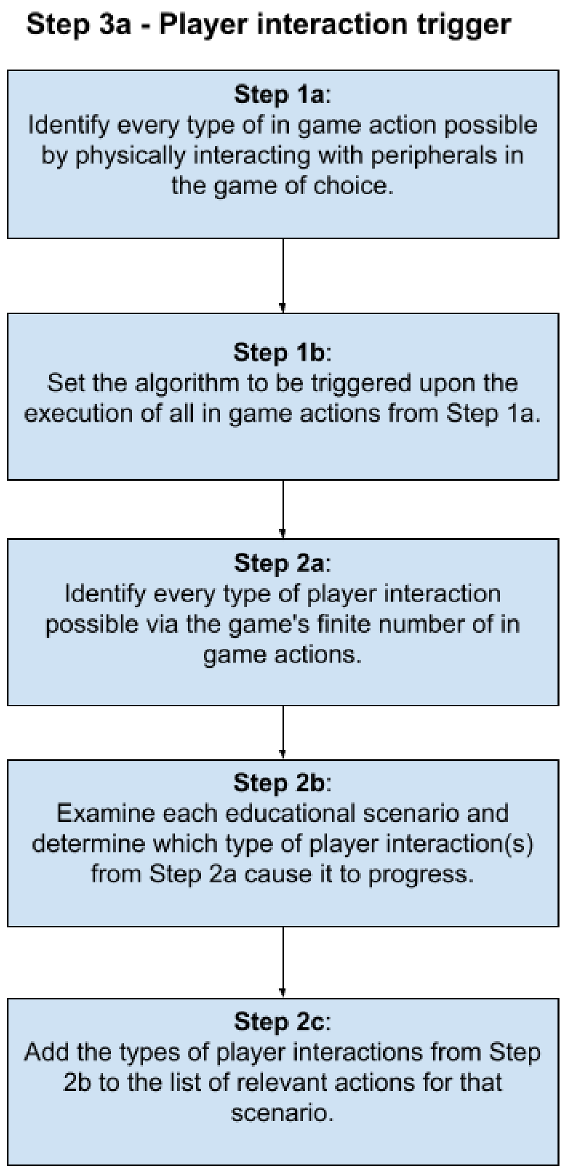

Figure 7.4: A diagram showing the steps to use the player interaction trigger.

Figure 7.4 acts a visual representation and a checklist for documenting progress and decisions made when using the player interaction trigger. An example of using the player interaction trigger methodology on the implementation of the algorithm in the custom build of Minecraft explained in Chapter 6 is as follows :

1. (a) Move forward, move backward, move to the left, move to the right, jump, look around, initiate conversation with NPCs and select dialogues options.

   (b) The algorithm was set to trigger upon the game engine detecting any of the in game actions in step 1A.

2. (a) Move forward, move backward, move to the left, move to the right, jump, look around, initiate conversation with NPCs and select dialogues options.

(b) All the six educational scenarios are progressed by starting conversations with NPCs and selecting the appropriate dialog options to uncover the educational content. Thus, the relevant interactions are:

  i. Initiate conversation with NPCs.

  ii. Select dialogues options.

(c) Add initiate conversation with NPCs and select dialog options to SBD.1.

7.1.3.2 Non player interaction triggers

As explained in Chapter 3, two situations where adaptation could benefit a player's learning that do not involve a player's interactions were identified. These include timing players progress and tracking if the player has reached a predefined point of progress in educational scenarios. The **non player interaction trigger methodology** is used to implement the non player interaction triggers:

1. **Timing the player**. Timing the player is linked to Adaptation Conditions 2 and 3. This trigger is implemented by:

   (a) Instruct the game engine to trigger the algorithm once per second. This does not include time spent paused.

2. **Point of progress**. When a player reaches predefined points of progress. This trigger is linked to Adaptation Condition 5. This trigger is implemented using one of two ways:

   (a) Instruct the game engine to automatically trigger the algorithm upon detecting that any predefined point of progress has been reached.

   (b) Hard code the algorithm's trigger at the end of each point of progress's script and/or sequence of events.

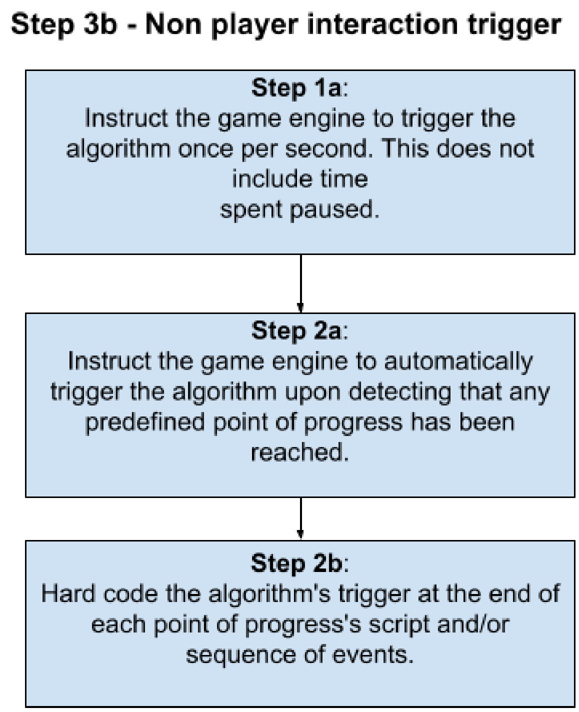

Figure 7.5: A diagram showing the steps to use the non player interaction trigger.

Figure 7.5 acts a visual representation and a checklist for documenting progress and decisions made when using the non player interaction trigger.

7.1.4 Coding the algorithm

Step four involves coding the algorithm itself. To do this, game designers would use the psuedo-code for the algorithm (Section 4.4) and the inputs and outputs structures (Section 4.2) as blue prints for coding the algorithm and declaring the inputs and outputs in any programming language. As Section 7.1.1 explained, as long as a game possesses the functionality to run custom/modified code along with its original processes, the recreation of the algorithm as well as the programming language to use will be dependant on the chosen game and game designers' discretion.

7.1.5 Element specific content

To carry out step five, each element chosen to be adaptive requires specific content for the algorithm to switch to and from. Below are the methodologies to create element specific content for preference and performance elements along with the application of the methodologies to the RAES.

7.1.5.1 Preference elements

The following is the **preference element methodology** to be used to create element specific content for any preference element:

1. Identify the purpose the element serves to educational games.
2. Select the options of the element to switch to and from.
3. For each option, create appropriate assets, scripts, instructions and/or sequence of events.
4. Tag the things created in step 3 with an identifier linking them to the option they belong to.

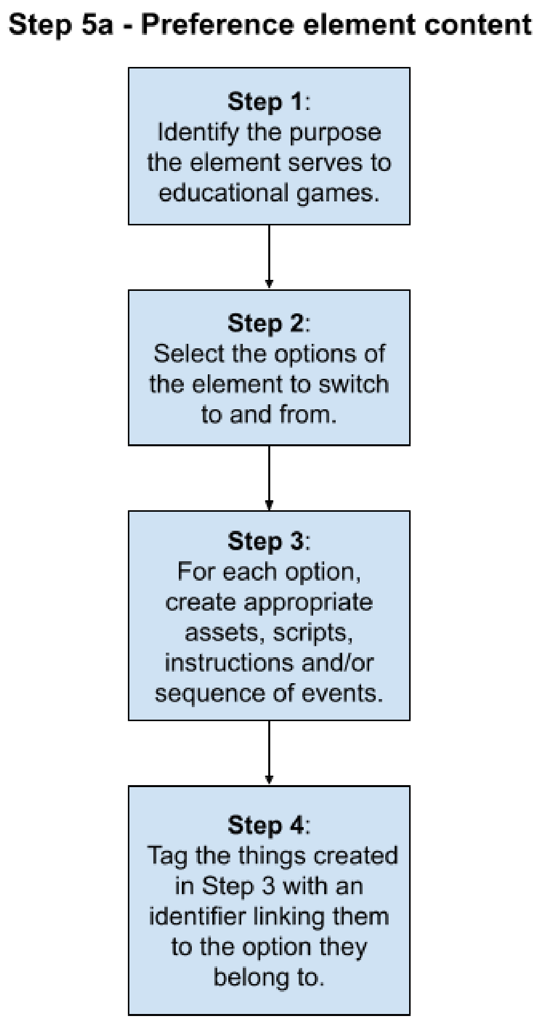

Figure 7.6: A diagram showing the steps to developing adaptive content for preference elements.

Figure 7.6 acts a visual representation and a checklist for documenting progress and decisions to make when developing adaptive content for preference elements. Using this methodology for the four preference elements in the RAES yielded the following results.

Level Style Via the preference element methodology:

1. To alter the way in which the player is challenged in an educational scenario.

2. Memory, navigation, quick thinking, problem solving, strategic thinking.

3. Create for each of the options:

   (a) Memory. A series of questions and/or puzzles that require the player to recall the information they were presented.

   (b) Navigation. A task where the player needs to traverse to a different area in the game world with obstacles that can only be overcome by application of the educational content blocking his./her path

   (c) Quick thinking. A puzzle/problem that the player must solve. The educational content must be available at the player's discretion as the challenge is not aimed at their memory.

   (d) Problem solving. A contextual situation posing a problem/question that the player must find an answer for by reasoning using the context and the educational content.

   (e) Strategic thinking. A contextual situation posing a problem/question where the solution was not explicitly stated or eluded to by the educational content. The challenge stems from the player examining the educational content for a pattern(s) and inferring the next logical step in said pattern as the solution.

4. Tag each asset, script and instruction to link it to the level style option it belongs to.

When developing level style specific content, there exist two constraints. Firstly, the nature of the educational content chosen might render some of the level styles unusable. For example, the educational content chosen for the evaluation of this research in Chapter 6, was a part of a critical thinking education syllabus pertaining to the examination of evidence for credibility. Thus, the quick thinking, navigation and memory level styles proved to be inappropriate due to the style of challenge conflicting with the skepticism, inquisitional, and reasoning as opposed to memorisation skills being taught in the scenarios. Depending the educational content chosen, it will be up to the developer to decide which style(s) are appropriate and develop content for the one(s) selected.

Secondly, as explained in Chapter 3, ascertaining a player's preference in level style involves ensuring that the player has understood the educational content and/or the challenge he/she has been presented with. This is done to ensure that a player's lack of progress is due to a lack of preference towards a level style. Therefore, the communication element must have element specific content developed before content for the level style element can be developed.

Communication Via the preference element methodology:

1. To change the manner at which instructions and educational content is presented to the player.

2. Formal and informal.

3. (a) Selecting an initial communication option.

   (b) Write/record all text and/or recordings using the initial option.

(c) Convert the text and/or recordings to other styles using linguistic techniques such as the ones explained by Cho et al. (2007) and Cortazzi and Jin (1997).

4. Tag each text and/or recording to link it to the communication option it belongs to.

There is a constraint when it comes to developing content for the communication element. Chapter 3 discussed the need for the player to have received assistance using his/her preferred option of motivator before the algorithm adapts the communication element. This is so that a player's lack of progress can be attributed to not comprehending the way in which information is presented to him/her as opposed to failing to understand the information itself. Therefore, content for the motivators element must be developed with at least one of it's options active before content for the communication element can be developed.

Motivators Via the preference element methodology:

1. To aid the player in completing the educational scenario and understanding the educational content.

2. Verbal hint, visual hint, direct hint, encouragement.

3. Create for each of the options:

   (a) Verbal hint. A text message or audio recording suggesting for the player to reconsider changing his/her approach and interactions to the ones needed to progress. Examples include text messages or an audio recordings repeating the relevant educational content or suggesting that the player is in the wrong game area to find what he/she is looking for.

   (b) Visual hint. A form of visual cue indicating the course of action to progress to the player. Examples include a glowing line appearing temporarily to lead the player to the relevant information, and an incorrect dialogue option briefly flashing red.

   (c) Direct hint. A text message or audio recording more explicit stating of the correct approach to take and/or the specific educational content needed to progress. An example is telling the player which NPC or area he/she would need to interact with or be in to get the information he/she needs.

   (d) Encouragement. A text message or audio recording meant to motivate a player to continue to attempt learning and playing. Examples of this include quotes such as "Don't worry you can do this" and "Keep trying, you're almost there". This can also be done by a text message or an audio recording directed at the player.

4. Tag each motivator to link it to the motivator option it belongs to.

Feedback Via the preference element methodology:

1. To convey the effect(s) of the player's interactions to himself/herself.

2. Audio, visual, world.

3. Create for each option:

   (a) Audio feedback. An audio cue when the player performs an action in the game. Examples of this include, footstep sounds when the player character walks and a horn sound effect when the player tries to pick up the wrong object.

(b) Visual feedback. A visual cue when the player performs an action. Examples of this implementation are an object physically moving towards the player character to symbolise the character picking it up, and NPCs changing their facial expressions depending on the right and/or wrong dialogue options that the player chose.

(c) World feedback. A form of change to the game world when the player performs an action. For example, the screen could shake when the player attempts to pick up and object, or for it to start raining in the game world if the player traverses to the wrong area.

4. Tag each asset, script and instruction to link it to the feedback option it belongs to.

7.1.5.2 Performance elements

The following is the **performance element methodology** to create element specific content for any performance element:

1. Choose player metrics to be used by the algorithm.
2. Identify the purpose the element serves to educational games.
3. Link the element to the appropriate player metric(s).
4. Use the formulae from Section 3.4.1.2 to create completion multipliers (CM) and completion thresholds (CT) for each linked player metric.

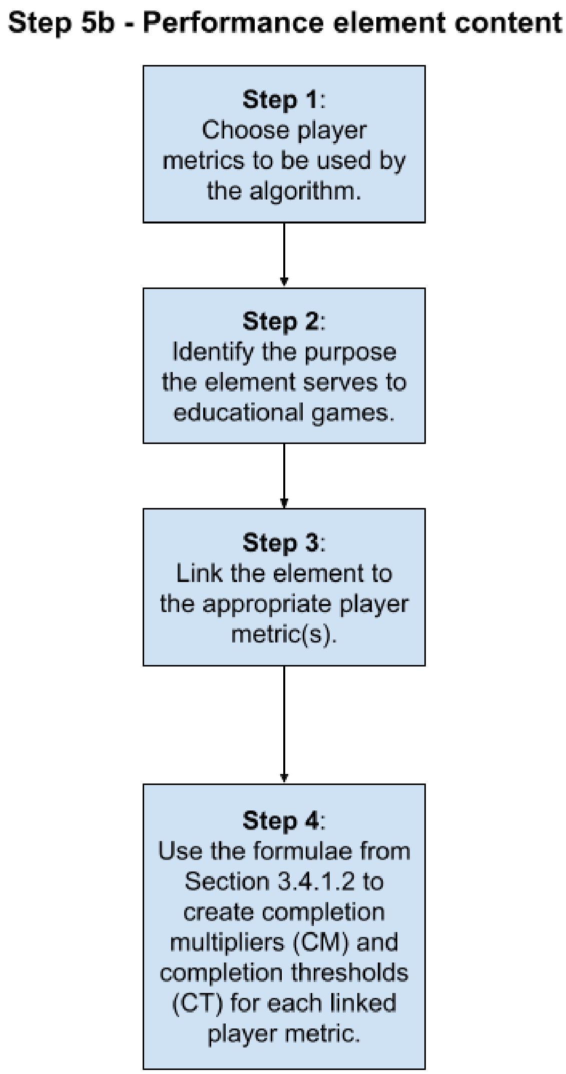

Figure 7.7: A diagram showing the steps to developing adaptive content for performance elements.

Figure 7.7 acts a visual representation and a checklist for documenting progress and decisions to make when developing adaptive content for performance elements. Using this methodology for the two performance elements in the RAES yielded the following results.

Level difficulty Via the performance element methodology:

1. Number of interactions taken to complete a scenario and amount of time taken to complete a scenario.

2. To alter the number of interactions a player can perform before receiving assistance.

3. Number of interactions taken to complete a scenario is linked to the level difficulty element.

4. Use formula 3.4 to calculate the CIM and formula 3.5 to calculate the CIT.

Time pressure Via the performance element methodology:

1. Number of interactions taken to complete a scenario and amount of time taken to complete a scenario.
2. To alter the amount of time that can elapse before giving the player assistance.
3. Amount of time taken to complete a scenario is linked to the level difficulty element.
4. Use formula 3.6 to calculate the CTM and formula 3.7 to calculate the CTT.

7.2 Evaluation of the methodology

With the utilisation methodology established, the next step involved evaluating the methodology's effectiveness at accomplishing it goal of providing game designers with instructions to incorporate the adaption algorithm with educational games. It was reasoned that an evaluation comprising the data and/or reflections of other game designers concerning the utilisation methodology could aid in validating it's effectiveness. The techniques used for conducting such an evaluation and the results yielded are explored below.

7.2.1 Purpose and aim

The goal of this evaluation was to determine the extent that the utilisation methodology, explained in Section 7.1, could instruct educational game developers to incorporate the algorithm with educational games to make them adaptive. An educational game developer is defined as someone in either academia and/or the entertainment industry that has worked on a game(s) whose primary focus is to educate its players. This evaluation was aimed at such educational game developers for two reasons. Firstly, they are the target audience for the utilisation methodology. Thus their opinions are directly relevant. Secondly, it was reasoned that they would possess experience with developing educational games and/or their own methodologies for educational game design. Such experience could give greater weight to their opinions and insights regarding my utilisation methodology. Therefore, this evaluation was designed to test the following hypothesis:

H_0 = The utilisation methodology does not provide sufficient instruction for educational game developers, who are looking to improve player learning through adaptivity, to incorporate the adaptation algorithm with educational games.

H_1 = The utilisation methodology provides sufficient instruction for educational game developers, who are looking to improve player learning through adaptivity, to incorporate the adaptation algorithm with educational games.

The following section will explain and justify the evaluation technique used.

7.2.2 Experimental design

Using the works of Arachchilage and Love (2013) and Shi and Shih (2015) as a template, the evaluations of educational games and their accompanying design methodologies were conducted with the use of questionnaires. Both consisted of a five point Likert scale where 5 indicated strongly agree and 1 indicated strongly disagree. Hence similar questionnaires were used for this evaluation as well.

Building on this, the option of doing a semi structured interview was also provided. This was done as it would allow for the questions from the questionnaire to be asked while also leaving room for follow up questions, which could provide additional subjective data. Thus, the same pool of questions and Likert scale scoring, were used for both the interviews and questionnaires.

The approached game designers were given a choice between conducting the evaluation session with a semi-structured interview or answering the questionnaire. Using the works of Arachchilage and Love (2013) and Shi and Shih (2015) as a basis, 10 questions were devised for the interviews and questionnaires.

These 10 questions were geared towards the hypothesis of ascertaining if the utilisation methodology provides sufficient instruction for educational game developers to incorporate the adaptation algorithm with educational games. Hence the questions pertain to the methodology's overall purpose and four of the five steps of the utilisation methodology. Questions pertaining to step 4 of the methodology (coding the algorithm and input/output data structures) were omitted due to step 4 being a straight forward recreation of code and data structures, both of which have pseudo-code provided as a reference.

Questions 1 and 10 pertain to the overall purpose of and perceived effectiveness of the utilisation algorithm at instructing game developers. Questions 2 and 3 pertain to step 1 (educational game compatibility), which aims to allow game developers to quickly discern whether the algorithm would be applicable to the educational game they are developing through specifying requirements clear enough to filter out games that are incompatible. Questions 4 and 5 pertain to step 2 (educational scenario development), which aims to let game developers quickly understand the purpose, structure and how to develop their own educational scenarios. Questions 6 and 7 pertain to step 3 (coding the algorithms triggers), aimed to instruct game developers in discerning how to place the algorithm's two triggers in a game of their choice. Questions 8 and 9 pertain to step 5 (developing adaptive content), whose purpose is to make game developers understand how to distinguish between preference and performance elements and be able to develop adaptive content for the elements they have chosen.

Furthermore, in keeping with the interview option leaving room for follow up questions from the original 10, the questionnaires also included space for game designers to comment on each of their answers. This was done in an effort to gain as much data as possible regardless of the choice of evaluation. The 10 questions comprised the following:

1. The purpose of the methodology given is concise and clear. Why?
2. The compatibility requirements allow for compatible games to be easily identified. Why?
3. The compatibility requirements make the functionalities needed to turn incompatible games into compatible ones clear. Why?
4. The concept, purpose and stages of educational scenarios are clear. Why?
5. Game designers trying to create educational scenarios akin to the ones in the this study will be able to do so using the educational scenario creation methodology. Why?
6. The player interaction trigger methodology paints a clear picture of how to place the algorithm's player interaction trigger in other games. Why?
7. The non player interaction trigger methodology paints a clear picture of how to place the algorithm's non player interaction trigger in other games. Why?
8. The distinction between and the method to classify preference and performance elements is clear. Why?

9. The preference element methodology and performance element methodology, along with their provided examples, illustrate how to develop adaptive content for any preference and/or performance element in general. Why?
10. Assuming that the design principles and logic behind the algorithm, the methodology provided could help other game designers in utilising the algorithm to make other educational games adaptive. Why?

Please see Section 9.3 to view the questionnaire and interview sheets used for this evaluation.

7.2.3 Measure of success

Concerning the measure of effectiveness of the results, both Arachchilage and Love (2013) and Shi and Shih (2015) used the mean scores of the Likert scale as a basis for their analysis. However, these two projects had students as their target audience. The abundance of students willing to participate in their studies lead to both projects having more than 100 participants each take part in their evaluations. This was reasoned to strengthen the reliability of the Likert scale scoring. But, obtaining questionnaire responses and/or conducting interviews for a similar number of educational game designers was deemed too large and unrealistic for the scope of this evaluation. This was due to the fact that the target audience of this evaluation (educational game designers) are not as abundant as students.

Given the smaller pool of participants for this evaluation, the Likert scale scores and by extension any mean values derived from them would therefore be less credible. However, it was reasoned that the Likert scale for all 10 questions be kept in order to guide the participants focus towards the critical points of each question. Hence, the answers gathered from the comments and/or follow up questions from the base questions was taken as the actual measure of success for the evaluation. The answers obtained and their accompanying analysis and reflections are presented in Section 7.2.5.

7.2.4 Recruitment procedure and ethical issues

The typical recruitment procedure for an evaluation regarding software would be to make members of the targeted audience aware of the evaluation, establish a line of contact with them, explain the purpose and expectation of the evaluation, obtain their consent to participate and then run the evaluation session. As the target audience for the utilisation methodology evaluation were educational game designers, who would be over the age of 18, they are not classified as a vulnerable group. Thus, recruitment procedure was straight forward.

Potential candidates were identified either online or through recommendations from other researchers. They would be emailed directly to establish a line of contact before explaining the purpose and expectation of the evaluation. From there consent to participate and have the results published were obtained and the evaluation sessions were run. Anonymity was maintained for all participant names, contact details and any other personal information. Furthermore, as this evaluation made use of interviews and questionnaires, all evaluation sessions were conducted online for the convenience of all parties.

7.2.5 Results

A total of six participants engaged in the evaluation. This section presents the results obtained from each participant, a review of the obtained results for patterns (positive, negative, neutral and suggestions), and conclusions and potential future work regarding the utilisation methodology. As explained before, the answers gathered from the comments and/or follow up questions from the base questions, were taken as the actual measure of success for the evaluation, as the low number of participants made taking the average scores from the Likert scale less credible.

Furthermore, all participants were additionally queried regarding there background and specific roles in the game development process, as it was reasoned that this could provide context for their answers. The responses to this line of querying are listed under the "Additional" query rows in the tables below.

7.2.5.1 Participant 1

Participant 1 was an academic game designer focused on the aspects of educational game design. The results from Table 7.1 reflect this, as aspects pertaining to pedagogy and game design including the methodology's overall purpose (Question 1), game compatibility (Question 2) and educational scenarios (Questions 4 and 5) received high scores of between 4 to 5 and positive comments. Conversely, Participant 1 gave lower scores of between 1 and 2, expressed concern and experienced difficulty with the technical software engineering components of the methodology, namely regarding the algorithm's triggers (Question 6 and 7).

Participant 1's inclination towards pedagogy and game design, and difficulty with technical implementation, resulted in neutral scores of 3 with regards to making non-compatible games compatible (Question 3), distinguishing between and developing adaptive content for both types of elements elements (Questions 8 and 9) and the overall effectiveness of the methodology (Question 10). An inspection of the comments attributed to these questions further bolstered this trend, with Participant 1 outright expressing clarity regarding pedagogical aspects of the methodology and difficulty regarding the technical software engineering components. Hence, Participant 1 suggested that including a low-level explanation, in plain English, of the utilisation methodology being applied to a game could serve as a concrete example to game designers who are less inclined towards software engineering.

Table 7.1: The answers and and comments given by participant 1.

Question	**Score**	**Comment**
1	4	It's quite clear that this methodology is meant to instruct others on incorporating your algorithm with other games.
2	5	The requirements were very technically described and straight forward.
3	2	As game designers possess varying skills, those who are not as adept at software engineering might struggle with the suggestions provided.
4	5	The specification and structure of educational scenarios was very clear and concise.
5	5	It was clear, from a design perspective, as it took the form of an ordered list of steps.
6	1	I understand the purpose of this trigger and the instructions to code it on a higher level, but actually coding into a game is too technical for me.
7	1	Similar to the last trigger, I understand the purpose of this trigger and the instructions to code it on a higher level, but actually coding into a game is too technical for me.

8	3	It's a good theory and the descriptions from your implementation of Minecraft in the user evaluation are clear, but it might be tougher to generalise this to games that are different from minecraft.
9	3	It's a good theory and the descriptions from your implementation of Minecraft in the user evaluation are clear, but it might be tougher to generalise this to games that are different from minecraft.
10	3	The pedagogical methods are sound are easy to follow, but the technical implementation portions are a little tough to follow.
Additional	n/a	These views should all be taken in the context of someone eager more towards educational game design compared to software engineering. The pedagogy and design approach to algorithm seems very sound, but challenge of applying it to other games is less so. Perhaps a low level explanation of how to implement everything in plain English would benefit designers not as inclined towards software engineering.

7.2.5.2 Participant 2

Participant 2 was an academic game designer whose background lay in software engineering. The results from Table 7.2 reflect this, as aspects pertaining the methodology's purpose (Question 1), game compatibility (Question 2), making non-compatible games compatible (Question 3), and the algorithm's triggers (Question 6 and 7) received high scores of between 4 to 5 and positive comments. Conversely, Participant 2 gave low scores of 1 to 2, expressed concern and experienced difficulty with pedagogical aspects such as educational scenario purpose and design (Questions 4 and 5), and distinguishing between and developing adaptive content for both types of elements (Questions 8 and 9). Participant 2 explicitly explained that he/she would not be able to apply the aforementioned pedagogical concepts towards a different educational game than Minecraft.

Although Participant 2 gave a high score of 4 to the overall effectiveness of the methodology (Question 10), there were some criticisms. Participant 2 explained that while it gives game designers a clear set of instructions to approach adaptivity in games, applying it to games that are radically different from Minecraft (the game used in this study) might prove challenging. Hence, Participant 2 suggested that including more examples of the utilisation methodology being applied to radically different games could aid in game developers in extrapolating the generalised steps for their own purposes.

Table 7.2: The answers and and comments given by participant 2.

Question	Score	Comment
1	5	It is definitely clear and it did not feel rambling.
2	4	The general ideas were described well but some of the concepts, such as educational scenarios could have used additional examples. Because you only used the one game's example, it is not always clear how to apply the concepts to a very different game. For example, it is not clear how to define progress in a game with only "final" actions such as one that is trying to teach a specific skill.

3	5	They are definitely clear. The best way to execute them isn't always obvious. There may be a way to present educational method, but is it an effective way?
4	4	Same concern as before. A radically different example would help.
5	3	It's not the methodology. It's the deeper understanding of what you are trying to accomplish that is the problem. The methodology gives things to think about but it is the pedagogy and content that tells you how to apply it. It's how to present the material that is hard.
6	5	This was pretty unambiguous.
7	5	Limited things to worry about here. Clear.
8	3	I thought this was the least clear part. The metrics are easy to understand but the underlying differences and how they impact people is harder to understand.
9	2	I don't think that I could apply it to a game that is radically different from the one that you describe.
10	4	Clearly gives the designers a way to approach the problem but in order to use it fully you need to understand what the misunderstandings are and build solutions that address them. That's the hard part!
Additional	n/a	I lean more towards the educational game design in practice, but my background is in software engineering.

7.2.5.3 Participant 3

Participant 3 was a game designer from industry whose background lay in design choices for games in general. Due to this, he/she admitted to comparing the utilisation methodology to a workflow akin to one found in a professional game studio. As such, Participants 3's scoring for all questions were lower, between 2 and 3, as shown in Table 7.3. When examining Participant 3's comments, two patterns were discerned.

The first pattern was a concern with the compatibility of the utilisation methodology in the context of a professional game studio work flow. This is reflected by the comments concerning the methodology's purpose (Question 1), educational scenario purpose and design (Questions 4 and 5), and the algorithm's triggers (Question 6 and 7). The second pattern concerned issues with generalising the implementation of the methodology to other games. This is reflected by the comments concerning game compatibility (Question 2), making non-compatible games compatible (Question 3), distinguishing between and developing adaptive content for both types of elements (Questions 8 and 9) and the overall effectiveness of the methodology (Question 10).

When given clarification of the academic game development nature of this study, Participant 3 relented on the concern regarding the utilisation methodology being used in the context of a professional game studio. However, concerning the issues with generalising the implementation of the methodology, Participant 3 suggested improving upon the methodology by fleshing out the Minecraft implementation done for the user evaluation to help game designers better extrapolate what to do for other games. This could take the form of a detailed low level walk through of everything done for developing the custom version of Minecraft used for the user evaluation. Furthermore, it was postulated that the utilisation methodology could also be expanded and/or restructured to mirror a professional game studio's workflow to further increase the methodology's generalisability in the industrial context.

Table 7.3: The answers and and comments given by participant 3.

Question	Score	Comment
1	3	It's mostly clear. The biggest point of confusion for me is where in the development process this is designed to fit. The scenario in which a game designer is taking an existing game and making it adaptive is a bit confusing to me. The process of adding adaptation seems to require a lot of design work and access to game logic which isn't often available with existing games. Is this for games in development? It feels like it is, but I'm not certain.
2	2	From the description, my general understanding of what makes a game compatible, is one that can be modified and one that has designed to engage the player in educational scenarios. The methodology to identify games feels like it would only reject obviously incompatible games (ones without the necessary ability to either change or augment the code). The evaluation of the education scenarios doesn't seem to address the question of whether or not the techniques could be made adaptive in a meaningful way, which feels like the harder and more important question. I have a lot of confidence that almost any game could have an event system added to it that's tracking the player's behavior, but I'm not confident effective adaptive elements could be added to most games.
3	2	I think it's pretty clear how you would technically make an incompatible game compatible: get access to the source code and development tools. It's not as clear how a game might fail the educational scenario test and then be made compatible. The educational scenario feels like a fundamental part of the design and making it compatible would amount to a redesign of the game. It's also not clear where in the process this fits. Is this identifying an existing game, or is this identifying a game in development? The methodology states its to be applied to existing games, but the inclusion of guidance on how to design scenarios makes me unsure of this.
4	3	I think I understand educational scenarios. It feels like it's a breakdown of the educational strategy the game incorporates into its design.

5	2	There is a structure described, but it doesn't really connect to a game development process that I would expect developers to follow. The details of understanding learning objectives, connecting them to learning strategies, developing game design and structure to complement and support those strategies, and prototyping and iterating those designs are all parts of the process that more fundamentally determine how game designers create educational experiences. I could see this methodology being used as a checklist of sorts, but it doesn't feel like it would be used to guide game development.
6	2	I'd be concerned giving this sequence of steps to a development team to follow. In particular, step 1a feels like it's presenting a simple idea that's actually incredibly complicated. What defines a player interaction? Is it just an enumeration of input events like left mouse down, up arrow release, etc. ? Or is it higher level interactions like jump, run, pick up, shoot, interaction, etc.? Or is even higher level ideas like jump on, talk to, craft, respond, attack, etc.? Or do you work on enumeration meaningful events like "answers question" or "scores a goal"? Even the example given for minecraft feels like a mix of input triggers (keyboard button presses and mouse movement) and higher level interactions (talk to NPC and select dialogue). At the very least I would expect Step 2b to drive the development discussion and generate a list of meaningful interactions rather than trying to develop an exhaustive list.
7	3	I mostly understand this. My biggest point of confusion is how to draw a line between player interaction triggers and non-player interaction triggers. These feel like high level events that are still connected to player interaction (for instance, completing a level will be tied to the action a player takes to trigger level completion). Similar to the challenge of defining what sort of events/inputs you are monitoring, I would expect some confusing in defining the difference between what is a player interaction and what isn't.
8	2	I had to revisit this several times and I'm not sure I really get it. I'm confused if the preference is a design technique where the designer categorises players and their playing preferences and then designed elements to align with those styles, or if it's just a collection of interaction techniques. I'm also confused with when the distinction between performance and preference happens. Does a designer categorize things after the fact, or is it guidance for creating the design?

9	2	I think I got a decent sense of the basic idea of tracking data and using that to detect patterns in player behavior. What I don't really have a good sense for is how to actually change an element in response to the data. Difficulty ramps, hint systems, interactive tutorials, and such are all very specific to the problems they're solving and I don't really understand how this process helps me make those systems more adaptive. I really wanted a specific example to help illustrate this.
10	2	I feel like the high level idea of identifying events, connecting those events to important player behaviors, and changing the game based on measured patterns is a fine overall approach to making a game adaptive. But using the steps as presented seems to clash with the practice of game development–particularly the iterative process and the tight collaboration between design, educational expertise, and development.
Additional	n/a	I am a game designer from industry, so I was viewing this from the context of a work flow that would be found in a professional game development studio. But now that you have clarified that this methodology is meant to be taken in the context of an academic setting with a single person or small development team, I would improve upon this methodology by fleshing out the Minecraft implementation you did. A low level walk through of everything you did to develop your game would be a of great help to extrapolate what to do for other games.

7.2.5.4 Participant 4

Participant 4 was an academic game designer whose background lay in geographic information science. Participant 4's role in the game development process involved the teaching of educational content and general decisions for game design. The results from Table 7.4 reflect this, as aspects pertaining to pedagogy and game design including the methodology's overall purpose (Question 1), game compatibility (Question 2), educational scenarios (Questions 4 and 5) and classifying performance and preference elements (Questions 8) received high scores of between 4 to 5, and positive comments. Conversely, Participant 4 gave lower scores of 2, expressed concern and experienced difficulty with the technical software engineering components of the methodology, namely regarding the algorithm's triggers (Question 6 and 7).

Participant 4's inclination towards pedagogy and game design, and difficulty with technical implementation, resulted in neutral scores of 3 with regards to making non-compatible games compatible (Question 3), developing adaptive content for preference and performance elements (Questions 9) and the overall effectiveness of the methodology (Question 10). An deeper dive into the comments attributed to these questions further bolstered this trend, with Participant 4 having fewer issues grasping the pedagogical aspects of the methodology as compared to the technical software engineering components. Hence, Participant 4 suggested that including a more examples of the algorithm being applied to a variety of games starkly different from one another could be a promising method for making the methodology more generalisable.

Table 7.4: The answers and and comments given by participant 4.

Question	Score	Comment
1	4	It's quite straight forward that your methodology is meant to give other game designers instructions to use your algorithm with other games.
2	4	It's quite easy to tell if a game is compatible with your algorithm by referring to the compatibility requirements.
3	3	You give the gist of this and provide some suggestions, but there isn't a step by step approach for designers who may need more detail.
4	5	The description and stages of educational scenarios was clear and easy to understand.
5	4	Presenting it as a list of steps made it easy to follow, but giving more examples using games than other Minecraft would improve this.
6	2	I understand that your algorithm has to be triggered in some way, but I'd struggle to implement this trigger in a game different to Minecraft.
7	2	Same comment as the last trigger, I get the idea of what you are trying to do, but actually implementing it in a game would be a challenge for me.
8	4	I understand the descriptions from your custom version of Minecraft, but it's not completely clear how to carry this over to other games.
9	3	I see this in a similar light as the previous one, but the inclusion of player metrics makes applying this to other games more difficult than the other type of element.
10	3	I'd say it's a good starting point, especially with the educational theory portions, but implementing your algorithm especially with other games might prove more challenging. Maybe if you provide more examples of your algorithm being implemented with starkly different games, then game designers might be able to better extrapolate what to do.
Additional	n/a	My background is in geographic information science and I have had experience making serious games for urban planning. I was involved with the educational content and game design aspects of the development process.

7.2.5.5 Participant 5

Participant 5 was an academic researcher, with a focus on computer graphics, who had experience in making educational games in collaboration with a variety of industrial partners. The results from Table 7.5 reflect this, as technical aspects pertaining to game compatibility (Question 2), making non-compatible games compatible (Question 3), and the algorithm's triggers (Question 6 and 7) received high scores of between 4 to 5 and positive comments. Conversely, Participant 5 gave either low or neutral scores of 2 and 3 with regards to the application of pedagogical aspects. This included topics such as educational scenario purpose and design (Questions 4 and 5), and distinguishing between and developing adaptive content for both types of elements (Questions 8 and 9). Participant 5 claimed to understand the theory behind the pedagogical aspects but expressed concern with the generalisation of implementing these aspects to games whose mechanics are different to Minecraft.

Although Participant 5 gave a high score of 4 to the overall effectiveness of the methodology (Question 10), there were some criticisms. Participant 5 explained that while the principles behind the methodology are clear, the exact steps to implement it are less so. He/She clarifies that designers would be able to apply the utilisation methodology to Minecraft and other games that are mechanically similar. However, this generalisability may not carry over to games which are different from Minecraft. Hence, Participant 5 suggested including more examples of the utilisation methodology being applied to radically different games from Minecraft to make the methodology more generalisable.

Table 7.5: The answers and and comments given by participant 5.

Question	**Score**	**Comment**
1	5	I found it very clear.
2	5	Yes they were quite straight forward and easy to check in games.
3	4	Everything was clear with the exception of implementing educational scenarios in games. I'm not confident that the way forward is clear for games that are greatly different from Minecraft.
4	3	The purpose and stages of educational scenarios are clear in theory but in practice it's more challenging to imagine how it can be applied to more types of games.
5	3	Similar comment to before. For games that are mechanically or functionally similar to Minecraft, the list of steps would work fine. But for games that are different, applying this is not very clear.
6	5	Very straight forward to me.
7	5	Same as the other one, it was straight forward.
8	4	The difference and how to sort elements into these two categories are quite clear but I think the idea of player metrics could be given more detail by including more examples using different games than Minecraft.
9	2	I didn't find this clear. It's a running trend that when thinking about implementing your system to very different games, the list of steps you provided don't always make sense. You can't assume that games similar to Minecraft will always be used.
10	4	Overall it's quite a clear methodology and i can understand the challenge of trying to come up with something that works across games in general, but I'd recommend including more examples of this methodology being applied to a variety of games so that others can discern what to do for the educational portions.
Additional	n/a	I focus on computer graphics, virtual reality, scientific visualization and education. I've made educational games in collaboration with a variety of industrial partners. Yes I'd say I am well versed with software engineering components of game design.

7.2.5.6 Participant 6

Participant 6 was a university lecturer, for programming courses related video game development, who also had a wealth of experience with working with modern (2023) video game engines. Hence, Participant 6 was more geared towards the software engineering aspect of game design.

The results from Table 7.6 reflect this, as aspects pertaining to the methodology's purpose (Question 1), game compatibility (Question 2), making non-compatible games compatible (Question 3), the concept and purpose of educational scenarios (Question 4), and the algorithm's triggers (Question 6 and 7) received high scores of between 4 to 5 and positive comments. Conversely, Participant 6 gave low scores of 1 to 2, expressed concern and experienced difficulty with pedagogical aspects such as designing educational scenarios (Question 5), and distinguishing between and developing adaptive content for both types of elements (Questions 8 and 9). Participant 5 explicitly explained that he/she would not be able to apply the aforementioned pedagogical concepts towards a different educational game than Minecraft.

Participant 6 gave a neutral score of 3 to the overall effectiveness of the methodology (Question 10). Participant 6 explained that while the methodology is a good start, more examples of it being incorporated with different games, especially with making adaptive content for the elements, would improve its generalisability.

Table 7.6: The answers and and comments given by participant 6.

Question	**Score**	**Comment**
1	4	It's clear what you intend the methodology to be used for, but whether it succeeds at its purpose is another matter.
2	4	I understood the compatibility requirements without any issue.
3	4	I also understood how to make games compatible, but I'll comment that not all games can accommodate educational scenarios. You're limiting yourself here.
4	4	I understand the concept of educational scenarios you put forward just fine.
5	2	I understand how you modified Minecraft and made educational scenarios for it, but what you have presented does not effectively explain how to achieve the same thing for different games. To bring up something you said previously, modifying Fortnite (a popular online shooting and building game) to achieve the same result would require a different process from what you described.
6	5	Yes, it's using the game engine to monitor for specific actions done by players.
7	5	Yes, it's like an internal game clock and the engine monitoring for events.
8	2	The concept of player metrics is clear, but I don't think classifying the game elements the way you describe is so straight forward. It's possible that these elements could have different effects on different players. So classifying these elementsr might actually be different from player to player.
9	1	I understand the description of what you did for your game, but I don't think it can be used for any game other than Minecraft.

10	3	It's a good start, but I think you need to include more examples of your system being used with different games, especially with making adaptive content for the elements. That way, how to use your algorithm with other games will be a lot be clearer.
Additional	n/a	I teach game AI programming and courses in game animation. I've also been involved in university research for motion capture and virtual reality projects. In terms of software engineering for video games, I have a lot of experience using modern triple-A game engines to make games myself.

7.2.5.7 Discussions and conclusions

Although a multitude of feedback and suggestions for improvement were obtained, H_0 could not be rejected. This was due to the low sample size of 6 game designers in comparison to researchers such as Arachchilage and Love (2013) and Shi and Shih (2015), who had more than 100 participants each. These researchers' large sample size allowed for the mean scores of the Likert scale used, to be a statistically significant basis for their analysis. This was not the case for the utilisation methodology's evaluation. Despite this evaluation's shortcoming, it yielded valuable subjective feedback indicating that the methodology in its current form serves as a promising base, while also bringing to light the methodology's weaknesses and possible solutions. These are presented below.

Upon examining and reflecting on the all the feedback obtained, it was observed that the aspects of the utilisation methodology that participants found clear or struggled with, depended on their individual background and strength(s) within the field of game design. This suggested that ease and/or difficulty that game designers could experience when utilising the methodology could be subject to the context of their role in game design. Hence, varying suggestions to improve and make the utilisation methodology more generalisable were proposed based on said contexts.

Three major contexts were identified in this evaluation. This included an individual game designers skill in pedagogy and educational game design, technical software engineering skills, and whether that individual developed games for an academic and/or industrial purpose. Game designers who were more inclined towards pedagogy and educational game design, such as Participant 1 and 4, struggled with the implementation of the software engineering components of the methodology. Two possible remedies for such game designers were proposed. The first revolved around expanding the utilisation methodology to the include a more detailed explanation, written in plain English, of what was done to the custom version of Minecraft to turn it into a compatible game, the coding of both triggers and the development of adaptive content for both preference and performance elements. The second, was the idea of including more examples of the aforementioned technical aspects being applied to a variety of games starkly different from one another, so that game developers could better extrapolate the process.

On the other hand, game designers more inclined towards software engineering, such as Participant 2, 5 and 6, experienced difficulty with regards to applying the pedagogical principles to games other than the custom version of Minecraft used in the user evaluation. Hence, it was suggested that generalisability could be improved by including more examples of educational scenario design and implementation, and the development of adaptive content for both preference and performance elements, being applied to a variety of games starkly different from one another.

The final context identified, involved game designers from industrial game development studios, such as Participant 3. Participant 3 compared the utilisation methodology to that of to a workflow typically found in a professional game studio (which was described as radically different from the utilisation methodology), resulting in poorer scores and confusion in terms of implementation. This exposed a weakness of the utilisation methodology as being more geared towards game design in an academic setting and thus not translating as well to an industrial setting. Hence, another direction to expand the utilisation methodology could be to convert/translate it from its present form to one more akin to a professional game studio's workflow.

7.3 Summary

A utilisation methodology meant to instruct other game designers on how to incorporate the adaptation algorithm with other educational games, was devised for the purpose of expansion and/or future work. The utilisation methodology is split into five steps. First, is to ensure that the educational game chosen/made to be paired with the algorithm, possesses the necessary functionality to be compatible. Second, is the design of educational scenarios. Third, is to make sure that the algorithm is triggered at the correct instances. Fourth, is coding the algorithm and data structures using the pseudo-code in Section 4.4. Fifth, is to develop element specific content for each of the elements chosen to be made adaptive using the specified methodologies. Ensuring that each of the five stages of the utilisation methodology are fulfilled will allow for the algorithm to function with any educational game.

The effectiveness of the utilisation methodology at instructing other game designers was also evaluated by obtaining the thoughts and/or opinions of other game designers. This was achieved by reaching out to game designers via email, providing them with the utilisation methodology and the supplementary background material required to understand it, and having them take part in a 10 question interview or an open ended questionnaire concerning the effectiveness of the methodology. Six game designers participated in the evaluation. Due to the lower number of participants when compared to studies that draw statistically significant conclusions based on Likert-scale scoring, at least 100 participants, the Likert-scale scores for this evaluation could not yield statistically significant results. Despite this evaluation's shortcoming, it yielded valuable subjective feedback indicating that the methodology in its current form serves as a promising base, while also bringing to light three of the methodology's weaknesses and possible solutions. These are as follows:

Game designers who are more inclined towards pedagogy and educational game design might struggle with implementing the technical software portions of the methodology. This could be remedied with a more detailed explanation, written in plain English, of the technical portions being implemented in one or preferably a variety of games to serve as example(s).

Game designers who are more inclined towards software engineering may have difficulty implementing the pedagogical and educational game design aspects of the methodology. The suggested solution for this was a more detailed explanation of the pedagogical portions being implemented in one or preferably a variety of games to serve as example(s).

Game designers who are more familiar with a work flow of an industrial game development studio might experience difficulty following the structure of the methodology and discerning how to best implement it. It was reasoned that a possible expansion for the methodology could be to convert/translate it from its present form to one more akin to a professional game studio's workflow, in order to improve its generalisability.

It was reasoned that implementing one or more of the three solutions above could further bolster the effectiveness of the utilisation methodology at instructing game designers to incorporate the adaptive algorithm with other games. With the utilisation methodology established and evaluated, the next chapter presents a reflective discussion of all work done, details possible future work and draws a formal conclusion.

Chapter 8

Conclusions

This chapter reviews the work done and results achieved against the four research contributions and research question detailed in Chapter 1. Furthermore, it details the limitations of this study, possible future work and draws a formal conclusion.

8.1 Review of work

This study aimed to answer the research question, *"How can educational games in general be made adaptive to as many individual players as possible so as to best improve their learning gains?"*. We believe that this question was answered. The answer comprised the extraction of a framework (from academic literature) to make games in general adaptive, the use of appropriate techniques and principles to create an algorithm to make adaptations, the choice of an educational game able to entertain its players as an evaluation platform to encourage a wider player base, and the use of the utilisation methodology to incorporate educational games with the algorithm. Each of the four components align with the four research contributions put forth in Chapter 1 and are delved into greater detail below.

8.1.1 Research Contribution 1

The first contribution was the extraction of a framework, from academic literature, to make games in general adaptive. The adaptive framework was split into the four adaptivity issues of what to adapt, when to adapt, how to adapt, and how to collect data for the first three issues. This framework was extracted from literature pertaining to pedagogy, pedagogy in educational games, educational game design, and adaptivity in both entertainment and educational games.

8.1.2 Research Contribution 2

The second contribution was the creation of the algorithm that monitors players, updates player models and executes adaptations. This process was split into the algorithm design specification and the algorithm implementation. The algorithm design specification explored and justified solutions and/or techniques to address the four issues posed by the framework. These included:

What to adapt. This was resolved by identifying game elements whose adaptivity could improve learning in players. This was done by breaking up the educational process found in educational scenarios, into four stages to determine the element(s) that affect the educational process. Next, four adaptability requirements were devised/identified and combined with the three motivating requirements to filter the elements that affect the educational process, to those whose adaptivity could benefit a player's learning. Elements that fulfill all requirements are deemed as adaptable and part of the adaptable element set (AES). From the initial 12 elements, six would fulfill all requirements. These six elements would be referred to as the recommended adaptable element set (RAES). A method of modelling any element in the AES across three dimensions was identified. The first dimension detailed the kind of element, preference or performance. The second dimension pertained to the type of adaptivity the element would encompass, active or passive. The third classified when the adaptation of the element should occur, at predefined points or at any point during the game. Lastly, a survey of adaptive educational games yielded a framework of options for adaptivity for the RAES.

When to adapt. This was resolved through the devising/identifying adaptation conditions. An adaptation condition is defined as a state during the playing of an educational game, where the adaptation of an element(s) in the AES could potentially benefit a player's learning. The adaptation conditions were devised/identified by first deriving three situations, within educational games in general, where adaptation could aid a player's learning. These were referred to as potential adaptation situations. The three potential adaptation situations were then combined with the algorithms two triggers of player interactions and player non-interactions to establish the five adaptation conditions.

How to adapt. This was resolved through the technique of player modelling. A player model is defined as an internal model of the player's proficiency at the game, which is used to reason over and allow for educational adaptation. Although there exist multiple techniques for player modelling, rule based logic was identified as the most appropriate method for this study due to it best fulfilling the motivating requirements. Using rule based logic entailed identifying formulae to adapt preference and performance elements, and establishing a rule base to determine the course of the adaptation of AES to take, when one or more of the adaptation condition(s) have been met.

Data collection. This was resolved through the use of a player monitoring technique. More specifically, gameplay player experience modeling (PEM) was identified as the most suitable method due to fulfilling all adaptability and motivating requirements. Furthermore, the two algorithm triggers of detecting player interactions and non-interactions were identified and devised respectively.

The algorithm implementation drew from the design specification to detail the implementation of the adaptive algorithm. This included a high level description of design choices and functionality, input and output data structures, and pseudo-code illustrating how the algorithm should be coded. The high level functionality portion split the algorithm into the three stages of player monitoring, player modelling and executing adaptations. Input data structures included the include the data set pertaining to a player's behaviour (PBS); statistics of previous adaptations made (ASD); static benchmark data specific to each educational scenario (SBD); a data set pertaining to the educational game's running state (RSD) and a set of data representing the player's ability at completing educational scenarios while each of the differing options of the AES are active (PM). Output data structures included the updated player model and command to the game engine to execute adaptations deemed beneficial (AC). Lastly, generalisable pseudo code and low level explanations to implement the algorithm in any educational game were created.

8.1.3 Research Contribution 3

The third contribution consisted of the system testing and user evaluation of the algorithm. The system testing was conducted by getting simulated to players to behave like actual players going through educational scenarios while the algorithm monitored them, updated their player models and made adaptations. Simulated player scripts (SPS) were generated to serve as instructions for simulated players to behave like actual players. Reference and false player models were also generated to compare against the algorithm's output player models as a means to ascertain the accuracy of the output. The algorithm's output player models achieved an overall accuracy of 82.5% when compared to the reference model, indicating that the algorithm could in theory accurately identify actual player models. The system testing also showed that the accuracy of the algorithm's output increased with more educational scenarios played, and the algorithm was unable to generate accurate player models from players who possessed above average knowledge of the educational content due to needing little to no adaptation.

The user evaluation made use of a custom version of Minecraft as a evaluation platform, aimed at teaching the identification and credibility of evidence section of critical thinking. Player learning gains were measured by taking the difference in scores between the post-game and pre-game test. 23 participants in total, took part in the evaluation. Using Minecraft allowed for the entertainment factors of educational game design to be fulfilled, leaving more focus to be directed towards the creation of six learning scenarios and adaptive content for four elements from the RAES. Although statistically positive results were achieved, a concrete conclusion of the algorithm's effectiveness could not be reached due to the following four factors:

Preliminary study. The evaluation conducted was a preliminary one due to time and resource constraints. The unusually positive results from this preliminary evaluation suggest a high degree of noise and/or artefacts. Hence, a more thorough evaluation would have to be conducted to shed further light on these preliminary results.

Adaptive content for a partial RAES model. The evaluation did not include adaptive content for the entire RAES model. Hence, any conclusions that could be drawn from this evaluation, positive or negative, would be in reference to a partial model. Content for the full model would need to be developed and evaluated before any concrete conclusions regarding the effectiveness of the algorithm can be drawn.

Teaching approach used. An instructional approach was use to teach the educational content as opposed to a programmatic approach. While not as accurate as a programmatic approach, an instructional approach is still described as worth doing when resource and time constraints do not allow for a programmatic approach to be used. Hence more accurate results regarding the effectiveness of the adaptation algorithm could be brought forward by utilising a programmatic approach to teaching in future user evaluations.

Sample size. The studies involving instructional approaches to teaching made used of participant samples sizes of 30 or more. Thus, aiming for a similar or higher number of participants in future evaluations may lead to more accurate results concerning the algorithm's effectiveness.

Despite these limitations, the results of both evaluations indicate that the identification and modelling of adaptable elements, techniques used to build the algorithm and the methodology used to create educational scenarios and adaptive content, could have been done correctly. Hence the algorithm and its evaluation platform could serve as a good base for future research. However, a more thorough evaluation (using any of the four suggestions above) would have to be conducted before any concrete conclusions of the algorithm's effectiveness can be drawn.

8.1.4 Research Contribution 4

The fourth contribution was the methodology for utilising the adaptation algorithm with any educational game. The utilisation methodology is split into five steps. First, is to ensure that the educational game chosen/made to be paired with the algorithm, possesses the necessary functionality to be compatible. Second, is the design of educational scenarios. Third, is to make sure that the algorithm is triggered at the correct instances. Fourth, is coding the algorithm and data structures using the pseudo-code in Section 4.4. Fifth, is to develop element specific content for each of the elements chosen to be made adaptable using the specified methodologies. Ensuring that each of the five stages of the utilisation methodology are fulfilled will allow for the algorithm to function with any educational game.

The effectiveness of the utilisation methodology at instructing other game designers was also evaluated by obtaining the thoughts and/or opinions of other game designers. This was achieved by reaching out to game designers via email, providing them with the utilisation methodology and the supplementary background material required to understand it, and having them take part in a 10 question interview or an open ended Likert-scale questionnaire concerning the effectiveness of the methodology. Six game designers participated in the evaluation. Due to the lower number of participants when compared to studies that draw statistically significant conclusions based on Likert-scale scoring, at least 100 participants, the Likert-scale scores for this evaluation could not yield statistically significant results. Despite this evaluation's shortcoming, it yielded valuable subjective feedback indicating that the methodology in its current form serves as a promising base, while also bringing to light three of the methodology's weaknesses and possible solutions. These included:

Game designers who are more inclined towards pedagogy and educational game design might struggle with implementing the technical software portions of the methodology. This could be remedied with a more detailed explanation, written in plain English, of the technical portions being implemented in one or preferably a variety of games to serve as example(s).

Game designers who are more inclined towards software engineering may have difficulty implementing the pedagogical and educational game design aspects of the methodology. The suggested solution for this was a more detailed explanation of the pedagogical portions being implemented in one or preferably a variety of games to serve as example(s).

Game designers who are more familiar with a work flow of an industrial game development studio might experience difficulty following the structure of the methodology and discerning how to best implement it. It was reasoned that a possible expansion for the methodology could be to convert/translate it from its present form to one more akin to a professional game studio's workflow, in order to improve its generalisability.

Although the utilisation methodology in its current was described as a promising base, implementing one or more of the three solutions above could further bolster its effectiveness. With the research question and four contributions reviewed, the next section details the limitations of this study and the opportunities for future work they present.

8.2 Limitations and future work

Although the results of this study indicated that it is a promising base for educational adaptivity, it suffers from four limitations, which also present possible directions for future research and/or expansion. These include:

Expanding the AES. New elements could be identified and added to the AES and/or new options for adaptation could be found for the RAES. Doing so could lead to more effective and/or a greater number of adaptations which could improve players' learning. This could be accomplished by reviewing more educational games and video games with adaptivity.

Choosing a different player modelling technique. Although rule based logic was the player modelling technique used in this study, six other methods were identified. These include fuzzy logic, Bayesian networks, hidden Markov models, decision trees, neural networks and genetic algorithms. Using any of these six player modelling techniques in place of rule based logic might be prove more effective and/or yield different outcomes.

Expanding the user evaluation. The full extent of the algorithm's effectiveness may not have been achieved due to the time and resource constraints the user evaluation suffered from. Hence the following four ways to run a more comprehensive user evaluation to determine the algorithm's effectiveness, were devised:

1. **Developing adaptive content for the full RAES model**. As explained before, the evaluation did not include adaptive content for the entire RAES model. Content for the full model would need to be developed and evaluated before any concrete conclusions regarding the effectiveness of the algorithm can be drawn.
2. **Using a programmatic approach to teaching**. An instructional approach was use to teach the educational content as opposed to a programmatic approach. The programmatic approach involves the blending of critical thinking into the background with the student's focus on a separate subject, as opposed to critical thinking explicitly being the subject. As programmatic approaches to teaching critical thinking yielded more significant improvement learning gains when compared to the instructional, it was reasoned that utilising a programmatic approach with the algorithm could also improve the algorithm's effectiveness.
3. **Running a time series intervention**. The user evaluation made use of a one-shot intervention teaching approach as opposed to the time-series intervention. As explained in Chapter 6, the time series intervention produces more accurate results when measuring learning gains due to conducting the intervention and measuring the learning gains of players multiple times to track changes throughout the course of the interventions. The one-intervention however, produces less accurate results due to conducting only one intervention and measuring the learning gains of players once. Thus, a more accurate result of the adaptation algorithm's effect on learning gains could be measured if future user evaluations make use of the time-series approach.
4. **Obtaining a larger sample size**. A user evaluation comprised a total of 23 participants. However, the studies geared towards teaching critical thinking, reviewed in Section 6.2.2, had at least 30, if not significantly more, participants in their user evaluations. Thus, aiming for a

similar or higher number of participants in future evaluations may lead to more accurate results concerning the algorithm's effectiveness.

Improving the utilisation methodology. The utilisation methodology's evaluation identified three weaknesses of and their possible solutions (Section 7.2.5.7). These include making improvements to the utilisation methodology for the following three subsets of game designers:

1. **Game designers who are more inclined towards pedagogy and educational game design than software engineering**. Providing a more detailed explanation, written in plain English, of the technical portions being implemented in one or preferably a variety of games to serve as example(s), could help this type of developer with the software engineering components of the methodology.
2. **Game designers who are more inclined towards software engineering than educational game design**. A more detailed explanation of the pedagogical portions being implemented in one or preferably a variety of games to serve as example(s), could help this type of developer with educational game design in general.
3. **Game designers who are more familiar with industrial game development than academic**. Converting/translating the utilisation methodology from its present form to one more akin to a professional game studio's workflow, could help this type of game designer implement the methodology as a whole.

With the limitations and potential future work of this study explored, the next section draws a formal conclusion for the overall study.

8.3 Conclusion

Educational games have proven to be an effective teaching medium due to their ability to both entertain and educate. However, one of the limitations of educational games as a teaching medium is the lack of real time educational adaptivity to better teach players, similar to the manner a teacher would in a classroom.

Educational adaptivity revolves around adapting game elements that effect a player's learning based upon that player's model, as the player is progressing through the educational game. Thus, challenges arise in identifying relevant game elements and their adaptable options, determining conditions within a game where adaptivity could aid players' learning, generating accurate player models, identifying a suitable data collection technique to monitor players during play sessions, and designing an algorithm to execute adaptations that is also capable of running in parallel with an educational game.

This study attempted to address these challenges through the identification of an adaptive framework for games in general, the creation of an adaptive algorithm and a methodology to utilise said algorithm. Although the user evaluation of the algorithm showed that players who received adaptations from the algorithm produced statistically significant higher learning gains than players who did not, a concrete conclusion of the algorithm's effectiveness could not be reached. This was due to the four limitations of evaluation being a preliminary one and thus exhibiting a high degree of noise and/or artefacts in its results, adaptive content being developed for a partial model of the RAES, the less effective instructional and one-shot approaches to teaching used, and the relatively small sample size. While the evaluation of the utilisation methodology also suffered from a relatively low sample size, three key weaknesses and their potential solutions were identified as possible directions for the improvement of said methodology.

Despite the limitations of both evaluations, the results obtained indicate that the design principles, implementation and utilisation methodology of the adaptive algorithm serve as a promising base to make educational games adaptive, to better teach their players. The effectiveness of this base could be further improved via additional research towards expanding the AES, choosing a different player modelling technique, running a more comprehensive user evaluation, and improving the utilisation methodology.

Chapter 9

Appendices

This chapter contains all the additional work and resources used in this study. This includes the specifications for the educational scenarios described in chapter 6, all documentation used for the user evaluation described in chapter 6 and all documentation used for the methodology evaluation described in chapter 7.

9.1 Educational scenario specifications

Below are the specifications for scenarios 2 to 6. Each includes an explanation of the educational content, a high level description of the events in the scenario, a figure showing the sequence of events and dialog options, all the possible dialog options, and a map of the in game area the scenario takes place in and the placement of NPCs. The labels in the figures refer to the labels of each dialog option detailed in their respective tables. Note that all NPCs were given names in the game. All references to NPCs in all lines of dialog were replaced with the respective name of the NPC in the final build of the game.

9.1.1 Scenario 2

Scenario two is aimed at teaching players to check sources of information for neutrality when attempting to determine the most reliable and/or accurate information. This scenario is based on a scenario described by Thompson (2001), in which a man is interested in trying hunting and wishes to know if it is dangerous. He proceeds to ask the opinion of four people, an avid hunter, someone who has hunted before, someone who is phobic of animals but has tried hunting and someone who has never been hunting. The first person claims that hunting is completely safe. The second says that there is risk but can be planned for and minimised. The third states that the animals and terrain are too much of a threat to be enjoyable. The fourth states that safety is highly dependent on the weather.

The man learns to discredit the first and third persons' opinion due to their bias for and against hunting respectively. He also discredits the fourth person's opinion due to having no practical experience with hunting. He concludes that the second person, having experienced hunting and giving the most neutral opinion, gave the most accurate information. He makes his decision based on this.

The player completes this scenario by navigating the world, finding and speaking to each of the four people. After gathering all four opinions, the player will then return to the helper NPC. The helper NPC will engage the player in a conversation where each of the opinions is evaluated against the disposition of the person who gave it. The scenario is concluded with the secondary validating the second person's opinion due to its neutrality.

Figure 9.1 details the sequence of events and the dialog options available to the player for scenario 2. Table 9.1 displays all dialog options and the dependencies of each option signifying the order in which dialog options become available. Figure 9.2 is a map showing the area scenario 2 occurs in and placements of all NPCs. Figure 9.3 displays the general layout of the playable area. Figure 9.4 shows the interior of one of the buildings. Figure 9.5 displays an example of the player engaging in dialog with one of the NPCs.

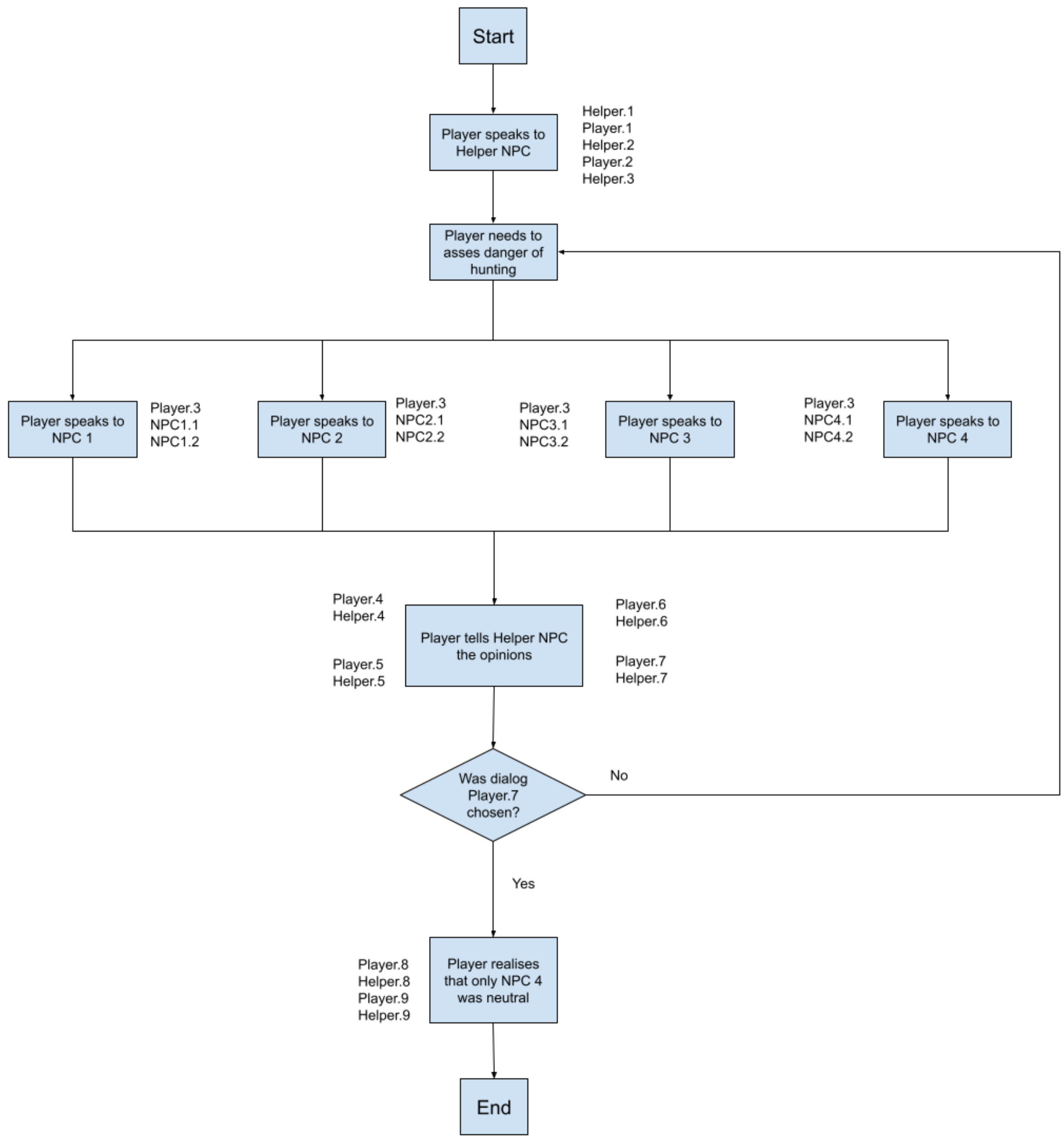

Figure 9.1: A flowchart showing the sequence of events and the dialog options for scenario 2

Table 9.1: The specification of all dialog options and their dependencies for scenario 2.

Label	**Dependencies**	**Full dialog**
Player.1	Helper.1	Yes, how can I help?
Player.2	Helper.2	I'm not actually sure myself...
Player.3	Helper.3	Could you tell me if hunting is dangerous?
Player.4	NPC1.2	Repeat what NPC1 said.

Player.5	NPC2.2	Repeat what NPC2 said.
Player.6	NPC3.2	Repeat what NPC3 said.
Player.7	NPC4.2	Repeat what NPC4 said.
Player.8	Helper.7	Why do you say so?
Player.9	Helper.8	I guess neutral sources give the best information.
Helper.1		Excuse me, could you help me out?
Helper.2	Player.1	I'm trying to figure out if hunting is dangerous.
Helper.3	Player.2	Ask the villagers around here for their opinions, one of them should know.
Helper.4	Player.4	He really loves hunting. His opinion might not be neutral. Maybe he isn't being completely honest about the dangers that could exist...
Helper.5	Player.5	He seems scared of forests and animals. He is probably biased against them. I don't think he is giving a fair opinion.
Helper.6	Player.6	But, he has never been hunting. I don't think he can give a fair or accurate opinion on something he has never experienced.
Helper.7	Player.7	That sounds like a fair and neutral opinion.
Helper.8	Player.8	He has been hunting before, he explained the possible dangers and how to stay safe. He is opinion is neutral and fair.
Helper.9	Player.9	Thanks for the help, I'm going to buy hunting equipment.
NPC1.1	Player.3	No hunting isn't dangerous at all.
NPC1.2	NPC1.1	I love hunting! I do it all the time!
NPC2.1	Player.3	Yes it is, there are so many ways you could get hurt.
NPC2.2	NPC2.1	I've only done it once and I was scared the whole time. Forests and animals are scary.
NPC3.1	Player.3	I think it is. Many people go all the time and they don't get hurt.
NPC3.2	NPC3.1	I've never gone hunting but I'd like to try.
NPC4.1	Player.3	It depends. The forest and animals can be dangerous but if you take the right equipment and stay alert, you should be safe.
NPC4.2	NPC4.1	I've gone hunting a few times and done those two things. It turned out fine.

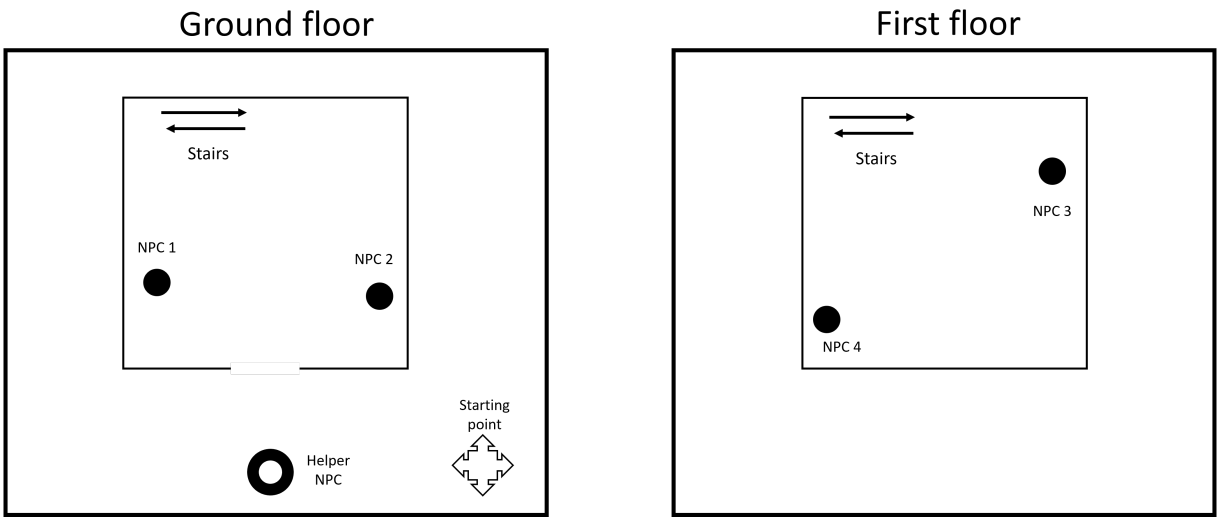

Figure 9.2: A map showing the area and placements of NPCs for scenario 2

Figure 9.3: A screen shot showing the general layout of the playable area.

Figure 9.4: A screen shot of the interior of one of the buildings.

Figure 9.5: A screen shot of the player engaging in dialog with one of the NPCs.

9.1.2 Scenario 3

Scenario three is aimed at teaching players to consider the possible vested interests of information sources when deciding which source is most reliable. This scenario is based on a scenario described by Butterworth and Thwaithes (2013), in which a man is trying to find the culprit who stole his cake. He has two suspects, both of whom deny the theft. He also has two witnesses whose testimony contradicts one another. The man then questions the motives and relationships between the witnesses and the suspects. The man realises that one of the witnesses is lying due to his vested interest for helping the culprit, as they are best friends. The man then confronts the culprit and retrieves the cake.

The player completes the scenario by questioning both witnesses to resolve the conflicting testimonies. The player needs to question the motives and interests of each of the witnesses to discover the vested interest of one of them. The lying witness then confesses which allows the player to make the culprit confess.

Figure 9.6 details the sequence of events and the dialog options available to the player for scenario 3. Table 9.2 displays all dialog options and the dependencies of each option signifying the order in which dialog options become available. Figure 9.7 is a map showing the area scenario 3 occurs in and placements of all NPCs. Figure 9.8 displays the general layout of the playable area. Figure 9.9 shows the interior of one of the buildings. Figure 9.10 displays an example of the player engaging in dialog with one of the NPCs.

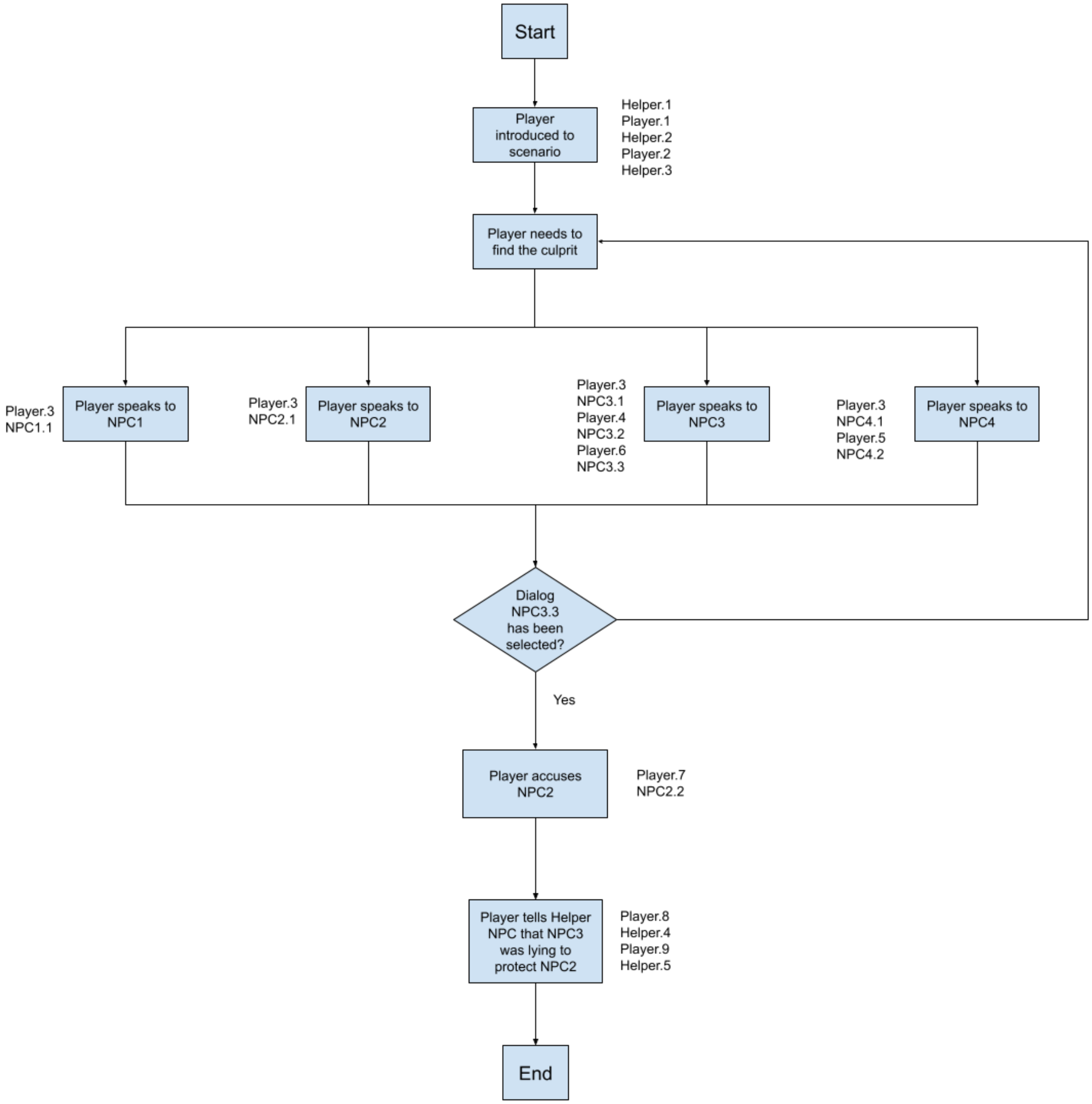

Figure 9.6: A flowchart showing the sequence of events and the dialog options for scenario 3

Table 9.2: The specification of all dialog options and their dependencies for scenario 3.

Label	**Dependencies**	**Full dialog**
Player.1	Helper.1	Alright, what's the problem?
Player.2	Helper.2	I'll find the culprit.
Player.3	Helper.3	Do you know who stole the cake?
Player.4	NPC4.1	Did you happen to see anyone near the kitchens last night?

Player.5	NPC3.2	NPC3 claims that NPC2 was with him all day yesterday. So could the culprit be NPC1?
Player.6	NPC4.2	I found out that NPC2 wasn't with you yesterday. Why did you lie?
Player.7	NPC3.3	NPC 3 told me the truth, I know it was you.
Player.8	NPC2.2	NPC2 stole the cake last night. NPC3 tried to lie about it to protect him.
Player.9	Helper.4	People with vested interests can't always be trusted I suppose.
Helper.1		Could you help me out with something?
Helper.2	Player.1	Someone in this village has stolen my cake! Could you find out who?
Helper.3	Player.2	Talk to the villagers and find out who the suspects and witnesses are.
Helper.4	Player.8	NPC2 and NPC3 are good friends... I guess NPC3 had a vested interested to lie and protect his friend.
Helper.5	Player.9	Yes! Thanks for your help. I'm going to give NPC2 and NPC3 a piece of my mind!
NPC1.1	Player.3	I have no idea, but it wasn't me.
NPC2.1	Player.3	I don't have a clue.
NPC2.2	Player.7	Ok, I confess, I stole the cake.
NPC3.1	Player.3	I really don't know.
NPC3.2	Player.4	I didn't but it couldn't have been NPC2 as he was with me in my house all of yesterday.
NPC3.3	Player.6	Well... he is my friend and I was just trying to protect him...he stole the cake when he got here last night... we already ate it.
NPC4.1	Player.3	I'm not sure. But I saw someone lurking near the kitchen last night.
NPC4.2	Player.5	What? NPC2 couldn't have been with NPC3 all of yesterday. He only arrived last night. I think NPC3 is lying, maybe go ask him about it.

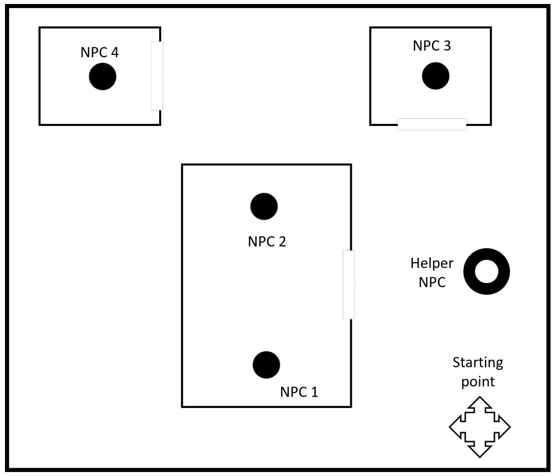

Figure 9.7: A map showing the area and placements of NPCs for scenario 3

Figure 9.8: A screen shot showing the general layout of the playable area.

Figure 9.9: A screen shot of the interior of one of the buildings.

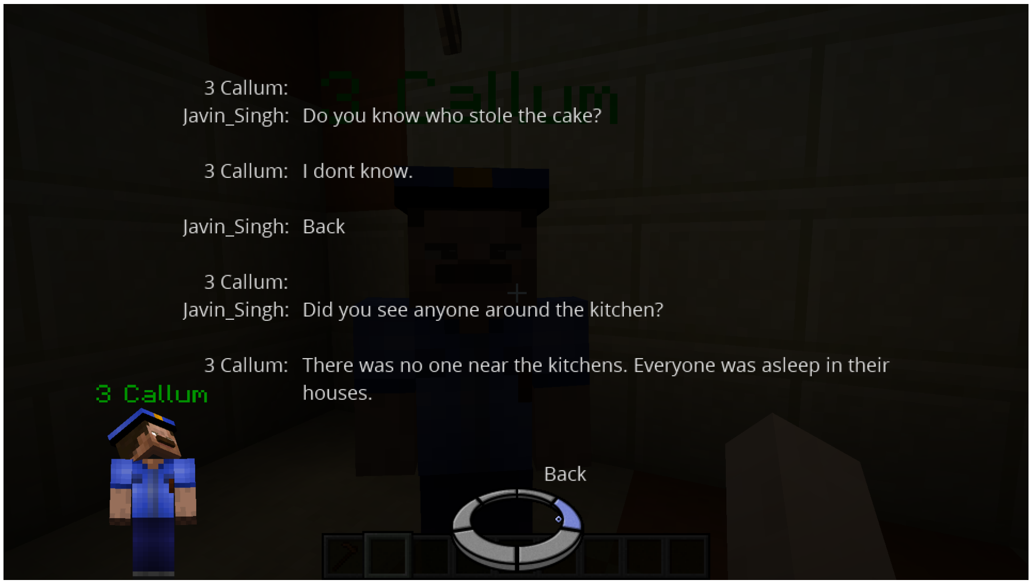

Figure 9.10: A screen shot of the player engaging in dialog with one of the NPCs.

9.1.3 Scenario 4

Scenario four is aimed at teaching players to consider the corroborative evidence and their reliability before coming to a conclusion. This scenario is based on a scenario described by Ennis and Millman (1989), in which a man is trying to determine the better soccer player between two candidates. He approaches 3 people. The first person claims the superiority of one of the candidates on account of his size and athleticism. The second rebuffs this claim, citing support for the opposing candidate due to his dexterity and ability to perform tricks. These two argue back and forth regarding the validity of each of their candidate's traits. However, the third person chooses to compare the game statistics of the two candidates. He cites that one candidate has scored more goals, won more games and won more competitions. He chose the superior soccer player based on factual corroborative evidence. Recognising this, the man opts to believe the third person.

The player completes the scenario by questioning all three of the people for their opinions and analysis. The player then needs to realise that the third person has the strongest corroborative evidence due to its factual and statistical nature. The scenario is concluded with the player relaying this information to the NPC.

Figure 9.11 details the sequence of events and the dialog options available to the player for scenario 4. Table 9.3 displays all dialog options and the dependencies of each option signifying the order in which dialog options become available. Figure 9.12 is a map showing the area scenario 4 occurs in and placements of all NPCs. Figure 9.13 displays the general layout of the playable area. Figure 9.14 shows the interior of one of the buildings.

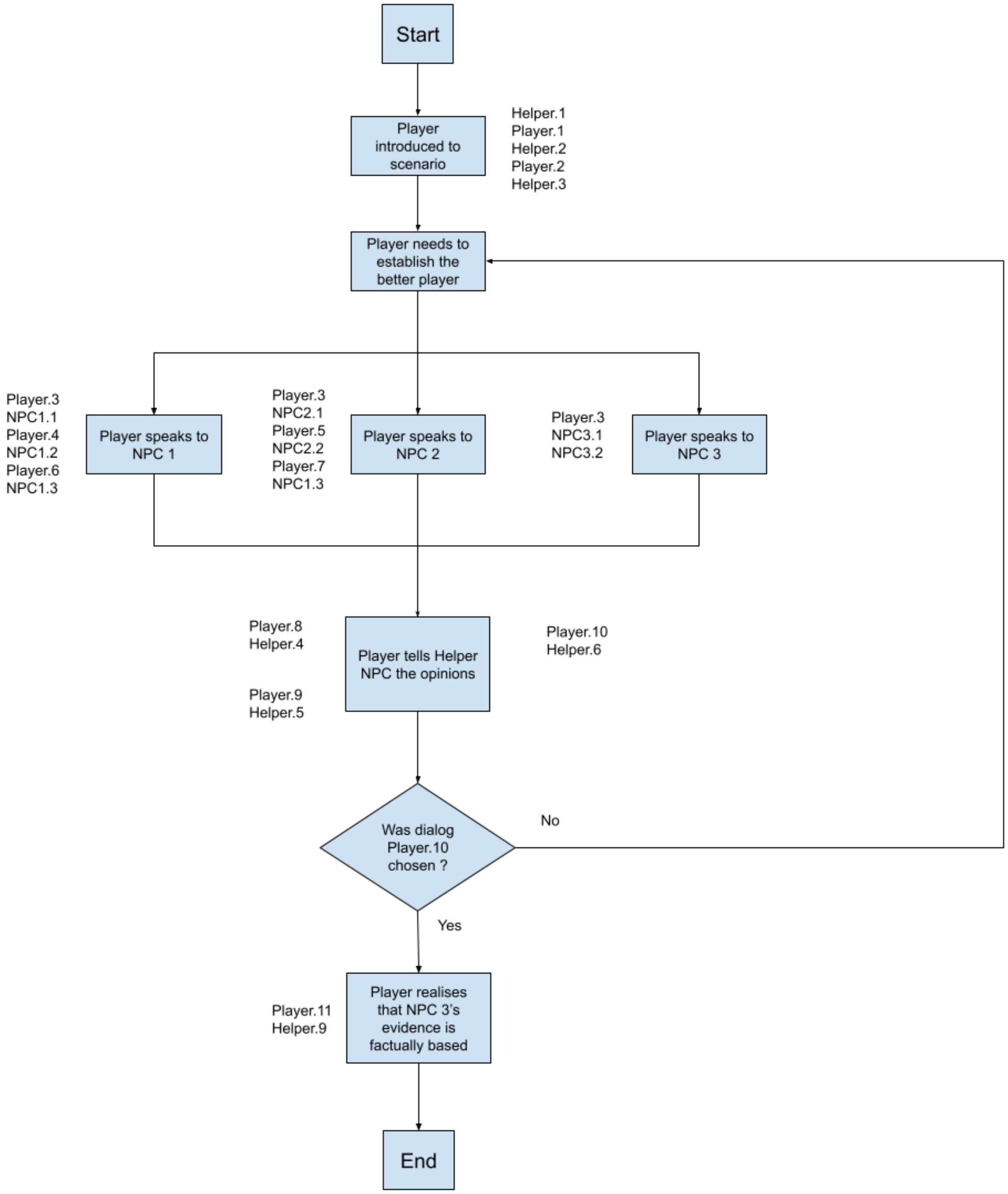

Figure 9.11: A flowchart showing the sequence of events and the dialog options for scenario 4

Table 9.3: The specification of all dialog options and their dependencies for scenario 4.

Label	**Dependencies**	**Full dialog**
Player.1	Helper.1	What is it?
Player.2	Helper.2	I don't know those players.
Player.3	Helper.3	Who is the better football player, Zane or Krieg?
Player.4	NPC1.1	NPC2 says Zane is faster.

Player.5	NPC2.1	NPC1 says Krieg is stronger.
Player.6	NPC1.2	NPC2 says that Zane's kicks are more accurate than Krieg's.
Player.7	NPC2.2	NPC1 says that Krieg is too big to be tackled by Zane.
Player.8	NPC1.3	Explain NPC1's opinion
Player.9	NPC2.3	Explain NPC2's opinion.
Player.10	NPC3.2	Explain NPC3's finding.
Player.11	Helper.6	Yes, I needed to check the facts instead of the opinions to find out.
Helper.1		Excuse me, I'm having a bit of trouble figuring something out.
Helper.2	Player.1	I'm a huge football fan and I'm trying to figure out if Zane or Krieg is the better player.
Helper.3	Player.2	Maybe you could ask the villagers around here for their opinions.
Helper.4	Player.8	Well yes but Zane seems to have some advantages over Krieg, so I have some doubt.
Helper.5	Player.9	Krieg seems to be better than Zane in certain areas. I'm still not sure that these opinions are providing a clear answer
Helper.6	Player.10	Well I can't argue with the facts. Although they both have their advantages, Zane has accomplished more than Krieg. Zane is the better player.
Helper.7	Player.11	Well, now I have my answer. Thanks for your help.
Helper.8	Player.8	He has been hunting before, he explained the possible dangers and how to stay safe. He is opinion is neutral and fair.
Helper.9	Player.9	Thanks for the help, I'm going to buy hunting equipment.
NPC1.1	Player.3	Krieg, he is a lot stronger and can kick the ball much harder.
NPC1.2	Player.4	Well speed is important but Zane can't tackle the ball from Krieg. Krieg is too big for that.
NPC1.3	Player.6	Well Zane is more accurate, but I still think Krieg is the better player.
NPC2.1	Player.3	Zane, he is faster and can outrun Krieg any day.
NPC2.2	Player.5	Ok, Krieg can kick the ball harder, but Zane's kicks are more accurate at scoring goals.
NPC2.3	Player.7	Yes, Zane can't Tackle Kreig but I think overall, Zane is still better.
NPC3.1	Player.3	I'm not sure, but I can check the facts on my computer.
NPC3.2	NPC3.1	Zane has scored more goals, won more matches and won more championships. Even though Krieg looks bigger and stronger, the facts don't lie, Zane has accomplished more and is better all around.

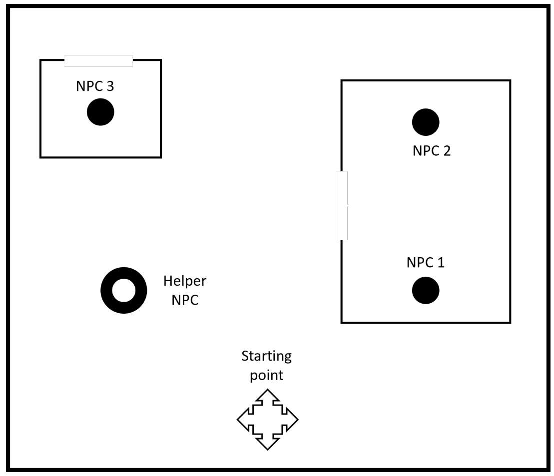

Figure 9.12: A map showing the area and placements of NPCs for scenario 4

Figure 9.13: A screen shot showing the general layout of the playable area.

Figure 9.14: A screen shot of the interior of one of the buildings.

9.1.4 Scenario 5

Scenario five is aimed at teaching players to check sources of information for relevance before trusting them. This scenario is based on a scenario described by Ennis and Millman (1989), in which a man is trying to decide between two restaurants to go to for dinner. He elicits the opinion of four others. The first person reveals his preference of restaurant due to his brother being the head chef of that restaurant. The second person states his preference based on that restaurant serving his favourite dish whereas the other does not. The third person makes his choice based on the comfortability of the chairs in his preferred restaurant. The fourth states that while differences in food preference always play a part, he experienced distinctly better service at his preferred restaurant. The man then realises that the first three people were biased in ways that he was not. Whereas, the fourth person gave him the most neutral and honest information, resulting in the man using this testimony to make his decision.

The player completes the scenario by questioning all four people, of their opinion and justifications. The player must then realise the bias of the first three people and the neutrality and relevance of the fourth. The player then must inform the NPC of the neutral opinion to complete the scenario.

Figure 9.15 details the sequence of events and the dialog options available to the player for scenario 5. Table 9.4 displays all dialog options and the dependencies of each option signifying the order in which dialog options become available. Figure 9.16 is a map showing the area scenario 5 occurs in and placements of all NPCs. Figure 9.17 displays the general layout of the playable area. Figure 9.18 shows the interior of one of the buildings. Figure 9.19 displays an example of the player engaging in dialog with one of the NPCs.

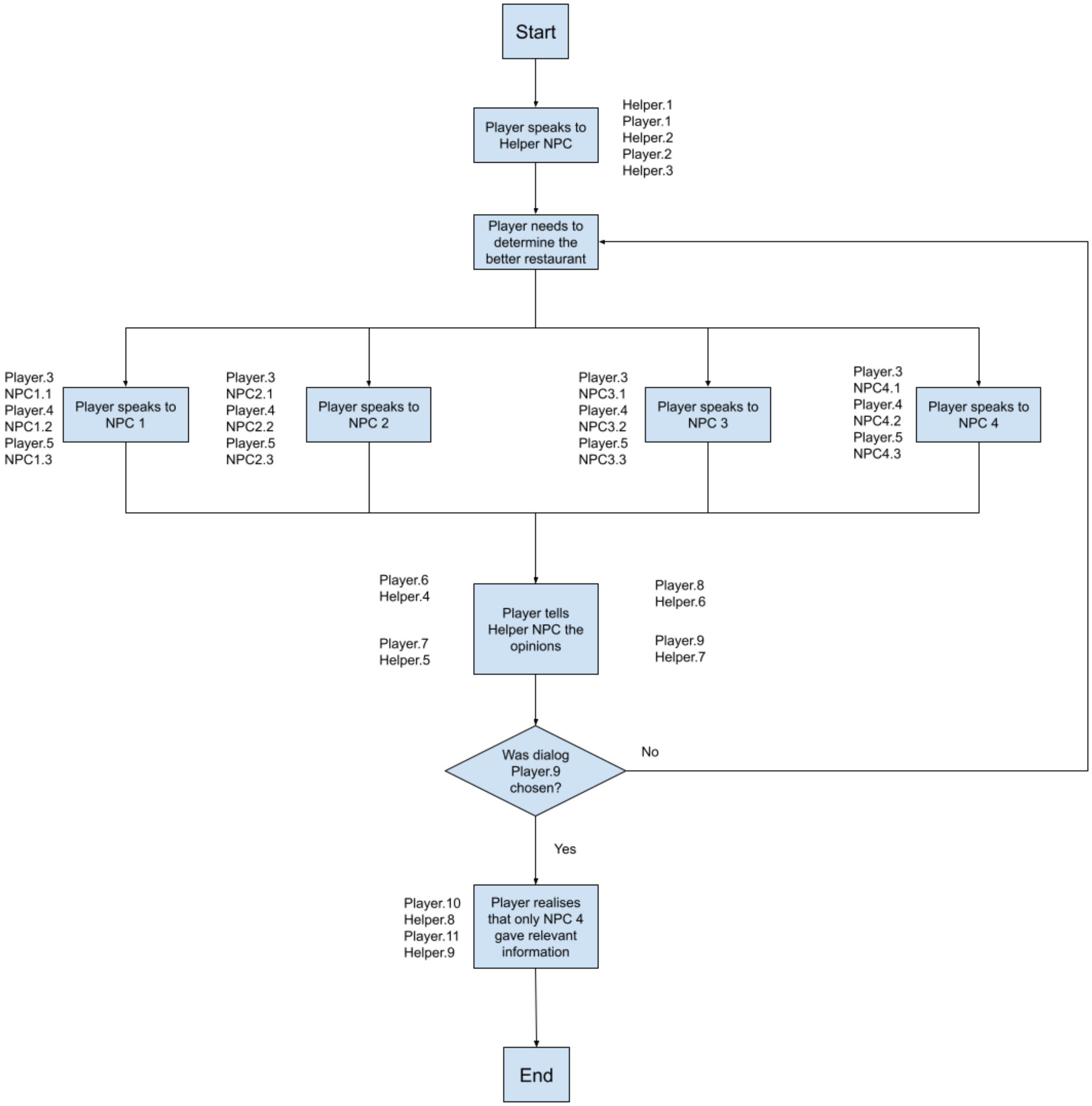

Figure 9.15: A flowchart showing the sequence of events and the dialog options for scenario 5

Table 9.4: The specification of all dialog options and their dependencies for scenario 5.

Label	**Dependencies**	**Full dialog**
Player.1	Helper.1	Alright, what do you need?
Player.2	Helper.2	Hmmm, I've never been to either of them.
Player.3	Helper.3	Is Mick's or Ralphs the better Restaurant?
Player.4	Helper.3	Why do you say so?

Player.5	Helper.3	Do you have any other reasons?
Player.6	NPC1.3	Repeat what NPC1 said.
Player.7	NPC2.3	Repeat what NPC2 said.
Player.8	NPC3.3	Repeat what NPC3 said.
Player.9	NPC4.3	Repeat what NPC4 said.
Player.10	Helper.7	What do you mean?
Player.11	Helper.8	I guess it's best to use relevant information to make decisions.
Helper.1		Excuse me, could you help me with a problem?
Helper.2	Player.1	I'm trying to decide between two restaurants for dinner. Should I go to Mick's or Ralph's?
Helper.3	Player.2	Ask the villagers around here for their opinions, they might be able to help.
Helper.4	Player.6	Ok, but it seems as though he is bias in favour of his brother.
Helper.5	Player.7	Good for him, but I don't really want pizza so I'm still not sure.
Helper.6	Player.8	I'm not as bothered by tables and chairs as him.
Helper.7	Player.9	That sounds like a neutral and relevant opinion.
Helper.8	Player.10	The customer service and cleanliness of a restaurant are relevant and important points compared to the others.
Helper.9	Player.11	Thanks for the help, I'm off to Micks.
NPC1.1	Player.3	I'd pick Mick's over Ralphs any day.
NPC1.2	NPC1.1	My brother is the chef at Micks. I love his cooking.
NPC1.3	NPC1.2	No, I just like supporting my brother.
NPC2.1	Player.3	Micks is better.
NPC2.2	NPC2.1	The pizza at Mick's is better than Ralph's.
NPC2.3	NPC2.2	No, I just tend to eat a lot of pizza.
NPC3.1	Player.3	I prefer Ralph's.
NPC3.2	NPC3.1	The chairs and tables are more comfortable there.
NPC3.3	NPC3.2	Well I care more about my comfort. The food from both places is roughly equal.
NPC4.1	Player.3	I'd say Micks's is the better place.
NPC4.2	NPC4.1	The customer service at Mick's is far better.
NPC4.3	NPC4.2	Mick's is much cleaner too.

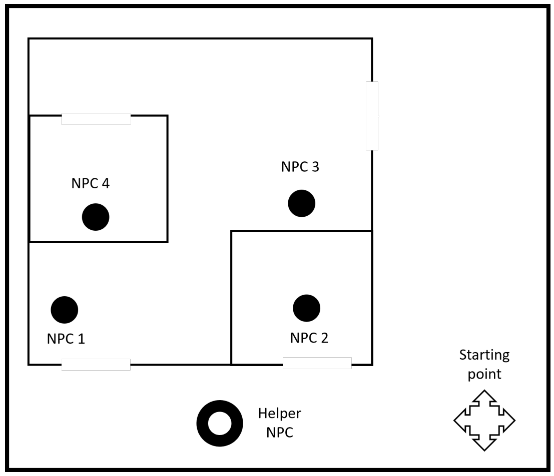

Figure 9.16: A map showing the area and placements of NPCs for scenario 5

Figure 9.17: A screen shot showing the general layout of the playable area.

Figure 9.18: A screen shot of the interior of one of the buildings.

Figure 9.19: A screen shot of the player engaging in dialog with one of the NPCs.

9.1.5 Scenario 6

Scenario six is aimed at teaching players to evaluate the strength of corroborative evidence before using it. This scenario is based on a scenario described by Ennis and Millman (1989), in which a man is trying to determine the culprit who destroyed his notes in his office. He questions four witnesses. The first witness claims to have heard a rumor that the housekeeper was the culprit. The second witness also blames the housekeeper as he claims to have seen a person in the office building from the outside. When pressed, he admits that the time of night affected his vision. The third witness claims to have been in the building but one floor below. He claims to have heard a commotion and yelling from the office in question and suspects a colleague based on the sound of the voice he heard. The fourth witness claims to have been next to the office and confirms the noise and the yells heard by the third witness. He explains that he investigated and found the housekeeper attempting to catch a stray cat that had entered the office. The cat had shredded the notes amidst the chaos.

The man realises that the first witness provided hearsay information with no evidence. The second witness, although somewhat present during the ordeal, did not provide any concrete evidence that the person he saw was the housekeeper. The third witness provides more direct testimony but also employs guesswork with no further evidence to name his culprit. Realising that the fourth witness's eyewitness account proves the strongest corroborative evidence due to physically witnessing and being present during the ordeal, the man believes the fourth witness.

The player completes the scenario by questioning all four people, of their accounts and suspicion. By doing this, the player is walked through the differing strengths of each testimony and must come to the conclusion that the eyewitness testimony proves the strongest and most direct. The player then relays this testimony to the NPC.

Figure 9.20 details the sequence of events and the dialog options available to the player for scenario 6. Table 9.5 displays all dialog options and the dependencies of each option signifying the order in which dialog options become available. Figure 9.21 is a map showing the area scenario 6 occurs in and placements of all NPCs. Figure 9.22 displays the general layout of the playable area. Figure 9.23 shows the interior of one of the buildings. Figure 9.24 displays an example of the player engaging in dialog with one of the NPCs.

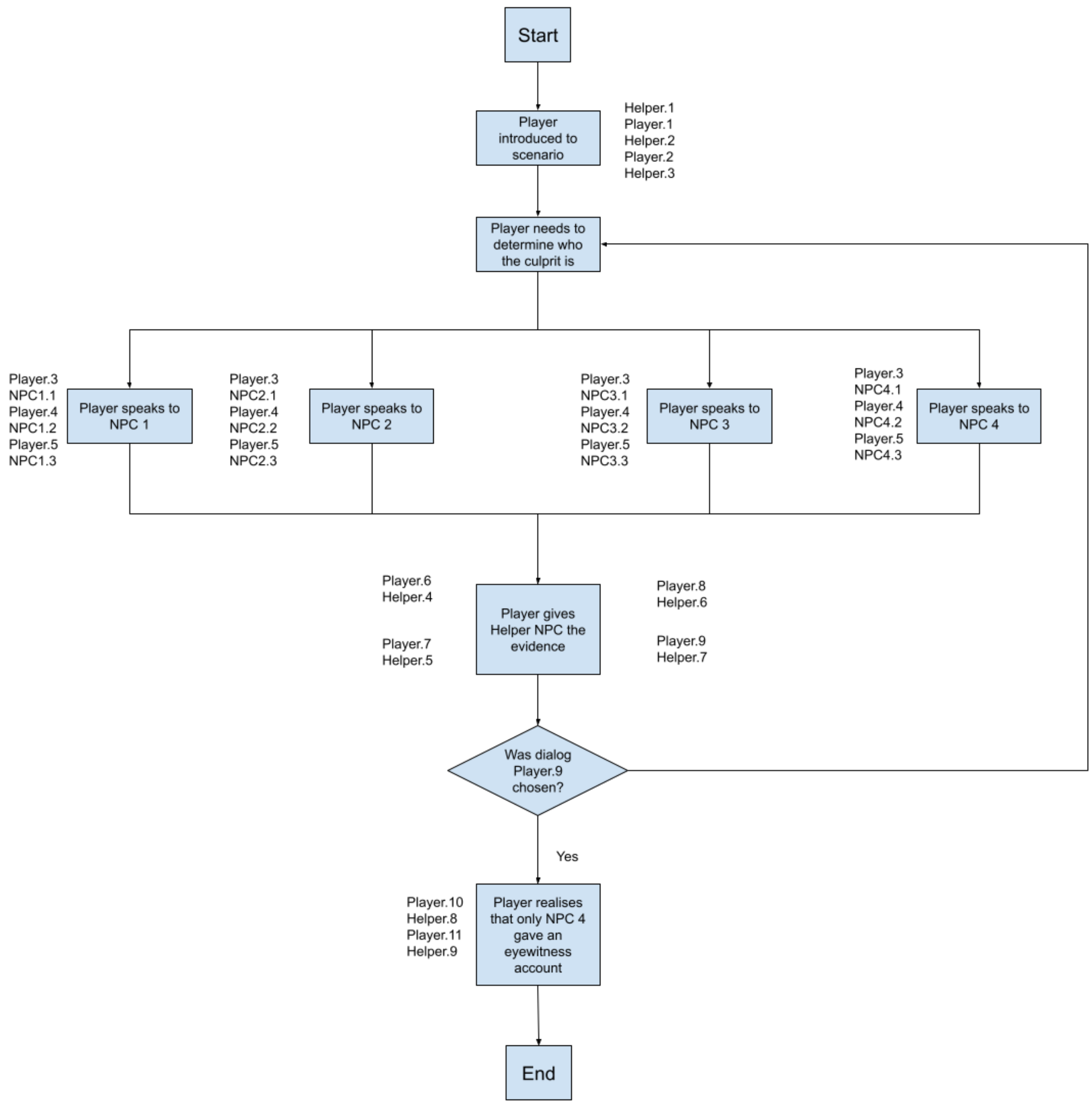

Figure 9.20: A flowchart showing the sequence of events and the dialog options for scenario 6

Table 9.5: The specification of all dialog options and their dependencies for scenario 6.

Label	Dependencies	Full dialog
Player.1	Helper.1	Sure, what is it?
Player.2	Helper.2	Ok, I can do that.
Player.3	Helper.3	Who wrecked NPC1's notes?
Player.4	Helper.3	Do you have any evidence?

Player.5	Helper.3	Are you sure that evidence is Strong?
Player.6	NPC1.3	Repeat what NPC1 said.
Player.7	NPC2.3	Repeat what NPC2 said.
Player.8	NPC3.3	Repeat what NPC3 said.
Player.9	NPC4.3	Repeat what NPC4 said.
Player.10	Helper.7	Yes it does.
Player.11	Helper.8	Eyewitness testimony should be sought after if possible.
Helper.1		Excuse me, could you help me with something?
Helper.2	Player.1	I left my notes in the office last night on the second floor. When i came back this morning my notes were destroyed. I need you to help me find out who did it.
Helper.3	Player.2	Thanks, people in the office if the saw or know anything.
Helper.4	Player.6	That's just a rumor, I can't accuse someone based on a rumor.
Helper.5	Player.7	Alright but he didn't actually see the house keeper did he?
Helper.6	Player.8	That just means someone was in my office but we don't know who.
Helper.7	Player.9	Well that settles it.
Helper.8	Player.10	Since NPC4 was an eyewitness, his evidence is stronger than the rest.
Helper.9	Player.11	Thanks for the help, I can't believe my bad luck!.
NPC1.1	Player.3	It was the housekeeper.
NPC1.2	NPC1.1	Someone else told me about it this morning.
NPC1.3	NPC1.2	I'm not sure, it might be a rumor.
NPC2.1	Player.3	I'd guess the house keeper.
NPC2.2	NPC2.1	I saw someone in the office last night from outside the building.
NPC2.3	NPC2.2	Well since it was dark I couldn't see that well. But there was definitely someone there.
NPC3.1	Player.3	It might have been NPC1, NPC2 or NPC4.
NPC3.2	NPC3.1	I was in the building last night and heard a lot of noise coming from that office.
NPC3.3	NPC3.2	I didn't actually see who it was but the other three and I were the only ones in the office late last night.
NPC4.1	Player.3	It was a cat.
NPC4.2	NPC4.1	I heard noise coming from the office and went to investigate. A stray had and climbed through the window and was messing everything up.
NPC4.3	NPC4.2	I saw it with my own eyes. And I called the housekeeper to catch the cat.

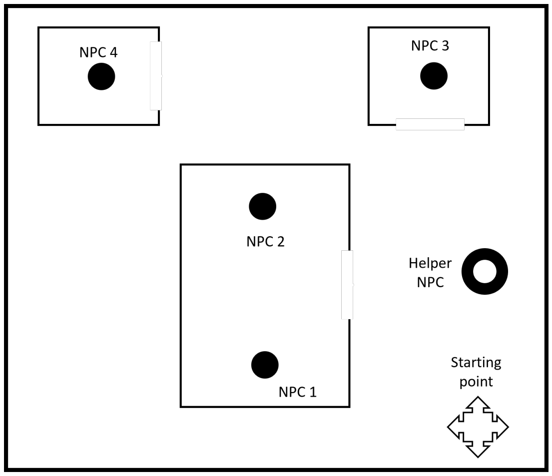

Figure 9.21: A map showing the area and placements of NPCs for scenario 6

Figure 9.22: A screen shot showing the general layout of the playable area.

Figure 9.23: A screen shot of the interior of one of the buildings.

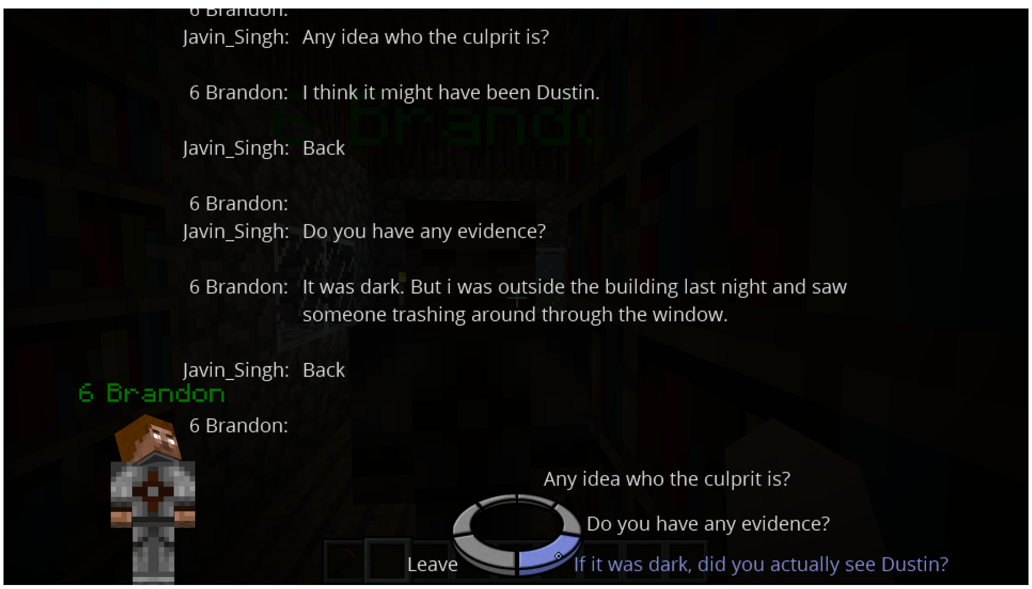

Figure 9.24: A screen shot of the player engaging in dialog with one of the NPCs.

9.2 User evaluation documents

Below are the information sheets and consent forms for both parents and participants, along with the pre-test and post-test used for the user evaluation.

INFORMATION SHEET FOR PARENTS

*Ethical Clearance Reference Number: **21806***

YOU WILL BE GIVEN A COPY OF THIS INFORMATION SHEET

Title of project

An algorithm for adapting educational video games to improve learning effectiveness.

Invitation Paragraph

I would like to invite your child to participate in this research project which forms part of my PhD programme. Before you decide whether you want your child to take part, it is important for you to understand why the research is being done and what your participation will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask me if there is anything that is not clear or if you would like more information.

What is the purpose of the project?

I have developed an adaptive artificial intelligence program that I believe will help students learn better when playing educational games. To test this, I have modified the commercial game Minecraft (developed by Mojang studios) to include educational scenarios. The educational content being taught is the analysis and credibility of evidence section of the Cambridge Thinking Skills Assessment syllabus. I would like your child to play my modified version of Minecraft to test the effectiveness of my artificial intelligence program.

Why have I been invited to take part?

You are being invited to participate in this project as you have a child between 13 and 16 years of age. The game and its content is aimed at that age range.

What will happen if I take part?

If your child chooses to take part in the project, he/she will be randomly assigned to either a control group, who will have the artificial intelligence deactivated, or an experimental group, who will have the artificial intelligence activated. This is because I require some participants to play my game with the artificial intelligence deactivated for me to measure the difference in its effects. Regardless of the group, your child will be asked to take a short multiple-choice test, play the Minecraft scenarios I made, and take another short multiple-choice test. Each multiple-choice test will take a maximum of 15 minutes and will ask your child to figure out the truth from made up scenarios given. Before starting, I will explain to your child the task he/she is to carry out and give him some background information about the game as well as encouragement. I will also make clear to your child that he/she may stop the evaluation at any time without having to give a reason. Each evaluation should take about one and a half hours.

Participation will take place in:
Public places - Parks/ Outdoor cafes with seating

Please note, if your child chooses to participate, at least one parent/guardian must present during the evaluation.

Do I have to take part?

Participation is completely voluntary. Your child should only take part if he/she wants to. Once you have read the information sheet, please contact me if you have any questions that will help you make a decision about taking part. If your child decides to take part, I will ask you to sign a consent form and you will be given a copy of this consent form to keep.

What are the possible risks of taking part?

There are no risks associated with this research as it involves two short paper-based tests and the playing of a video game with a minimum age rating of 7 years and up.

What are the possible benefits of taking part?

Participants may gain knowledge of analysing the credibility of evidence. Additionally, participants may also experience enjoyment from playing a new and unseen version of Minecraft.

Data handling and confidentiality

All data will be processed under the terms of UK data protection law (including the UK General Data Protection Regulation (UK GDPR) and the Data Protection Act 2018).

No personal information will be recorded from your child except their age. This information will be deleted upon the completion of my PhD in October 2021. For anonymity purposes, the name of your child will not be recorded. Instead, the child will be referred to using an identifiable code e.g., Participant 1.

All information recorded will only be kept until the end of this project (October 2021) on the King's College online storage system. The data collected will only be shared between myself, my supervisors and my examiners. The data will not be shared with anyone else or third party within or outside the UK.

Data Protection Statement

If you would like more information about how your data will be processed under the terms of UK data protection laws, please visit the link below:

https://www.kcl.ac.uk/research/support/research-ethics/kings-college-london-statement-on-use-of-personal-data-in-research

What if I change my mind about taking part?

You and/or your child are free to withdraw at any point of the project, without having to give a reason. Withdrawing from the project will not affect your child in any way. Your child will be able to withdraw from the project up until **01/08/2021** after which withdrawal of their data will no longer be possible due to the data being committed to the final report.

What will happen to the results of the project?

The results of the project will be summarised in the thesis for my PhD programme.

Who should I contact for further information?

If you or your child have any questions or require more information about this project, please contact me using the following contact details:

Javin Singh

K1803541@kcl.ac.uk

What if I have further questions, or if something goes wrong?

If this project has harmed you or your child in any way or if you wish to make a complaint about the conduct of the project you can contact King's College London using the details below for further advice and information:

Dr. Simon Miles

Department of Informatics
King's College London
Bush House
London
WC2B 4BG
UK

simon.miles@kcl.ac.uk

Thank you for reading this information sheet and for considering taking part in this research.

CONSENT FORM FOR PARENTS IN RESEARCH PROJECTS

Please complete this form after you have read the Information Sheet and/or listened to an explanation about the research

KING'S College LONDON

Title of project: An algorithm for adapting educational video games to improve learning effectiveness.	
Ethical review reference number: *21806*	**Version number: 1.1 – 16/01/21**

	Tick or initial
1. I confirm that I have read and understood the information sheet dated **16/6/2021 V 1.1** for the above project. I have had the opportunity to consider the information and asked questions which have been answered to my satisfaction.	
2. I consent voluntarily for my child to be a participant in this project and understand that I can withdraw my child from the project at any time, without having to give a reason, up until **01/08/2021**.	
3. I consent to the processing of my child's personal information for the purposes explained to me in the Information Sheet. I understand that such information will be handled under the terms of UK data protection law, including the UK General Data Protection Regulation (UK GDPR) and the Data Protection Act 2018.	
4. I understand that my child's information may be subject to review by responsible individuals from the College for monitoring and audit purposes.	
5. I understand that confidentiality and anonymity will be maintained, and it will not be possible to identify me or my child in any research outputs	
6. I understand that the information my child and I have submitted will be published as a report	

Name of Parent **Date** **Signature**

Name of Researcher **Date** **Signature**

INFORMATION SHEET FOR PARTICIPANTS

*Ethical Clearance Reference Number: **21806***

YOU WILL BE GIVEN A COPY OF THIS INFORMATION SHEET

Title of project

An algorithm for adapting educational video games to improve learning effectiveness.

Invitation Paragraph

I would like to invite you to participate in this research project which forms part of my PhD programme. Before you decide whether you want to take part, it is important for you to understand why the research is being done and what you will need to do. Please take time to read the following information carefully and discuss it with others if you wish. Ask me if there is anything that is not clear or if you would like more information.

What is the purpose of the project?

I have developed an adaptive artificial intelligence program that I believe will help students learn better when playing educational games. To test this, I have modified the commercial game Minecraft (developed by Mojang studios) to be an educational game. I would like you to play my modified version of Minecraft to test the effectiveness of my artificial intelligence program.

Why have I been invited to take part?

You are being invited to participate in this project as are between 13 and 16 years of age. The game and its content is aimed at that age range.

What will happen if I take part?

If you choose to take part in the project, you will be put into one of two groups. One group will play with the artificial intelligence activated while the other group will play with it deactivated. This is done so that I can measure the effect of my artificial intelligence. Regardless of which group you are in, you will be asked to take a short multiple-choice test, play the Minecraft scenarios I made, and take another short multiple-choice test. Each multiple-choice test will take a maximum of 15 minutes and will ask you to figure out the truth from made up scenarios given. Before starting, I will explain what will be asked of you. I will also make it clear to you can stop at any time without having to give a reason. Each evaluation should take about one and a half hours.

This will take place in:
Public places - Parks/ Outdoor cafes with seating

Please note, if you choose to participate, you need at least one of your parent(s)/guardian(s) to be present during the evaluation.

Do I have to take part?

Participation is completely voluntary. You should only take part if you want to. Once you have read the information sheet, please contact me if you have any questions that will help you make a decision about taking part. If you decide to take part, I will ask you and one of your parents to sign a consent form and you will be given a copy of this consent form to keep.

What are the possible risks of taking part?

There are no risks associated with this research as it involves two short paper-based tests and the playing of a video game with a minimum age rating of 7 years and up.

What are the possible benefits of taking part?

You may enjoy playing a new and unseen version of Minecraft.

Data handling and confidentiality

All data will be processed under the terms of UK data protection law (including the UK General Data Protection Regulation (UK GDPR) and the Data Protection Act 2018).

No personal information will be recorded from you except your age. This information will be deleted upon the completion of my PhD in October 2021. For anonymity purposes, your name will not be recorded. Instead, you will be referred to using an identifiable code e.g., Participant 1.

All information recorded will only be kept until the end of this project (October 2021) on the King's College online storage system. The data collected will only be shared between myself, my supervisors and my examiners. The data will not be shared with anyone else or third party within or outside the UK.

Data Protection Statement

If you would like more information about how your data will be processed under the terms of UK data protection laws, please visit the link below:

https://www.kcl.ac.uk/research/support/research-ethics/kings-college-london-statement-on-use-of-personal-data-in-research

What if I change my mind about taking part?

You are free to withdraw at any point of the project, without having to give a reason. Withdrawing from the project will not affect you in any way. You will be able to withdraw from the project up until **01/08/2021** after which withdrawal will no longer be possible due to the data being committed to the final report.

What will happen to the results of the project?

The results of the project will be summarised in the thesis for my PhD programme.

Who should I contact for further information?

If you or your child have any questions or require more information about this project, please contact me using the following contact details:

Javin Singh

K1803541@kcl.ac.uk

What if I have further questions, or if something goes wrong?

If this project has harmed you in any way or if you wish to make a complaint about the conduct of the project you can contact King's College London using the details below for further advice and information:

Dr. Simon Miles

Department of Informatics
King's College London
Bush House
London
WC2B 4BG
UK

simon.miles@kcl.ac.uk

Thank you for reading this information sheet and for considering taking part in this research.

CONSENT FORM FOR PARTICIPANTS IN RESEARCH PROJECTS

Please complete this form after you have read the Information Sheet and/or listened to an explanation about the research

Title of project: An algorithm for adapting educational video games to improve learning effectiveness.	
Ethical review reference number: *21806*	**Version number: 1.1 – 16/06/21**

	Tick or initial
1. I confirm that I have read and understood the information sheet dated **16/6/2021 V 1.1** for the above project. I have had the opportunity to consider the information and asked questions which have been answered to my satisfaction.	
2. I consent voluntarily to be a participant in this project and understand that I can refuse to take part and can withdraw from the project at any time, without having to give a reason, up until **01/08/2021**.	
3. I consent to the processing of my personal information for the purposes explained to me in the Information Sheet. I understand that such information will be handled under the terms of UK data protection law, including the UK General Data Protection Regulation (UK GDPR) and the Data Protection Act 2018.	
4. I understand that my information may be subject to review by responsible individuals from the College for monitoring and audit purposes.	
5. I understand that confidentiality and anonymity will be maintained, and it will not be possible to identify me in any research outputs	
6. I understand that the information I have submitted will be published as a report	

Name of Participant **Date** **Signature**

Name of Researcher **Date** **Signature**

Cornell Critical Thinking Test Series

The Cornell Critical Thinking Test, Form X

By

Robert H. Ennis
Jason Millman

Pre-Test

Participant Number : _______

Participant age: _______

General Instructions:

This is a test to see how well you do a particular kind of thinking. It is called “evidence and credibility”. You might see that you already do some of this kind of thinking. There are two keywords you will encounter in the questions. This is what they mean

A. Might - It may be true or it may not be true. The truth is not certain.
B. Definite - The truth is completely certain.

DO NOT GUESS WILDLY. There is a scoring penalty for guessing wrong. This means that marks will be deducted for every wrong answer. If you think you have the answer, but are not sure, mark that answer. But if you have no idea, then **SKIP** the question.

There are three possible answers. This is what they mean:

A. **YES** - It must be true.
B. **NO** - It can't be true.
C. **MAYBE** - It may be true or it may not be true. You weren't told enough to be certain whether it is "YES" or "NO".

Remember, if you have no idea what the correct answer is, **SKIP** the question.

---

1. A cookie is missing from the jar. Mary claims she saw Fred in the kitchen.

   Would this be true?

   Fred might have taken the cookie.

YES	NO	MAYBE

2. John claims to have climbed a mountain. He tells you to ask his friends, who did not go with him, if he is telling the truth.

   Would this be true?

   John climbed the mountain.

YES	NO	MAYBE

3. Jack enters his home. He sees a broken vase on the floor and his dog sitting nearby.

   Would this be true?

   Jack's dog definitely broke the vase?

YES	NO	MAYBE

4. Harry claims to have gotten full marks on his spelling test. He asks you to quiz him as proof.

   Would this be true?

   Harry definitely got full marks on his spelling test.

YES	NO	MAYBE

5. Beth's pen is missing. She later sees David writing with the same type of pen she had.

   Would this be true?

   David stole Beth's pen.

YES	NO	MAYBE

6. Susie claims to be a very good ballet dancer. She shows you all her ballet outfits as proof.

   Would this be true?

   Susie is a very good ballet dancer.

YES	NO	MAYBE

7. Cole tells you that you might be ill. Cole happens to be a fully qualified medical doctor.

   Would this be true?

   Cole’s judgment can be trusted.

YES	NO	MAYBE

8. Jenny needs to study for a test. Jenny’s older sister suggests using her old notes to study instead of her own.

   Would this be true?

   Jenny should use her sister’s old notes.

YES	NO	MAYBE

9. You need to buy Tom a birthday gift. You know that he has two good friends, Paul and Peter. You only have time to talk to one of them.

   Would this be true?

   You should talk to Paul.

YES	NO	MAYBE

10. Dennis wants to know what the best sport is. He could ask Peter, who is a basketball player. Or, he could ask Tim, who is a football player.

    Would this be true?

    Dennis should ask both Peter and Tim.

YES	NO	MAYBE

11. You want to know if Jane or Jack is better at geography. You happen to also know their geography teacher.

    Would this be true?

    Asking their teacher will tell you who is better.

YES	NO	MAYBE

12. Mary and Betty had a debate. Several people watched the debate, including Betty's brother Willaim. You want to know who had the stronger opinion.

    Would this be true?

    Asking William will reveal who had the stronger opinion.

YES	NO	MAYBE

13. Kate wants to buy a cake but cannot decide which cake to buy. She decides to ask her baker for his opinion.

    Would this be true?

    Kate can trust the baker's opinion.

YES	NO	MAYBE

14. Barry is wondering if he should buy new shoes. He decides to ask Dave for his opinion. Dave tells Barry to buy new shoes. Dave is a shoe salesman.

    Would this be true?

    Barry should listen to Dave.

YES	NO	MAYBE

15. Jenny accuses Max of breaking her table. Max denies that he did it. Jenny asks Dennis if Max could be guilty.

    Would this be true?

    Jenny can take what Dennis says seriously.

YES	NO	MAYBE

16. Jack claims to be good at tennis. He shows you all his tennis trophies where he came in first place.

    Would this be true?

    Jack is a good tennis player.

YES	NO	MAYBE

17. Grace is unsure what to do for her school science project. She decides to ask her teacher if an experiment involving magnets is a good idea.

    Would this be true?

    Grace should now choose to do her school science project on magnets.

YES	NO	MAYBE

18. Beth is not feeling well. She decides to ask her classmate if she is sick. Her classmate says that she has a fever because she feels slightly warm.

    Would this be true?

    Beth definitely has a fever.

YES	NO	MAYBE

Cornell Critical Thinking Test Series

The Cornell Critical Thinking Test, Form X

By

Robert H. Ennis
Jason Millman

Post-Test

Participant Number : _______

Participant age: _______

General Instructions:

This is a test to see how well you do a particular kind of thinking. It is called “evidence and credibility”. You might see that you already do some of this kind of thinking. There are two keywords you will encounter in the questions. This is what they mean

A. Might - It may be true or it may not be true. The truth is not certain.
B. Definite - The truth is completely certain.

DO NOT GUESS WILDLY. There is a scoring penalty for guessing wrong. This means that marks will be deducted for every wrong answer. If you think you have the answer, but are not sure, mark that answer. But if you have no idea, then **SKIP** the question.

There are three possible answers. This is what they mean:

A. **YES** - It must be true.
B. **NO** - It can't be true.
C. **MAYBE** - It may be true or it may not be true. You weren't told enough to be certain whether it is "YES" or "NO".

Remember, if you have no idea what the correct answer is, **SKIP** the question.

---

1. John claims to be a professional football player. He shows you his expensive pair of football boots as proof.

   Would this be true?

   John is a professional football player.

YES	NO	MAYBE

2. Daisy has a bright pink car. One day you see a bright pink car parked in Jane's driveway.

   Would this be true?

   Daisy is definitely at Jane's house.

YES	NO	MAYBE

3. Randy claims to have run a complete marathon. He shows you a picture of himself running at the half-way point

   Would this be true?

   Randy ran the complete marathon.

YES	NO	MAYBE

4. Jack claims that he went to the cinema to watch a movie. Mary claims she saw Jack leave the house.

   Would this be true?

   Jack went to the cinema.

YES	NO	MAYBE

5. Wendy accuses Janet of cheating on a test. Wendy claims she saw Janet peaking at the textbook during the test.

   Would this be true?

   Janet might have cheated.

YES	NO	MAYBE

6. Gary's wallet is missing from his bedroom. The last person he saw coming out from his bedroom was Tom.

   Would this be true?

   Tom definitely stole Gary's wallet.

YES	NO	MAYBE

7. Mary is wondering if she should pack an umbrella to go to the park. Her friend Beth tells her that she should.

   Would this be true?

   Mary should pack an umbrella.

YES	NO	MAYBE

8. Jack is not sure if he did his history homework correctly. He decides to check his work against his history textbook.

   Would this be true?

   Jack will find out if he did his homework correctly.

YES	NO	MAYBE

9. William wants to buy the fastest car available, but he does not know which car is the fastest. He decides to ask his friend Gary for advice. Gary has never owned a car.

   Would this be true?

   Gary will be able to tell William which car is fastest.

YES	NO	MAYBE

10. Matt wants to try surfing but is worried that it may be dangerous. He asks Charlie if surfing is dangerous. Charlie loves surfing and goes whenever he can.

    Would this be true?

    Charlie's advice will be completely accurate.

YES	NO	MAYBE

11. Alex committed a crime. His best friend Peter claims that Alex could not have committed the crime.

    Would this be true?

    Peter's opinion should prove Alex's innocence.

YES	NO	MAYBE

12. You want to know if Sandra or Amy is the faster runner. You also know that they both raced during a sports event. You check the race results.

    Would this be true?

    The results will tell you who the faster runner is.

YES	NO	MAYBE

13. Martin was accused of breaking a vase. Robert was in the same room as Martin at the time. Robert claims that Martin did not break the vase.

    Would this be true?

    Robert is telling the truth.

YES	NO	MAYBE

14. Susie is unsure if she should join the drama club or the music club. She decides to ask the music teacher for her advice.

    Would this be true?

    Susie went to the right person for advice.

YES	NO	MAYBE

15. Jenny is trying to find her way into town but is lost. She decides to ask a policeman for directions.

    Would this be true?

    The policeman would give her the right directions.

YES	NO	MAYBE

16. James is unsure if he is ill. He knows that he has been feeling cold all day and has been sneezing.

    Would this be true?

    James is ill.

YES	NO	MAYBE

17. Tracy claims that she is very good at mathematics. She shows you all her mathematics test papers which all have a score of above 90%.

    Would this be true?

    Tracy’s mathematics test scores show that she is good at mathematics.

YES	NO	MAYBE

18. Tyler is unsure if going to New York is safe. He decides to ask his friend Matt. Matt says New York is safe as he has been to the city next to it.

    Would this be true?

    Tyler now knows that New York is safe.

YES	NO	MAYBE

9.3 Utilisation methodology evaluation documents

Below is the information sheet, consent form and questionnaire provided to participants taking past in the utilisation methodology evaluation.

INFORMATION SHEET FOR PARTICIPANTS

*Ethical Clearance Reference Number: **34840***

YOU WILL BE GIVEN A COPY OF THIS INFORMATION SHEET

Title of project

An algorithm for adapting educational video games to improve learning effectiveness.

Invitation Paragraph

I would like to invite you to participate in this research project which forms part of my PhD programme. Before you decide whether you want to take part, it is important for you to understand why the research is being done and what your participation will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask me if there is anything that is not clear or if you would like more information.

What is the purpose of the project?

I have developed an adaptive algorithm that I believe will help students learn better when playing educational games. I have already designed and evaluated the algorithm by running a user evaluation using a custom modified version of Minecraft. My PhD also includes a utilisation methodology intended to provide game designers with instructions for incorporating my algorithm with other educational games. I am now trying to evaluate the effectiveness of this utilisation methodology.

Why have I been invited to take part?

You are being invited to participate in this project as you have experience in educational game development.

What will happen if I take part?

If you choose to take part in the project, you will be given some documentation regarding the adaptation algorithm to read, and be given the choice between answering a questionnaire or taking a short semi structured interview with myself. Both the questionnaire and interview consists of 10 questions meant to be scored along a 5 point Likert scale along with a brief explanation for each answer. You may also withdraw from the evaluation at any time without having to give a reason.

Do I have to take part?

Participation is completely voluntary. You should only take part if you want to. Once you have read the information sheet, please contact me if you have any questions that will help you make a decision about taking part. If you decide to take part, I will ask you to sign a consent form and you will be given a copy of this consent form to keep.

What are the possible risks of taking part?

There are no risks associated with this research as it involves reading some documentation and answering some questions.

What are the possible benefits of taking part?

Participants may gain knowledge of a novel form of adaptivity that could improve the teaching ability of educational games.

Data handling and confidentiality

All data will be processed under the terms of UK data protection law (including the UK General Data Protection Regulation (UK GDPR) and the Data Protection Act 2018).

No personal information will be recorded from you. Only your responses during the questionnaire or interview will be noted. For anonymity purposes, your name of will not be recorded. Instead you will be referred to using an identifiable code e.g., Participant 1.

All information recorded will only be kept until the end of this project (July 2023) on the King's College online storage system. The data collected will only be shared between myself, my supervisors and my examiners. The data will not be shared with anyone else or third party within or outside the UK.

Data Protection Statement

If you would like more information about how your data will be processed under the terms of UK data protection laws, please visit the link below:

https://www.kcl.ac.uk/research/support/research-ethics/kings-college-london-statement-on-use-of-personal-data-in-research

What if I change my mind about taking part?

You are free to withdraw at any point of the project, without having to give a reason. Withdrawing from the project will not affect your child in any way. Your child will be able to withdraw from the project up until **30/04/2023** after which withdrawal of their data will no longer be possible due to the data being committed to the final report.

What will happen to the results of the project?

The results of the project will be summarised in the thesis for my PhD programme.

Who should I contact for further information?

If you or your child have any questions or require more information about this project, please contact me using the following contact details:

Javin Singh

K1803541@kcl.ac.uk

What if I have further questions, or if something goes wrong?

If this project has harmed you or your child in any way or if you wish to make a complaint about the conduct of the project you can contact King's College London using the details below for further advice and information:

Dr. Simon Miles

Department of Informatics
King's College London
Bush House
London
WC2B 4BG
UK

simon.miles@kcl.ac.uk

Thank you for reading this information sheet and for considering taking part in this research.

CONSENT FORM FOR PARENTS IN RESEARCH PROJECTS

Please complete this form after you have read the Information Sheet and/or listened to an explanation about the research

Title of project: An algorithm for adapting educational video games to improve learning effectiveness.	
Ethical review reference number: *34840*	**Version number: 1.0 – 5/12/22**

	Tick or initial
1. I confirm that I have read and understood the information sheet dated **5/12/2022 V 1.0** for the above project. I have had the opportunity to consider the information and asked questions which have been answered to my satisfaction.	
2. I consent voluntarily to be a participant in this project and understand that I can withdraw from the project at any time, without having to give a reason, up until **30/04/2023**.	
3. I understand that there will be no no processing of my personal information whatsoever. I understand that the response data I give will be handled under the terms of UK data protection law, including the UK General Data Protection Regulation (UK GDPR) and the Data Protection Act 2018.	
4. I understand that my information may be subject to review by responsible individuals from the College for monitoring and audit purposes.	
5. I understand that confidentiality and anonymity will be maintained, and it will not be possible to identify me in any research outputs	
6. I understand that the information I have submitted will be published as a report	

Name of Participant	**Date**	**Signature**
Name of Researcher	**Date**	**Signature**

Questionnaire

An algorithm for adapting educational video games to improve learning effectiveness.

This questionnaire consists of 10 questions. Each question is meant to be scored using a 5-point Likert scale with 5 being completely agree and 1 being completely disagree. There is also a short section after each question to explain your decision.

(1) The purpose of the methodology given is concise and clear. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____
____
____

For questions 2 and 3, I aimed to allow game developers to quickly discern whether my approach would be applicable to the educational game they are developing through specifying requirements clear enough to filter out cases where my approach wouldn't help.

(2) The compatibility requirements allow for compatible games to be easily identified. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____
____
____

(3) The compatibility requirements make the functionalities needed to turn incompatible games into compatible ones clear. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____
____
____

For questions 4 and 5, I aimed to allow game developers to quickly understand the purpose, structure and how to develop their own educational scenarios.

(4) The concept, purpose and stages of educational scenarios are clear. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____
____
____

(5) Game designers trying to create educational scenarios akin to the ones in the document will be able to do so using the educational scenario creation methodology. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____
____
____

For questions 6 and 7, I aimed to allow game developers to quickly discern how to place the algorithm's two triggers in a game of their choice.

(6) The player interaction trigger methodology paints a clear picture of how to place the algorithm's player interaction trigger in other games. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____
____
____

(7) The non player interaction trigger methodology paints a clear picture of how to place the algorithm's non player interaction trigger in other games. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____

For questions 8 and 9, I aimed to allow game developers to understand how to distinguish between preference and performance elements and be able to develop adaptive content for the elements they have chosen.

(8) The distinction between and the method to classify preference and performance elements is clear. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

____

(9) The preference element methodology and performance element methodology, along with their provided examples, illustrate how to develop adaptive content for any preference and/or performance element in general. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

(10) Assuming that the design principles and logic behind the algorithm sent to you (the document I show the game designers) is sound, the methodology provided could help other game designers in utilising the algorithm to make other educational games adaptive. Why?

☐ 5 - Completely agree

☐ 4 – Somewhat agree

☐ 3 – Neutral

☐ 2 – Somewhat disagree

☐ 1 – Completely disagree

Explanation for answer

Bibliography

4AGames (2019). “Metro exodus”. In: *Deep silver developer diaries* 1. DOI: `https://www.4a-games.com.mt`.

Aarseth, E. (2005). “From Hunt the Wumpus to EverQuest: Introduction to Quest Theory”. In: *Entertainment Computing - ICEC 2005: 4th International Conference*, pp. 496–506.

— (2012). “A narrative theory of games”. In: *International Conference on the Foundations of Digital Games*, pp. 129–133.

Aarseth, E., S.M. Smedstad, and L. Sunnanå (2003). “A multi-dimensional typology of games in Level Up Games”. In: *Digital Games Research Conference 2003* 2, pp. 48–53.

Adhatrao, K. et al. (2013). “Predicting Students’ Performance using ID3 and C4.5 classification algorithms”. In: *International Journal of Data Mining Knowledge Management Process* 3.5, pp. 39–52. DOI: `10.5121/ijdkp.2013.3504`.

Aggarwal, S. (2013). “Applications of Games with a Purpose”. In: *Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational Serious Games (CGAMES)*, pp. 54–59. DOI: `10.1109/CGames.2013.6632605`.

Ahlbrand, A. (2017). “Learning theories and law: Behaviorism, cognitivism, constructivism.” In: *Learning theories and law* 84.6, pp. 204–212.

Ahmad, A., O. Basir, and K. Hassanein (2004). “Adaptive user interfaces for Intelligent e-Learning: Issues and Trends”. In: *Fourth International Conference on Electronic Business*, pp. 925–934.

Ahn, L. and L. Dabbish (2008). “Designing games with a purpose”. In: *Communications of the Association for Computing Machinery* 51.8, pp. 57–67.

Almala, A. (2006). “Applying the principles of constructivism to a quality E-learning environment”. In: *Distance Learning* 3.1, pp. 33–40.

Almohammadi, K. et al. (2017). “A Survey of Artificial Intelligence Techniques Employed for Adaptive Educational Systems within E-Learning Platforms”. In: *Journal of Artificial Intelligence and Soft Computing Research* 7.47–64.

Amineh, R. and H. Asl (2015). “Review of Constructivism and Social Constructivism”. In: *Journal of Social Sciences, Literature and Languages* 1.1, pp. 9–16.

Amory, A. et al. (1999). “The use of computer games as an educational tool: identification of appropriate game types and game elements”. In: *British Journal of Educational Technology* 30, pp. 311–321. DOI: `10.1111/1467-8535.00121`.

Amsel, A. (1989). “Behaviorism, Neobehaviorism, and Cognitivism in Learning Theory”. In: *Historical and Contemporary Perspectives* 1.

Arachchilage, N. and S. Love (2013). “A game design framework for avoiding phishing attacks”. In: *Computers in Human Behavior* 29.3, pp. 706–714. DOI: `https://doi.org/10.1016/j.chb.2012.12.018`.

Arsenault, D. (2014). “Narratology”. In: *Routledge companion to Video Game Studies*, pp. 475–483. DOI: `http://hdl.handle.net/1866/18996`.

Ashmore, C. and M. Nitsche (2007). “The quest in a generated world”. In: *Digital Games Research Association*, pp. 503–509.

Asteriadis, S., K. Karpouzis, and S. Kollias (2008). “A neuro-fuzzy approach to user attention recognition”. In: *International Conference on Artificial Neural Networks*, pp. 927–936. DOI: `https://doi.org/10.1007/978-3-540-87536-9_95`.

Azough, S. (2010). “Adaptive E-learning using Genetic Algorithms”. In: *International Journal of Computer Science and Network Security* 10.7, pp. 237–277.

Badr, H. and E. Mohammed (2019). “A Personalized Pedagogical Objectives Based on a Genetic Algorithm in an Adaptive Learning System”. In: *Procedia Computer Science* 151, pp. 1152–1157.

Banning, M. (2006). “Measures that can be used to instill critical thinking in nurse prescribers.” In: *Nurse Education in Practice* 6.2, pp. 98–105.

Baradwaj, K. and S. Pal (2012). “Mining Educational Data to Analyze Students’ Performance”. In: *International Journal of Advanced Computer Science and Applications* 2.6, pp. 63–69.

Bartlett, D. and P. Cox (2002). “Measuring Change in Students Critical Thinking Ability: Implications for Health Care Education”. In: *Journal of Allied Health* 31.2, pp. 64–69.

Bateman, C. and R. Boon (2005). *21st century game design*. Vol. 1. Charles River Media.

Bateman, C., R. Lowenhaupt, and L. Nacke (2011). “Player Typology in Theory and Practice”. In: *Digital Games Research Association* 6.

Beetham, H. and R. Sharpe (2013). “Rethinking pedagogy for a digital age”. In: *Designing for 21st century learning*, pp. 25–36.

Behar-Horenstein, L. and N. Lian (2011). “Teaching Critical Thinking Skills In Higher Education: A Review Of The Literature.” In: *Journal of College Teaching and Learning.* 8.2. DOI: `8.10.19030/tlc.v8i2.3554`.

Bioware (2012). “Mass effect 3”. In: *Mass effect 3 design diary* 1. DOI: `https://www.bioware.com`.

Blickle, G. (1996). “Personality traits, learning strategies, and performance”. In: *Eur. J. Pers* 10, pp. 337–352.

Boghossian, P. (2002). “The Socratic Method (Or, Having a Right to Get Stoned)”. In: *Teaching Philosophy* 25.4.

— (2005). “How the Socratic Method Works”. In: *Informal Logic (Teaching Supplement)y* 23.2.

— (2006). “Behaviorism, Constructivism, and Socratic Pedagogy”. In: *Educational Philosophy and Theory* 38.6, pp. 713–722. DOI: `10.1111/j.1469-5812.2006.00226.x`.

Bontchev, B. and O. Georgieva (2018). “Playing style recognition through an adaptive video game”. In: *Computers in Human Behavior* 82, pp. 136–147. DOI: `https://doi.org/10.1016/j.chb.2017.12.040`.

Boyadjian-Samawi, Z. (2006). “The effect of concept mapping on critical thinking skills and dispositions of junior and senior baccalaureate nursing students”. In: *Widener University School of Nursing press.*

Braghirolli, L. et al. (2016). “Benefits of educational games as an introductory activity in industrial engineering education”. In: *Computers in Human Behavior* 58, pp. 315–324. DOI: `https://doi.org/10.1016/j.chb.2015.12.063`.

Brand, J. (2015). “The Narrative and Ludic Nexus in Computer Games: Diverse Worlds II”. In: *DiGRA 2015: Changing Views: Worlds in Play, 2015 International Conference.*

Brathwaite, B. and I. Schreiber (2008). *Challenges for Game Designers.* Vol. 1. Charles River Media.

Brown, J., Collins, and P. Duguid (1989). “Situated cognition and the culture of learning”. In: *Educational researcher* 18.1, pp. 32–42.

Bunt, A., C. Conati, and K. Muldner (2004). “Scaffolding Self-Explanation to Improve Learning in Exploratory Learning Environments”. In: *Intelligent Tutoring Systems, Lecture Notes in Computer Science* 3220. DOI: https://doi.org/10.1007/978-3-540-30139-4_62.

Butterworth, J. and G. Thwaites (2013). *Thinking skills: Critical thinking and problem solving.* Vol. 2. Cambridge University Press. ISBN: 1107606306.

Cakaj, S. (2010). *Modeling Simulation and Optimization: Focus on Applications.* Vol. 1. BoD – Books on Demand.

Campbell, D. and J. Stanley (1963). *Experimental and quasi-experimental designs for research.* Chicago: RandMcNally Company.

Capcom (2005). “Resident evil 4”. In: *Resident Evil 4: The Official Strategy Guide. Future Press* 1. DOI: https://archive.org/details/Resident_Evil_4_Wii_Prima_Official_eGuide.

Castellar, E. et al. (2014). “Improving arithmetic skills through gameplay: Assessment of the effectiveness of an educational game in terms of cognitive and affective learning outcomes”. In: *Information Sciences* 264, pp. 19–31. DOI: https://doi.org/10.1016/j.ins.2013.09.030.

Chanel, G. et al. (2011). “Emotion Assessment From Physiological Signals for Adaptation of Game Difficulty”. In: *Institute of Electrical and Electronics Engineers* 41.6, p. 99. DOI: 10.1109/TSMCA.2011.2116000.

Changchun, Liu. et al. (2009). “Dynamic Difficulty Adjustment in Computer Games Through Real-Time Anxiety-Based Affective Feedback”. In: *International Journal of Human-Computer Interaction* 25.6, pp. 506–529. DOI: 10.1080/10447310902963944.

Charles, D. et al. (2005). “Player-centred game design: Player modelling and adaptive digital games”. In: *DiGRA Conference: Changing Views—Worlds in Play*, pp. 285–298.

Chen, J. (2006). “Flow in Games”. In: *University of Southern California press.* DOI: http://www.jenovachen.com/flowingames/thesis.htm.

Chen and C. Lee (1999). “New methods for students’ evaluating using fuzzy sets”. In: *Fuzzy Sets and Systems* 104, pp. 209–218. DOI: https://doi.org/10.1016/S0165-0114(97)00208-X.

Cho, H. et al. (2007). “Social networks, communication styles, and learning performance in a CSCL community”. In: *Computers Education* 49.2, pp. 309–329. DOI: https://doi.org/10.1016/j.compedu.2005.07.003.

Cin, A. and F. Baba (2008). “Assessment of English proficiency by fuzzy logic approach”. In: *International Educational Technology Conference*, pp. 355–359.

Ciresan, D. et al. (2011). “Flexible, High Performance Convolutional Neural Networks for Image Classification”. In: *Twenty-Second International Joint Conference on Artificial Intelligence* 2, pp. 1237–1242.

Clark, D. et al. (2016). “Self-explanation and digital games: Adaptively increasing abstraction. Computers Education”. In: *Computers Education* 103, pp. 28–43. DOI: https://doi.org/10.1016/j.compedu.2016.09.010.

Clements, D. (1997). “Constructing Constructivism.” In: *Teaching children mathematics* 4.4, pp. 198–200.

Conati, C. (2002). “Probabilistic Assessment of User’s Emotions”. In: *Merging Cognition and Affect in Human Computer Interaction* 16, pp. 555–575.

Conati, C. and X. Zhao (2004). "Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game". In: *Association for Computing Machinery* 9, pp. 6–13. DOI: `https://doi.org/10.1145/964442.964446`.

Connolly, T. et al. (2012). "A systematic literature review of empirical evidence on computer games and serious games". In: *Computers and Education* 59.2, pp. 661–686. DOI: `https://doi.org/10.1145/964442.964446`.

Cook, J. (1992). "Negotiating the curriculum: Programming for learning". In: *Negotiating the curriculum: educating for the 21st century*, pp. 15–31.

Cortazzi, M. and L. Jin (1997). "Communication for learning across cultures". In: *Overseas Students in Higher Education* 1.

Cowley, B. and D. Charles (2016). "Adaptive Artificial Intelligence in Games: Issues, Requirements, and a Solution through Behavelets-based General Player Modelling". In: *Cornell University printing press*. DOI: `https://doi.org/10.48550/arXiv.1607.05028`.

Cowley, B., D. Charles, et al. (2008). "Toward an understanding of flow in video games". In: *Computer Entertainment* 6.2, pp. 1–27.

Csikszentmihalyi, M. (1975). *Beyond boredom and anxiety*. Vol. 1. San Francisco: Jossey-Bass Publishers.

— (1990). *Flow: The psychology of optimal experience*. Vol. 1. Harper and Row.

DeLope, R., J. Arcos, et al. (2017). "Design methodology for educational games based on graphical notations: Designing Urano". In: *Entertainment Computing* 18, pp. 1–14.

DeLope, R., N. Medina-Medina, et al. (2017). "Designing educational games: Key elements and methodological approach". In: *International Conference on Virtual Worlds and Games for Serious Applications* 9, pp. 63–70.

Derry, S. (1996). "Cognitive schema theory in the constructivist debate." In: *Educational Psychologist* 3, pp. 163–174.

Dietrich, A. et al. (2019). "Microadaptivity within complex learning situations-a personalized approach based on competence structures and problem spaces." In: *15th International Conference on Computers in Education*.

Drigas, A., K. Argyri, and J. Vrettaros (2009). "Decade Review, Artificial Intelligence Techniques in Student Modeling". In: *Knowledge, Learning, Development and Technology for All, in Best Practices for the Knowledge Society* 49, pp. 552–564.

Dunham, M. H. (2006). *Data Mining: Introductory and advance topic*. Vol. 1. Pearson Education India.

Ennis, R. (1989). "Critical thinking and subject specificity: Clarification and needed research". In: *Educational Researcher* 18.3.

Ennis, R. and J. Millman (1989). *Cornell Critical Thinking Test Level X*. Vol. 4. Cornell printing press. ISBN: 0-89455-286-4.

Ennis, R., J. Millman, and T. Tomko (2005). *Cornell Critical Thinking Tests Level X and level Z Manual*. Vol. 5. Cornell printing press.

Epic, E. (2017). *Epic games official site*. URL: `https://www.epicgames.com/fortnite/en-US/home`.

Ertmer, P. and T. Newby (1993). "Behaviorism, cognitivism and constructivism; Comparing critical features from an instructional design perspective". In: *Performance Improvement Quarterly* 6, pp. 50–66.

Espejo, R. and R. Harnden (1989). "The Viable system model: interpretations and applications of Stafford Beer's vsm". In: *Phrontis Limited SYNCHO Limited*.

EvolutionaryGames (2012). "Galactic Arms Race: an experiment in evolving video game content". In: *ACM SIGEVOlution* 4.4. DOI: https://doi.org/10.1145/1810136.1810137.

Fadem, B. (2008). *High-Yield Behavioral Science. High-Yield Series.* Vol. 2. Lippincott Williams Wilkins. ISBN: 978-0-7817-8258-6.

Fairclough, C. (2006). "Story games and the OPIATE system". In: *University of Dublin, Trinity College press* 13, pp. 33–34.

Fenza, G., F. Orciuoli, and D. Sampson (2017). "Building Adaptive Tutoring Model Using Artificial Neural Networks and Reinforcement Learning". In: *International Conference on Advanced Learning Technologies* 17, pp. 460–462. DOI: 10.1109/ICALT.2017.124.

Ferreira, H. et al. (2016). "An Automatic and Dynamic Student Modeling Approach for Adaptive and Intelligent Educational Systems Using Ontologies and Bayesian Networks". In: *International Conference on Tools with Artificial Intelligence* 28, pp. 738–745. DOI: doi:10.1109/ICTAI.2016.0116.

Firaxis (2016). "Civilisation VI". In: *2K Studious*. DOI: https://firaxis.com.

Freiberg, J. (1999). *Beyond Behaviorism: Changing the classroom management paradigm.* Vol. 1. Allyn Bacon.

Frias-Martinez, E. et al. (2004). "Recent soft computing approaches to user modeling in adaptive hypermedia". In: *Adaptive Hypermedia and Adaptive Web-Based Systems* 3137, pp. 104–114.

Garca, P. et al. (2005). "Using Bayesian networks to detect students' learning styles in a web-based education system". In: *ASAI Rosario*, pp. 115–126.

Garousi, V. and D. Pfahl (2016). "When to automate software testing? A decision-support approach based on process simulation." In: *Journal of Software: Evolution and Process* 28, pp. 272–285. DOI: doi : 10.1002/smr.1758.

Gertner, A. and K. VanLehno (2000). "Andes: A coached problem solving environment for physics". In: *Intelligent Tutoring Systems* 1839, pp. 133–142.

Gonzalez, C., J. Burguillo, and M. Llamas (2006). "A Qualitative Comparison of Techniques for Student Modeling in Intelligent Tutoring Systems". In: *Institute of Electrical and Electronics Engineers*, pp. 13–18. DOI: 10.1109/FIE.2006.322537.

Gutierrez, F. and J. Atkinson (2011). "Adaptive feedback selection for intelligent tutoring systems". In: *Expert Systems with Applications* 38.5, pp. 6146–6152. DOI: 10.1016/j.eswa.2010.11.058.

Habgood, M. and S. Ainsworth (2011). "Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games." In: *Journal of the Learning Sciences* 20.2, pp. 169–206. DOI: 10.1080/10508406.2010.508029.

Hamdaoui, N., M. Idrissi, and S. Bennani (2021). "Learner Modeling in Educational Games Based on Fuzzy Logic and Gameplay Data". In: *International Journal of Game-Based Learning* 11.2, pp. 38–60. DOI: http://doi.org/10.4018/IJGBL.2021040103.

Hare, D., E. Howard, and M. Pope (2005). "Enhancing technology use in student teaching: A case study". In: *Journal of Technology and Teacher Educations* 13.4, p. 573.

Harteveld, C. et al. (2010). "Balancing play, meaning and reality: The design philosophy of Levee Patroller". In: *Simulation and Gaming* 3.43, pp. 316–340.

Hastings, E., R. Guha, and K. Stanley (2009). "Evolving content in the galactic arms race video game". In: *Institute of Electrical and Electronics Engineers Symposium on Computational Intelligence and Games*, pp. 241–248. DOI: 10.1109/CIG.2009.5286468.

Hesterberg, L. (2005). "Evaluating of a problem-based learning practice course: Do self-efficacy, critical thinking, and assessment skills improve?" In: *University of Kentucky press*.

Hoover, W. (1996). "The practice implications of constructivism". In: *Science Education* 9.3, pp. 1–2.

Huang, X. et al. (2008). "Prediction of student actions using weighted Markov models, IT in Medicine and Education". In: *Institute of Electrical and Electronics Engineers International Symposium on Digital Object Identifier*. DOI: 10.1109/ITME.2008.4743842.

Hullett, K. and M. Mateas (2009). "Scenario generation for emergency rescue training games". In: *Proceedings of the 4th International Conference on Foundations of Digital Games* 4, pp. 99–106.

Huo, Y. (2019). "A Pedagogy-Based Framework for Optimizing Learning Efficiency across Multiple Disciplines in Educational Games". In: *International Journal of Information and Education Technology* 9.10, pp. 704–709.

Ibrahim, R. and A. Jaafar (2009). "Educational Games (EG) Design Framework: Combination of Game Design, Pedagogy and Content Modeling". In: *International Conference on Electrical Engineering and Informatics* 293.

Ibrahim, R. and D. Rusli (2007). "Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression". In: *21st Annual SAS Malaysia Forum*.

Idris, N., N. Yusof, and P. Saad (2009). "Adaptive course sequencing for personalization of learning path using neural network". In: *International Journal of Advanced Soft Computing and Its Applications* 1.1, pp. 49–61.

Jagust, T., I. Boticki, and H. So (2018). "Examining competitive, collaborative and adaptive gamifcation in young learners' math learning". In: *Computers Education* 125, pp. 444–457.

Jarvinen, A. (2007). "Introducing Applied Ludology: Hands-on Methods for Game Studies". In: *Situated Play, Proceedings of DiGRA 2007 Conference*. DOI: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.190.536&rep=rep1&type=pdf.

Jenkins, H. (2004). "Game Design as Narrative Architecture". In: *The Massachusetts Institute of Technology Press*, pp. 118–130.

Jennett, C. et al. (2008). "Measuring and defining the experience of immersion in games". In: *International Journal of Human Computer Studies* 64, pp. 641–661. DOI: 10.1016/j.ijhcs.2008.04.004.

Juul, J. (2005). "Half-real: video games between real rules and fictional worlds". In: *The Massachusetts Institute of Technology Press* 1.

Kapa, E. (2001). "A Metacognitive Support during the Process of Problem Solving in a Computerized Environment". In: *Educational Studies in Mathematics* 47.3, pp. 317–336. DOI: https://doi.org/10.1023/A:1015124013119.

Kavi, A. et al. (2003). "Student Modelling Based on Fuzzy Inference Mechanisms". In: *Proceedings of the IEEE Region 8 EUROCON 2003*.

Kickmeier-Rust, M. et al. (2006). "The ELEKTRA project: Towards a new learning experience". In: *Interdisciplinary aspects on digital media education*, pp. 19–48. DOI: 10.13140/2.1.2272.8646.

Kojima, H. (2015). "Metal Gear Solid 5". In: *Kojima Productions Developer Diaries* 1. DOI: https://www.kojimaproductions.jp/en.

Konami (2017). "Pro Evolution Soccer 2018". In: *Developer diaries* 1. DOI: https://www.konami.com/efootball/en/.

Koster, R. (2004). "A theory of fun for game design". In: *AZ: Paraglyph Press* 1.

Lazić, L. and D. Velašević (2004). “Applying simulation and design of experiments to the embedded software testing process”. In: *Software Testing, Verification and Reliability* 14, pp. 257–282. DOI: https://doi.org/10.1002/stvr.299.

Lepper, M. and D. Cordova (1992). “A desire to be taught: Instructional consequences of intrinsic motivation”. In: *Motivation and Emotion* 16, pp. 187–208.

Lepper, M. and J. Henderlong (2000). *Educational Psychology, Intrinsic and Extrinsic Motivation*. Vol. 2. Contemporary Educational Psychology. Chap. 10, pp. 257–307.

Li, G. (2012). “Culturally contested Pedagogy: Battles of literacy and schooling between mainstream teachers and Asian immigrant parents.” In: *State University of New York Press*.

Lin, C. et al. (2013). “Data Mining for Providing a Personalized Learning Path in Creativity: An Application of Decision Trees”. In: *Computers and Education* 68, pp. 199–210. DOI: https://doi.org/10.1016/j.compedu.2013.05.009ers.

Lipman, M. (1988). “Critical thinking: What can it be?” In: *Educational Leadership*, pp. 34–38.

Liu, Z., J. Moon, and B. Kim (2020). “Integrating adaptivity in educational games: a combined bibliometric analysis and meta-analysis review”. In: *Education Technology Research Development* 68, pp. 450–457. DOI: https://doi.org/10.1007/s11423-020-09791-4.

Lopes, R. and R. Bidarra (2011). “Adaptivity Challenges in Games and Simulations: A Survey”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 3.2, pp. 85–99.

Magdalena, L. (2015). *Fuzzy Rule-Based Systems*. Vol. 1. Springer Handbooks.

Marsh, T. and B. Costello (2012). “Experience in Serious Games: Between Positive and Serious Experience”. In: *Serious Games Development and Applications: Third International Conference* 3, pp. 256–267. DOI: 7528.255-267.10.1007/978-3-642-33687-4_22.

Mayra, F. and L. Ermi (2011). “Fundamental components of the gameplay experience”. In: *DIGAREC Series* 6, pp. 88–115.

Michael, D. and R. Chen (2006). “Serious Games: Games That Educate, Train, and Inform”. In: *Muska Lipman/Premier-Trade*.

Min, W. (2020). “DeepStealth: Game-Based Learning Stealth Assessment With Deep Neural Networks”. In: *IEEE Transactions on Learning Technologies* 13.2, pp. 312–325. DOI: 10.1109/TLT.2019.2922356.

Mojang (2011). “Minecraft”. In: *Developer design documents*. DOI: https://www.minecraft.net/en-us.

Monolith (2005). “F.E.A.R”. In: *F.E.A.R developer diaries* 1. DOI: https://lith.com.

Montgomery, D. (1997). *Design and Analysis of Experiment*. Vol. 4. Wiley.

Moreno, F. et al. (2005). “Using Bayesian Networks in the Global Adaptive E-learning Process”. In: *International Journal of Emerging technologies in Learning*, pp. 38–52.

Nacke, L. and C. Lindley (2010). “Affective Ludology, Flow and Immersion in a First- Person Shooter: Measurement of Player Experience”. In: *The Journal of the Canadian Game Studies Association* 3.5. DOI: https://doi.org/10.48550/arXiv.1004.0248.

Nations, United (2015). *Shockingly low proficiency rates in reading and mathematics signal a global learning crisis*. URL: https://unstats.un.org/sdgs/report/2019/goal-04/.

Newzoo, N. (2021). *Most Popular PC Games – Global*. URL: https://newzoo.com/insights/rankings/top-20-pc-games.

Nicalis (2014). “The Binding of Isaac Rebirth”. In: *Nicalis game studious*. DOI: https://www.nicalis.com.

Niehaus, J. and M. O. Riedl (2009). "Scenario adaptation: An approach to customizing computer-based training games and simulations". In: *Proceedings of the AIED 2009 workshop on intelligent educational games* 3, pp. 89–98.

Nintendo (2008). "Mario Kart Wii". In: *Design documents* 1. DOI: `Available:http://www.mariokart.com/wii/launch/`.

Nitsche, M. et al. (2006). "Designing procedural game spaces: A case study". In: *FuturePlay*.

Ortega, R. (2020). "Pedagogical practice preferences among generational groups of learners: Towards effective twenty-first century higher education". In: *Journal of University Teaching Learning Practice* 17.5. DOI: `https://doi.org/10.53761/1.17.5.6`.

Papadimitriou, S., K. Chrysafiadi, and M. Virvou (2019). "FuzzEG: Fuzzy logic for adaptive scenarios in an educational adventure game". In: *Multimedia Tools Applied* 78. DOI: `https://doi.org/10.1007/s11042-019-07955-w`.

Pashler, H. et al. (2008). "Learning Styles: Concepts and Evidence". In: *Psychological Science in the Public Interest* 9.3, pp. 105–119. DOI: `doi:10.1111/j.1539-6053.2009.01038.x`.

Pedersen, C., J. Togelius, and G. Yannakakis (2010). "Modeling Player Experience for Content Creation". In: *IEEE Transactions on Computational Intelligence and AI in Games* 2.1, pp. 54–67.

Peirce, N., O. Conlan, and V. Wade (2008). "Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences". In: *Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning*, pp. 28–35. DOI: `10.1109/DIGITEL.2008.30`.

Pelanek, R. (2016). "Applications of the Elo rating system in adaptive educational systems". In: *Computers Education* 98, pp. 169–179. DOI: `10.1016/j.compedu.2016.03.017`.

Poerksen, B. (2004). "We Can Never Know What Goes on in Somebody Else's Head': Ernst von Glasersfeld on truth and viability, language and knowledge, and the premises of constructivist education". In: *Cybernetics and Systems* 35.4, pp. 379–398.

Popkewitz, T. (1999). *Critical Theories in Education: Changing Terrains of Knowledge and Politics*. Vol. 2. Taylor Amp.

QuanticDream (2010). "Heavy Rain". In: *Sony Computer Entertainment* 1. DOI: `Available:http://www.heavyrainps3.com/`.

Rania, H. (2009). "Serious Games to Teach Ethics". In: *Columbia State faculty bibliography* 1056. DOI: `https://csuepress.columbusstate.edu/bibliography_faculty/1056`.

Rapeepisarn, K. et al. (2008). "The relationship between game genres, learning techniques and learning styles in educational computer games". In: *3rd International Conference on Technologies for E-Learning and Digital Entertainment* 3, pp. 497–508. DOI: `http://dx.doi.org/10.1007/978-3-540-69736-7_53`.

Reeves, B. and J. Read (2009). "Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete". In: *Harvard Business School Press*.

RemedyEntertainment, R. (2001). "Max Payne". In: *Gathering of Developers* 1. DOI: `Available:http://www.rockstargames.com/maxpayne/splash.html`.

Riemer, V. and C. Schrader (2015). "Learning with quizzes, simulations, and adventures: Students' attitudes, perceptions and intentions to learn with different types of serious games". In: *Computers Education* 88, pp. 160–168. DOI: `http://www.sciencedirect.com/science/article/pii/S0360131515001232`.

Roblox, Co. (2006). *Roblox official site*. URL: `https://www.roblox.com`.

Ronco, C. and E. Benini (2013). “A Simplex-Crossover-Based Multi-Objective Evolutionary Algorithm”. In: *IAENG Transactions on Engineering Technologies* 247, pp. 583–598. DOI: https://link.springer.com/chapter/10.1007%2F978-94-007-6818-5_41.

Salen, K. and E. Zimmerman (2004). *Rules of play : game design fundamentals*. Vol. 1. Massachusetts Institute of Technology printing press.

Saljo, R. (1979). “Learning in the learner’s perceptive: some common-sense conceptions”. In: *Reports from the Institute of Education, University of Gothenburg*.

Sandberg, J., M. Maris, and P. Hoogendoorn (2014). “The added value of a gaming context and intelligent adaptation for a mobile learning application for vocabulary learning”. In: *Computers Education* 76, pp. 119–130. DOI: https://doi.org/10.1016/j.compedu.2014.03.006.

Sandor, M. et al. (1998). “Evaluating critical thinking skills in a scenario-based community health course.” In: *Journal of Community Health Nursing.* 15.1, pp. 21–29.

Scarlett, M. (2015). “Gaming Geography: Using Minecraft to Teach Essential Geographic Skills.” In: *Society for Information Technology Teacher Education International Conference*, pp. 838–840.

Schmidt, R. and Y. Watanabe (2001). “Motivation, strategy use, and pedagogical preferences in foreign language learning”. In: *Motivation and second language acquisition* 23, pp. 313–359.

Scott, J., R. Markert, and M. Dunn (1998). “Critical thinking: change during medical school and relationship to performance in clinical clerkships”. In: *Medical Education* 32, pp. 14–18.

Sedano, C. et al. (2013). “Collaborative and cooperative games: Facts and assumptions”. In: *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 370–376.

Seldon, S. (2022). *Neural Network Models Explained*. URL: https://www.seldon.io/neural-network-models-explained.

Sener, J. (1997). “Constructivism: Asynchronous Learning Networks”. In: *ALN Magazine* 1.1.

Shabihi, N. and F. Taghiyareh (2017). “Toward a personalized game-based learning environment using personality type indicators”. In: *Research Gate*.

Shaker, N., J. Togelius, and M. Nelson (2016). *Procedural Content Generation in Games Computational Synthesis and Creative Systems*. Vol. 1. Springer.

Sharma, M. et al. (2007). “Towards player preference modeling for drama management in interactive stories”. In: *20th International FLAIRS Conference*, pp. 571–576.

Shi, Y. and B. Shih (2015). “Game factors and game-based learning design model”. In: *International Journal of Computer Games Technology* 11. DOI: https://doi.org/10.1155/2015/549684.

Shih, B., K. Koedinger, and R. Scheines (2010). “Discovery of Student Strategies using Hidden Markov Model Clustering”. In: *6th International Conference on Educational Data Mining*.

Shunk, D. (2000). *Learning theories: An educational perspective*. Vol. 3. Upper Saddle River New Jersey.

Sivanandam, S. (2013). *Principles of soft computing*. Vol. 1. John Wiley and Sons.

Skinner, B. (1983). “Origins of a Behaviorist”. In: *Psychology Today*, pp. 22–33.

Soflano, M., T. Connolly, and T. Hainey (2015). “An application of adaptive games-based learning based on learning style to teach SQL”. In: *Computers Education* 86, pp. 192–211. DOI: https://doi.org/10.1016/j.compedu.2015.03.015.

Sullivan, A., M. Mateas, and N. Wardrip (2010). “Rules of engagement: Moving beyond combat-based quests”. In: *Intelligent Narrative Technologies III Workshop* 11.1, pp. 1–8.

Sultan, W., P. Woods, and A. Koo (2011). “A constructivist approach for digital learning: Malaysian schools case study”. In: *Journal of Educational Technology Society* 14.4, pp. 149–163.

Tadayon, M. and G. Pottie (2020). "Predicting Student Performance in an Educational Game Using a Hidden Markov Model". In: *IEEE Transactions on Education* 63.4, pp. 299–304. DOI: 10.1109/TE.2020.2984900.

Tahmasebi, P. and A. Hezarkhani (2011). "Application of a Modular Feedforward Neural Network for Grade Estimation". In: *Natural Resources Research* 20.1, pp. 25–32. DOI: 10.1007/s11053-011-9135-3.

Thompson, J. (2009). "To question or not to question: The effects of two teaching approaches on students thinking dispositions, critical thinking skills, and course grades in a critical thinking courses". In: *Capella University press*.

Thomson, A. (2001). *Critical Reasoning: A Practical Introduction*. Vol. 3. Routledge.

Thue, D. et al. (2007). "Interactive storytelling: A player modelling approach". In: *Conference on Artificial Intelligence and Interactive Digital Entertainment* 3.1, pp. 43–48.

Valve (2008). "Left 4 Dead". In: *Developer design documents*. DOI: Available:http://www.l4d.com/game.html.

— (2009). "Left 4 Dead 2". In: *Developer design documents*. DOI: Available:http://www.l4d.com/game.html.

Veermans, K. (2019). "Pedagogy in Educational Simulations and Games". In: *Gaming Media and Social Effects, Springer*, pp. 5–14. DOI: https://doi.org/10.1007/978-981-13-2844-2_2.

Verduin, M. et al. (2013). "Computer Simulation Games as an Adjunct for Treatment in Male Veterans with Alcohol Use Disorder". In: *Journal of substance abuse treatment* 44.3, pp. 316–322. DOI: 10.1016/j.jsat.2012.08.006.

VicariousVisions (2017). "Crash Bandicoot N. Sane Trilogy". In: *Sony Computer Entertainment Developer Diaries* 1. DOI: https://web.archive.org/web/20220320043643/https://www.vvisions.com/.

Virvou, M., G. Katsionis, and K. Manos (2005). "Combining Software Games with Education: Evaluation of its Educational Effectiveness". In: *Journal of Educational Technology Society* 8.2, pp. 54–65.

Vittori, V. et al. (2000). "Intellectual ability, learning style, personality, achievement motivation and academic success of psychology students in higher education". In: *Personality and Individual Differences* 29.6, pp. 1057–1068.

Wang, P. et al. (2017). "Interactive Narrative Personalization with Deep Reinforcement Learning". In: *International Joint Conference on Artificial Intelligence* 26, pp. 3852–3858.

Wertsch, J. (1997). *Vygotsky and the formation of the mind*. Vol. 1. Cambridge Massachusetts Printing press.

Westra, J. and F. Dignum (2010). "Keeping the trainee on track". In: *IEEE Conference on Computational Intelligence and Games*, pp. 450–457.

Wheeler, L. and S. Collins (2003). "The influence of concept mapping on critical thinking in baccalaureate nursing students." In: *Journal of Professional Nursing* 19.6, pp. 339–346.

Yang, Y. (2008). "A catalyst for teaching critical thinking in a large university class in Taiwan: asynchronous online discussions with the facilitation of teaching assistants." In: *Educational Technology Research and Development* 56, pp. 241–264.

Yannakakis, G. (2012). "Game ai revisited". In: *9th Conference on Computing Frontiers*, pp. 285–292. DOI: https://doi.org/10.1145/2212908.2212954.

Yannakakis, G. and J. Togelius (2015). "Experience-driven procedural content generation". In: *International Conference on Affective Computing and Intelligent Interaction*, pp. 519–525. DOI: 10.1109/ACII.2015.7344619.

Yilmaz, K. (2011). “The Cognitive Perspective on Learning: Its Theoretical Underpinnings and Implications for Classroom Practices”. In: *The Clearing House: A Journal of Educational Strategies, Issues and Ideas* 84.6, pp. 204–212. DOI: 10.1080/00098655.2011.568989.

Zadeh, L. (1965). “Fuzzy sets”. In: *Information and Control* 8, pp. 338–353.

Zarraonandia, T. (2015). “Designing educational games through a conceptual model based on rules and scenarios.” In: *Multimedia Tools and Applications* 74.13, pp. 4535–4559.

Zorn, C. et al. (2013). “Exploring Minecraft as a conduit for increasing interest in programming”. In: *Proceedings of the International Conference on the Foundations of Digital Games*, pp. 352–359.