FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Virtual Environments' authoring tool for an IEER (Immersive Educational Escape Room)

Marcelo Augusto Reis

Master in Informatics and Computer Engineering

Supervisor: António Fernando Vasconcelos Cunha Castro Coelho (PhD)

Co-Supervisor: Diana Quitéria Teixeira de Sousa (MSc)

July 20, 2023

Virtual Environments' authoring tool for an IEER (Immersive Educational Escape Room)

Marcelo Augusto Reis

Master in Informatics and Computer Engineering

Approved in oral examination by the committee:

President: Prof. Jorge Manuel Gomes Barbosa

Referee: Prof. Hugo Miguel Gonçalves Crespo Machado Silva

Referee: Prof. António Fernando Vasconcelos Cunha Castro Coelho

July 20, 2023

Resumo

Este estudo teve como finalidade a criação de uma framework de geração de objetos interativos para implementar um percurso de aprendizagem definido, a um nível elevado, por um utilizador não informático. O objetivo consistiu no desenvolvimento de uma framework para a criação automática de um ambiente virtual imersivo em cada Escape Room Educativo Imersivo (designado nesta dissertação por Immersive Educational Escape Room - IEER), que foi explorado em Realidade Virtual (VR). Cada objeto pode ser associado a animações e ações/interações para que o utilizador possa mergulhar na imersão. Esta framework possibilita alterar os conteúdos, dependendo de diferentes condicionantes que são totalmete personalizáveis pelo docente. Deste modo, há possibilidade de modificar a narrativa de jogo e, por consequência, das diferentes mecânicas de jogo dos desafios propostos dentro dos IEERs. Estas salas, integrando objetos interativos, podem ser também alteradas através da Geração Procedimental de Conteúdo (PCG).

Uma framework como esta necessitou da integração de duas soluções de software, Houdini para PCG e Unity3D para o desenvolvimento do jogo. Foi desenvolvida uma ontologia para fazer a ponte entre estes dois sistemas de software baseada em JavaScript Object Notation (JSON).

Para testar esta framework, foi criado um grupo focal composto por alguns especialistas em educação, ou seja, professores e formadores que frequentam a unidade curricular de Software Educativo do Mestrado em Multimédia (UPorto: FEUP).

A metodologia de investigação adotada é a "Design-Based Research (DBR)", visto que é a que melhor se adapta para analisar o problema, formular, testar e refinar soluções de forma incremental. Deste modo, pretendeu-se atingir e desenvolver um excelente produto final, através do resultado da co-criação com uma equipa multidisciplinar.

Uma das limitações, que afetou o desenvolvimento deste trabalho, foi o número diferente de desafios que a framework poderá permitir. Foi também possível verificar que a framework poderá não suportar todos os desafios que o professor possa querer introduzir.

Palavras-chave: Educação, Imersão, Realidade Virtual, Aprendizagem, PCG, Sala de Fuga, IEER

Classificação ACM: Metodologias informáticas → Computação gráfica → Sistemas gráficos e interfaces → Realidade Virtual, Computação Aplicada → Educação → Ambientes de aprendizagem interactivos

Abstract

This study aimed to create a framework for generating interactive objects to implement a learning path defined, at a high level, by a non-computing professional user. The goal consisted of developing a framework for automatically creating an immersive virtual environment in each Immersive Educational Escape Room (designated in this dissertation as IEER) explored in Virtual Reality (VR). Each object can be associated with animations and actions/interactions so the user can dive into immersion. This framework makes it possible to change the contents, depending on different conditions, which are totally customisable by the teacher. In this way, modifying the game narrative and, consequently, the different game mechanics of the challenges proposed within the IEER is possible. These rooms, integrating interactive objects, can also be changed through Procedural Content Generation (PCG).

A framework like this required the integration of two software solutions, Houdini for PCG and Unity3D for game development. An ontology was developed to bridge these two software systems based on JavaScript Object Notation (JSON).

To test this framework, a focus group was created composed of some education specialists, i.e. teachers and trainers attending the subject Educational Software of the Master's Degree in Multimedia (UPorto: FEUP).

The research methodology adopted was the Design-Based Research (DBR), since it was the best suited to analyse the problem, formulate, test, and incrementally refine solutions. In this way, the aim was to develop an excellent final product through co-creation with a multidisciplinary team.

One limitation that affected this study's development was the different number of challenges the framework could allow. It was also possible to verify that the framework may not support all the challenges the teacher may want to introduce.

Keywords: Education, Immersive, Virtual Reality, Learning, PCG, Escape Room, IEER

ACM Classification: Computing methodologies → Computer graphics → Graphics systems and interfaces → Virtual Reality, Applied Computing → Education → Interactive Learning Environments

Acknowledgements

I would like to thank everyone who has accompanied me on this long journey and helped me reach the goal of writing my dissertation and concluding my master's degree.

First, I want to thank all the nurses, doctors and physiotherapists who helped me heal and recover from a health problem I had in my final year of secondary school.

To all the teachers and staff of the Grouping of Schools Tenente-Coronel Adão Carrapatoso from Vila Nova de Foz Côa who, due to my health problem, supported and motivated me, took care of all the procedures so that I could do the exams and not lose a year (the mathematics exam was done in the school and the Portuguese in the hospital), helped me to prepare for the national exams, made me a welcome-back party and gathered some funds so I could buy my first laptop (which is still being used to write this dissertation).

To all my friends for supporting and motivating me in difficult moments and when I needed it the most.

To my family for giving me everything I needed to achieve the goal of studying at the Faculty of Engineering of the University of Porto and for always supporting me in my decisions and in what I wanted for my life.

To my supervisor and co-supervisor, Professor António Coelho and Professor Diana Sousa, for accepting me in this challenge, always being there when I needed them, worrying about me, sharing their knowledge and believing that I would achieve good results.

To my colleague Guilherme Gama for coworking with me and sharing knowledge essential to my dissertation.

Last but not least, I would like to dedicate all my success in my academic path to the former headmaster of Vila Nova de Foz Côa school, Professor Jorge, who helped me a lot when I was forced to be absent from school due to a health problem. Unfortunately, he passed away, and I did not have the opportunity to thank him properly, but I have the conviction that he would be happy with what I have achieved.

Thank you all for helping me achieve this goal. I am eternally grateful to all of you.
Marcelo Augusto Reis

“Success is not the key to happiness. Happiness is the key to success.
If you love what you are doing, you will be successful.”

Albert Schweitzer

Contents

List of Figures

List of Tables

Listings

Abbreviations and Symbols

AI	Artificial intelligence
API	Application Programming Interface
CAEO	Correct Answer Element Object
DBR	Design-Based Research
DQN	Deep Q-Network
EA	Evolutionary Algorithm
EER	Educational Escape Room
ER	Escape Room
FEUP	Faculty of Engineering of the University of Porto
FOV	Field of View
GA	Genetic Algorithm
GAN	Generative Adversarial Networks
IEER	Immersive Educational Escape Room
INESC TEC	Institute for Systems and Computer Engineering, Technology and Science
JSON	JavaScript Object Notation
MEEC	Master in Electrical and Computer Engineering
MEIC	Master in Informatics and Computer Engineering
MM	Master of Multimedia
NPC	Non-Player Character
PCG	Procedural Content Generation
PCG-G	Procedural Content Generation for Games
PDIS	Dissertation Preparation
PLO	Pedagogical Learning Objective
QD	Quality-Diversity
RL	Reinforcement Learning
TTS	Text-To-Speech
URL	Uniform Resource Locator
UX	User Experience
VARK	Visual, Aural, Read/Write and Kinaesthetic
VR	Virtual Reality
WAEO	Wrong Answer Element Object

Chapter 1

Introduction

This chapter covers some introductory aspects regarding this project, like context, motivation, objectives, and document structure.

1.1 Context

Higher education has reinvented new pedagogical strategies to engage students and foster the acquisition of 21st-century skills through the development of teaching, learning, and assessment processes. However, when it comes to bringing new technological approaches to the classroom, it is not easy since, in general, teachers do not have the necessary technical knowledge to develop these types of pedagogical tools.

Theoretical lessons that implement the expository pedagogical methodology tend to be more tedious (from a student's point of view) and much more exhausting for the teacher than any other practical lesson. Moreover, subjects implementing more expository pedagogical methodologies tend to increase students' demotivation. They start skipping classes, lose essential skills, and decrease productivity, consequently influencing their professional future.

Bringing the game's world into education has shown that significant benefits can be gained. Educational games are an excellent example of that, becoming increasingly used nowadays. However, their full potential is far from being reached, the advance of technology as shown so far being limitless, providing a new horizon of possibilities for improvement.

Immersive Educational Escape Room is the leading research topic of this document, and if well implemented can be used as a tool for teachers to improve their teaching methods, allowing them to have the power to adapt the escape room to any topic, theme, difficulty level, objective, and many others available factors.

This research enhances what was achieved by the EdScape team (FEUP/INESC TEC) in the previous academic year, composed of Professor António Coelho, Diana Sousa (who is developing research for her doctoral thesis focused on IEERs), Professor Manuel Firmino Torres, Professor Armando Sousa, Professor Fátima Monteiro, and the recent Masters who defended their Master Thesis recently - Ana Rita Santos (MEIC), Ana Rita Garcia (MM), Tiago Rossini (MM), and Celina

Zheng (MEEC). The research is also being held with a colleague Guilherme Gama (finishing his master's degree in Multimedia).

1.2 Motivation

Games have been a part of our lives since we were children, first with more physical ones like puzzles, then later with digital games. Although there are many types of games out there, we enjoy multiplayer games and those that allow us to gain some knowledge. It is interesting to notice that video game knowledge, in general, is more easily acquired than, for example, knowledge from reading books. So, why not bring educational games into education since it benefits learning? This is one of the reasons why we are doing this research.

The application of procedural generation in VR environments within the context of an IEER is an innovative topic that also impacts learning processes. An IEER is an interactive learning experience that combines escape room elements with educational objectives. Participants engage in themed challenges and puzzles in physical or virtual environments to promote critical thinking, collaboration, and knowledge acquisition. IEERs aim to create an immersive and enjoyable learning atmosphere, encouraging active participation and teamwork. Using PCG as a tool for generating different playable Escape Rooms every run is extremely interesting, which has led us to choose this theme.

1.3 Problem Statement

1.3.1 Research Problem

The technological advance in gaming opened the door to new educational games. The challenge of bringing new technological approaches to the classroom is that, in general, teachers lack the technical knowledge to create such pedagogical tools. The available educational games are limited and are primarily located on the web, requiring a good internet connection. So far, not many educational games implemented in VR with PCG have been found, which is an opportunity for an investigation and possible implementation within this niche market.

The use of PCG in IEERs presents a promising research opportunity. However, several challenges need attention. These include aligning generated content with educational objectives, maintaining coherence and quality, balancing difficulty levels, exploring curriculum alignment, and investigating the impact on participant engagement and learning outcomes. Addressing these challenges contributes to a better understanding of how PCG can create meaningful and compelling learning experiences in IEERs.

1.3.2 Research Objectives

The main goal of this research is to create a framework to generate escape rooms and a set of interactive objects that allow the implementation of a learning path defined, at a high level, by a non-computing professional user. Thus, as expected results, we aim to create rooms and a group of interactive objects, using PCG "to provide content that is playable, of a high quality, and yet different from other content that came before or after" (Gravina et al., 2019). The application of virtual reality produces a positive impact from the new experiences in the form of increased memory strength associated with the playing experience, which speeds up players' learning process (David et al., 2019).

In each game level, there are a particular set of challenges, and each challenge, when solved, provides the student with a hint to complete the final challenge, which corresponds to the IEER at the last level. In this last level, the student is asked to enter a code that serves as a key to open the door to the exit IEER room. All this can be easily configurable in the backoffice application allowing the teacher to trace the learning paths, intuitively and practically, that they want.

The development of this framework requires integrating two software solutions, Houdini for the PCG and Unity3D for game development. An ontology was developed based on JSON to understand the bridge between these two software better.

The specific objectives of this project were as follows: (1) To select an ontology for the specification of the learning paths; (2) To generate the interactive objects from this specification; (3) To generate "realistic" rooms to integrate previously defined interactive objects; (4) To define the layout of the interactive objects and other "decorative" objects in rooms that maximise the student's user experience (UX); (5) To test and evaluate the developed prototype.

Finally, an article will be published with all the team members with some of the present research results.

1.3.3 Research Questions

To achieve the mentioned objectives, the following research questions were formulated:

- How can a teacher's learning path be translated into a simple graph?
- How to generate an IEER environment from a simple graph?
- What is the diversity of IEER generated by PCG?

1.4 Document Structure

This document is composed of five chapters, excluding the introduction. It is organised as follows: Chapter 2 (State of the Art) presents the background, related work, existing technologies, and approaches to the same or associated problems; Chapter 3 (Methodology) describes the research methodology adopted, the work plan followed during the dissertation development and how the project was managed; Chapter 4 (Design and Development of a PCG solution for IEER) addresses the design aspects, implementation details of the virtual environments' authoring tool and also presents the proposed prototype; Chapter 5 (Results and Evaluation) shows the results obtained in an experimental study with a focus group, followed by its evaluation, limitations and contributions;

Chapter 6 (Conclusions and Future Work) presents the conclusions that have been drawn from this work and propose some future topics that would be relevant to explore. To conclude this Thesis, the references and appendices used to develop it are presented at the end.

Chapter 2

State of the Art

This chapter presents the background, related work, existing technologies, and approaches to the same or related problems.

2.1 Introduction

Introducing games and new technologies in an educational context has improved student learning. However, the full potential of games and new technologies is not fully exploited in education. As technology advances, new ways that can be explored to improve learning arise, such as VR.

In this chapter, we introduce the main topics of this research, such as IEERs, VR environments, and PCG. Regarding EERs, we start by talking about the ERs and then the education application in ERs. In addition to this topic, we abord playful and immersive learning describing their impact on the student's learning process. Furthermore, we describe the PCG and identify where it can be helpful, enumerating its advantages and disadvantages in game development. The application of PCG in games, educational games and object placement are also topics that we abord. Finally, the last theme we discuss is the combination of PCG and IEERs, where we explore what these two can achieve together in the educational context. A relevant group of studies is analysed to enrich some of these topics, where valuable results were obtained.

In the last section of this chapter, we summarise the most vital aspects retained from the research of previous related works and problems, background, existing technologies, and approaches.

2.2 Educational Escape Rooms (EERs)

Escape Rooms (ER) have appeared in conventional games and are gaining popularity recently. This kind of game type is characterised by being "a physical adventure game where players solve a series of puzzles and puzzles using instructions and strategies to complete a goal" (David et al., 2019) in a limited amount of time to escape from the ER. These puzzles, usually hidden objects, for example, mathematical calculations, finding clues, sorting images, teamwork/team communication, and others can be of any kind.

Recently ERs have captured the interest of educators and institutions. Lopez-Pernas et al. (2019) and Reuter et al. (2020) explained that this occurred due to their ability to foster in students valuable soft skills considered essential for successful careers, such as teamwork, leadership, creative thinking, communication, and many others. Consequently, ERs are emerging as a learning activity that offers highly engaging experiences to enhance students' learning. By incorporating the course materials into puzzles, this learning activity ensures learners master these materials to succeed, i.e., to exit the ERs. The absorption of knowledge is more effective when students actively participate in class, practice it, and participate in discussions about it.

Considering what was stated above, EERs games emerged. Games that are equal to ERs but are designed for teaching. Lopez-Pernas et al. (2019) argue that ERs may seem like a good fit for younger students, but this type of gamified experience is mainly regarded as an attraction for adults, so students in higher education would benefit significantly from them.

However, EERs still need to be more widespread in educational institutions (schools and universities). This does not happen mainly because teachers need to create their EERs according to the specific learning needs of their course. Creating this learning activity takes time and requires teachers to have specific knowledge.

When creating an ER, some crucial aspects need to be considered. According to Reuter et al. (2020), "it must be considered who the participants will be, what the objective will be, as well as the development of a theme, and the creation of specific puzzles." Likewise, good development practices should be applied, such as testing equipment and materials needed for the activities and evaluating the final product. Figure 2.1, Reuter et al. (2020) show the main steps to develop an ER.

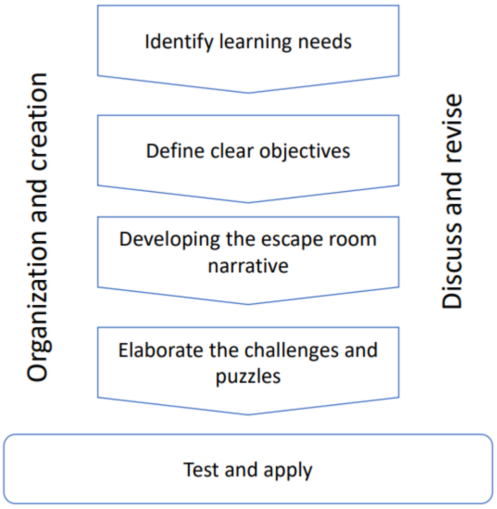

Organization and creation
Identify learning needs
Define clear objectives
Developing the escape room narrative
Elaborate the challenges and puzzles
Discuss and revise
Test and apply

Figure 2.1: Main steps in creating Escape rooms by Reuter et al. (2020)

When the student engages in the EER, his learning cycle follows some steps, as shown in

Figure 2.2. Reuter et al. (2020) describe that the first moment the students get in touch with this activity, they get surprised and start exploring the available tools. While trying to discover some clues and understand the game flow, students get what they need to do to solve the puzzles. During this activity, crucial skills will take place between students, such as teamwork, collaboration, communication, and critical thinking, to solve the challenges faced.

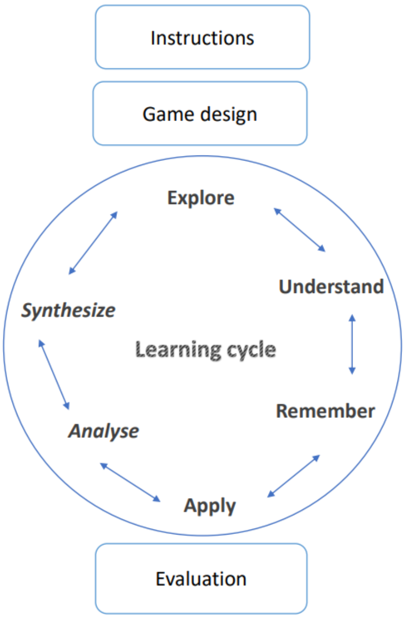


Figure 2.2: Student learning cycle in the Escape Room by Reuter et al. (2020)

Creating a flowing environment in EERs is essential because students, when involved in this flow, are engaged and focused only on the task they are doing. This state is reached, according to Nicholson (2015), "when the players are challenged and entertained" (as cited in Reuter et al., 2020). Students will feel frustrated, unmotivated, and bored if the task is too easy or tricky. Well-designed games prevent this from happening. If the EER is well developed, the chance of students entering this desired learning flow will increase.

Lastly, to close this topic of EERs, we will talk about two studies: (1) Lopez-Pernas et al. (2019) presented a pilot experience of an EER conducted in a programming course. This course is part of the Bachelor's Degree in Telecommunications Engineering from UPM (Universidad Politécnica de Madrid); (2) Borrego et al. (2017) presented a teaching experience of an ER activity developed at the Universitat Autonoma de Barcelona Engineering School. Computer Networks and Information and Security were two subjects in the second year of the Degree in Computer Engineering, where this activity was designed to increase student motivation.

In the first study, Lopez-Pernas et al. (2019) argue that most students enjoyed the experience, leaving positive comments and wishing that other courses embrace something close to this. The results can conclude that EERs have been effective in teaching programming. Another fact is that students preferred EERs over traditional computer lab sessions.

In the second study, Borrego et al. (2017) reported that the results were highly positive. There has been an increased motivation and willingness to learn among students due to participating in the ER access challenges in both subjects. As well as that, teachers have gained experience designing and organising escape games at the university level.

2.2.1 Playful Learning

Playful learning consists of "learning activities embedded with playful engagement and exploration when using novel tools and technologies in learning" (Kangas et al., 2017). Activities that are aimed to enhance the learning process of the learner. Embracing play, games, and technology affordances into curriculum-based learning is emphasised in this concept based on existing perspectives of play and playfulness in learning (Kangas et al., 2017). Piaget (1999) and Papert (1980) highlight that higher education institutions use less playful approaches to learning and teaching than children's learning environments (as cited in Rice, 2009).

The arrival of video games aroused psychologists' interest since playing digital games produced different psychological effects on the player's mind. Within these effects are positive ones, such as motivation, happiness, and intermittent, but also negative ones, such as boredom and frustration. Bad-developed games, that are too easy or difficult, tend to generate adverse effects on the player. According to Loftus and Loftus (1983), "good games aim for the "sweet spot" where players can succeed, but only with some struggle" (as cited in Plass et al., 2014).

According to Kangas et al. (2017), "playful learning is a useful approach for using novel tools and learning environments because it allows the use of imagination and a playful attitude toward learning and experimenting."

Plass et al. (2014) characterised playful learning in three ways:

- **Motivation:** There are two types which are intrinsic and extrinsic. Extrinsic motivation is related to all the game elements that incentives the learner, like trophies and rewards. On the other hand, intrinsic motivation is related to game design mechanics and activities that produce positive psychological effects in the learner, like happiness and internment.

- **Adaptivity:** An adaptive game can be designed to adjust to each learner according to his specific need, for example, the learner's current level of knowledge.

- **Self-Regulated Learning and Graceful Failure:** Learning from mistakes is an excellent analogy to graceful failure. Failure is an essential outcome in our learning process, making us try other alternatives and solutions for that specific problem until the solution is reached. Together with graceful failure, self-regulated learning is the goal to provide opportunities during the play, according to Barab et al. (2008), "where the player executes strategies of goal setting, monitoring of goal achievement, and assessment of the effectiveness of the strategies used to achieve the intended goal" (as cited in Plass et al., 2014).

Plass et al. (2014) suggest that to foster and enhance learners' cognitive engagement with learning mechanics, design features that use cognitive, affective, and sociocultural engagement, along with physical and behavioural engagement, are essential. In Figure 2.3, it is possible to observe the framework of playful learning by Plass et al. (2014), and at the bottom, we can find three pillars that summarise their experience in learning game design.

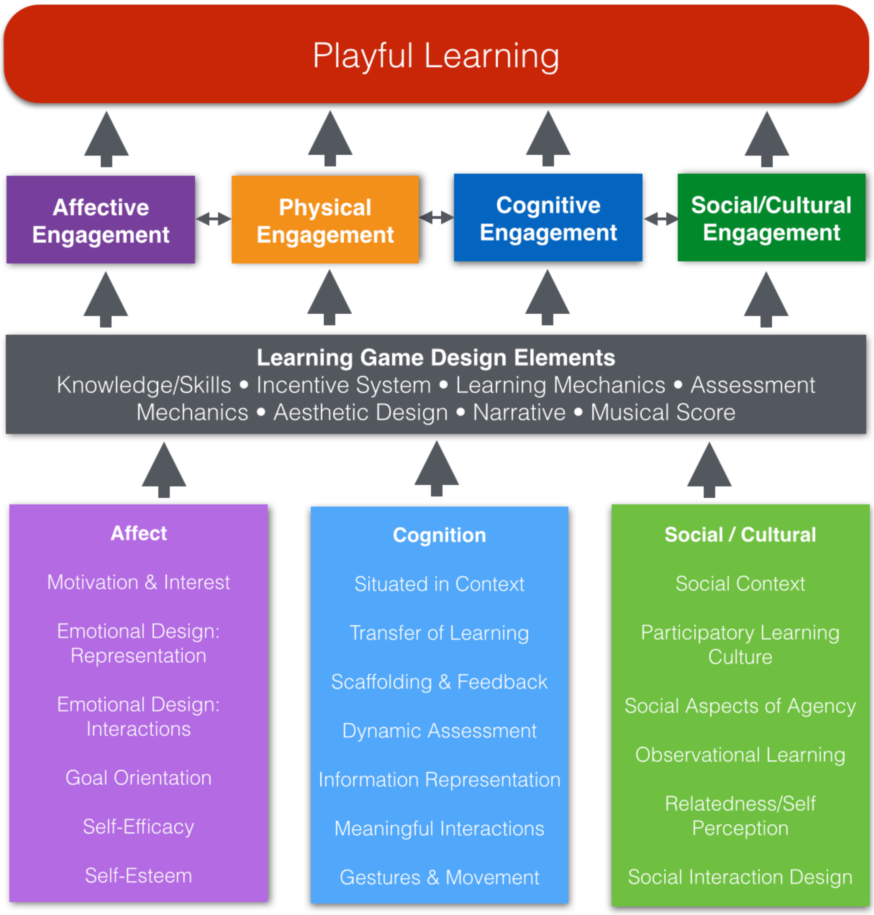

Playful Learning
Affective Engagement
Physical Engagement
Cognitive Engagement
Social/Cultural Engagement
Learning Game Design Elements
Knowledge/Skills • Incentive System • Learning Mechanics • Assessment Mechanics • Aesthetic Design • Narrative • Musical Score
Affect
Motivation & Interest
Emotional Design: Representation
Emotional Design: Interactions
Goal Orientation
Self-Efficacy
Self-Esteem
Cognition
Situated in Context
Transfer of Learning
Scaffolding & Feedback
Dynamic Assessment
Information Representation
Meaningful Interactions
Gestures & Movement
Social / Cultural
Social Context
Participatory Learning Culture
Social Aspects of Agency
Observational Learning
Relatedness/Self Perception
Social Interaction Design

Figure 2.3: Integrated design framework of Playful Learning by Plass et al. (2014)

Lastly, to close this topic of playful learning, we are going to talk about three studies: (1) Plass et al. (2014) proposed an integrated playful learning framework for each function of games for learning in Table 2.1, which will explain and predict the learning processes in a wide range of playful learning environment genres; (2) Kangas et al. (2017) investigated three elementary schools in Finland and the Netherlands to explore how students are satisfied with playful learning environments and how teacher engagement affects students' satisfaction with those environments; (3) Rice (2009) explored the playful learning potential in a Planning and Architecture school using a focus group and a questionnaire. In this study, the playful aspects of the 'dérive' (created in 1954 by the Situationists to understand an environment) were used to analyse the use of playful learning environments in learning and teaching.

Table 2.1: Function of games for learning by Plass et al. (2014)

Function	Explanation/ Main role of the game
Preparation of future learning	Sharing of experiences with students that can be incorporated into classroom discussion.
Teach new knowledge and skills	Include new knowledge and skills as part of the game.
Practice and reinforce existing knowledge and skills	Make it possible to automate existing knowledge or skills.
Develop 21st Century Skills	Encourages teamwork, collaboration, problem-solving, creativity, communication, etc.

In the first study, Plass et al. (2014) argue that the combined three perspectives mentioned earlier (cognitive, affective, and sociocultural) are essential for game design, its research to be fully captured and to provide the necessary game elements to offer them for learning. Then, considering playful learning as a series of learner engagements on different levels (like cognitive, affective, physical, and social/cultural) and game design elements as strategies to achieve this type of engagement based on the established levels potentiates the contribution on a more systematic process of conceptualising and designing games.

Regarding the second study, Kangas et al. (2017) argue that the teacher's involvement and pedagogical choices significantly impact how well the learning environment meets the needs of the students. Teachers adapt and develop educational approaches for the benefit of students. It is necessary to awaken teachers' interests and motivations to create pedagogical methods and curricula that encourage joyful learning if successful teaching and learning occur in such an atmosphere.

In the third study, Rice (2009) argues that learning and teaching can be effectively enhanced using playful approaches, making learning more exciting, enjoyable and interesting. In addition to motivating learners, engaging them, and allowing them to approach a subject from different perspectives, playful learning can also be viewed as part of the VARK (Visual, Aural, Read/Write and Kinaesthetic) learning approaches. Occasionally, playful approaches to knowledge and learning resulted in ontological changes in students. A creative and imaginative atmosphere was created through play where "normal" behaviour and practice were suspended, and students could play with boundaries, concepts, and disciplines more freely, becoming more confident, which improved through playful experiences.

2.2.2 Immersive Learning

Immersion is, according to Witmer and Singer (1998), a "psychological state characterized by one's perceptions of presence and interaction" (as cited in Beck et al., 2020). This can be achieved through technology, narrative and the challenges faced. The concept of immersion is also connected to interest and focus on what is happening. When these two, interest and focus, are met, immersion is one step away, making the person abstract from the real world and engage in the immersive environment.

Following the perspective of Nilsson et al. (2016), which can be seen in Figure 2.4, technology, narrative, and challenges all play a role in this psychological phenomenon (as cited in Beck et al., 2020).

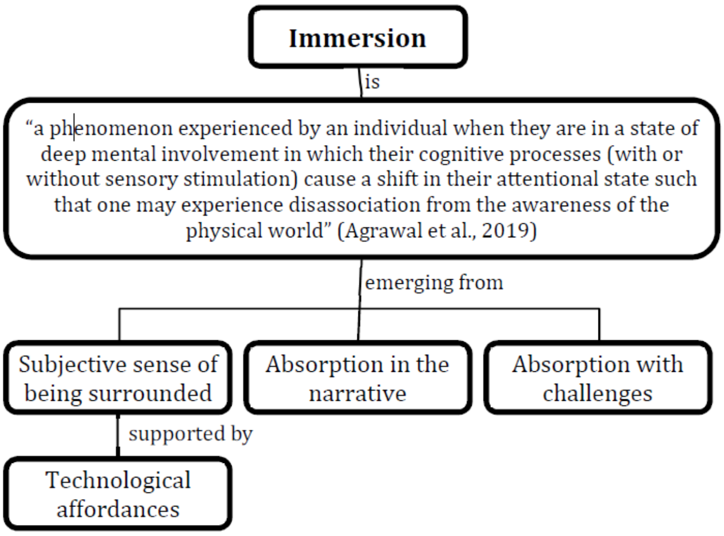

Immersion
is
"a phenomenon experienced by an individual when they are in a state of deep mental involvement in which their cognitive processes (with or without sensory stimulation) cause a shift in their attentional state such that one may experience disassociation from the awareness of the physical world" (Agrawal et al., 2019)
emerging from
Subjective sense of being surrounded
Absorption in the narrative
Absorption with challenges
supported by
Technological affordances

Figure 2.4: Syncretic perspective on Immersion by Agrawal et al. (2019) and Nilsson et al. (2016), as cited in Beck et al. (2020)

Incorporating immersive technologies into education can assist students, most particularly Generation Z, in acquiring skills that are much more difficult to achieve using traditional pedagogical methods, as they prefer learning from new technological methods to learning from conventional means. The use of these technologies, according to Fonseca et al. (2014), enhances engagement and, according to Huang et al. (2010), increases participation (as cited in Kuhail et al., 2022). So, in this sense, in this specific research, authors reinforce that immersive learning improves the student learning process by, according to Slater et al. (2003) and Dalgarno et al. (2010), "inducing a sense of presence (the feeling of being there), co-presence (the feeling of being there together), and the building of identity (connecting the visual representation to the self)" (as cited in Kuhail et al., 2022).

As David et al. (2019) point out, virtual reality can create a novel playing experience. New experiences are expected to positively impact memory strength, which is likely to result from the experience of playing. The students' learning process is anticipated to be sped up by the boost in memory power.

VR technology has been recently used to allow interaction, through senses and perception, with virtual environments, i.e., environments generated by computers. This technology has the potential to generate very immersive environments like the real world and even create experiences that in the real world are not possible. But unfortunately, when this technology is not well applied or developed, it can cause some inconveniences for the users. Some of these are described as follows:

- **Motion Sickness:** results from the discrepancy between what the user sees and what the inner ear detects. Vertigo, nausea, and vomiting may result from this.
- **Cybersickness:** a collection of symptoms similar to motion sickness that appears without actual physical motion. The three symptoms are nausea, oculomotor problems, and general disorientation.
- **Simulator Sickness:** motion sickness, known as simulator sickness, is brought on by interacting with a simulated environment. Simulator sickness is characterised by fatigue, nausea, vomiting, perspiration, headaches, unease, drowsiness, disorientation, and ocular motor abnormalities.

Lastly, to close this topic of immersive learning, we will talk about two studies: (1) Beck et al. (2020) proposed finding gaps in every survey conducted in immersive learning environments. Their focus was to see what the panorama of immersive learning environments was. Moreover, they search 47 papers, collecting all the potential contributions on this topic; (2) David et al. (2019) created an Escape Room game in VR called Locked Out, which is compatible with Samsung Gear VR and may be played on Android smartphones. The GAT labs' personnel and managers were contacted to provide feedback on this game.

In the first study, Beck et al. (2020) argue that "to analyze the impact of technology in learning, it is necessary to consider not just technology itself, but its overall context". The results have shown that relevant gaps exist in immersive learning environments, contributing to the need for future research in this area. They found these voids in the immersive learning environment knowledge: "Internet of Things, Interactive Escape Rooms, Mixed Reality Arcades, smart board games and similar (Void 1); the need for revisiting the use of traditional immersive environments like game books and traditional role-playing games (Void 2); the need to push for ambitious high-narrative, high-challenge, high-system environments, such as encompassing embodied participation in metaverses and highly present remote operation of physical equipment (Void 3)".

In the second study, David et al. (2019) argue that the evaluation's participants generally well-liked the Locked Out game. According to the study's findings, it is possible to say that fresh experiences in virtual reality (VR) boost the subject's memory of how to play the Locked Out game, which means that this improvement in memory power sped up the subject's learning process.

2.3 Procedural Content Generation (PCG)

PCG has been recently increasing its popularity in the field of game development due to the following reasons. PCG is based on the concept of randomness, i.e., by setting some parameters, it ensures the generation of an infinite number of possible game contents different from each other. Moreover, PCG can generate personalised game content considering the player's preference to improve their playing experience (Amato, 2017). Hendrikx et al. (2013) state that creating engaging and realistic experiences is possible by procedurally generating entity behaviour based on player involvement and action.

PCG seeks to automatically offer playable, high-quality, and unique game content that is distinct from other stuff that came before or later, whether before beginning a game level or even continually during gameplay (Gravina et al., 2019) using algorithms or processes to reduce the game design and development costs. Work that human designers manually do can be replaced by computers that execute a well-defined procedure, which is cheaper.

However, the content generated using PCG must satisfy specific minimal criteria of playability. According to Gravina et al. (2019), “content which do not satisfy these criteria of quality can break the gameplay explicitly or implicitly, resulting in a poor player experience”. Players are increasingly demanding, expecting new and never-before-seen content, which impacts games that are weak in this innovative aspect. To maintain control over the design process, human interaction will be needed to affect the PCG result by modifying the method parameters (Hendrikx et al., 2013).

PCG has shown to be an incredible and valuable tool with many advantages and some disadvantages. The benefits, where some have been mentioned above, are the following: (1) Procedural generation can be applied to almost every element in a game; (2) Automatically generates infinitely different game content; (3) Generate personalised content according to player preferences; (4) It is cheaper than human labour; (5) Uses less storage space since the content can be generated on demand; (6) Increase the human imagination boundaries.

Furthermore, its limitations are the following: (1) Human interaction is still necessary to maintain control over the design process; (2) It may generate unplayable content; (3) It may not guarantee to generate content that has both quality and diversity.

Gravina et al. (2019) proposed the integration of Quality-Diversity (QD) algorithms, which are a novel family of evolution-like algorithms that reward divergence (as novelty or surprise of the artefacts being generated) while controlling the quality of the solutions through hard limits or local competition amongst individuals with similar behavioural attributes, together with PCG to improve its outcome. QD algorithms aim to find the broadest collection of distinct and superior solutions in a single run.

Smaller teams have demonstrated the ability of PCG to produce magnificent gaming worlds, despite the misconceptions held by larger studios regarding this tool. Blatz and Korn (2017) affirm that the full potential of PCG has not yet been realized. Its significance is expanding, though, as players’ infatuation with "endless" environments and high replayability levels rises.

Although PCG may be able to work similarly to on-demand game designers, it must also be flexible, creating content as quickly as possible to meet players’ and designers’ needs (Amato, 2017). Despite this being a solid tool when a cost and time dependency exists, “without appropriate constraints and controls it is unsure if PCG creates the expected play experience” (Amato, 2017).

2.3.1 PCG in Games

The term "PCG for games" (PCG-G) refers to creating game content (levels, maps, tasks, characters, vegetation and others, excluding the game engine and NPC (Non-Player Character) behaviour) using algorithms.

Video game content has been produced using PCG-G from the late 1970s and early 1980s. The two most well-known examples of these games are Rouge and Elite, where new game content is generated every time the user goes on a brand-new journey in the game. Rogue-like is a genre that is now well-known in every home. Other famous works include Hades (2019), Slay the Spire (2017), The Binding of Isaac (2011), and Minecraft (2009). Zhang et al. (2022) claim that saving game content as seeds composed of pseudo-random number generators uses less memory than saving game content as a produced number sequence. Hendrikx et al. (2011) observed the emergence of PCG in the 1980s, which focused on interactable content, including quests, levels, and weapons (as cited in Smith, 2021). According to A. M. Smith and Mateas (2010) and Orwant (2000), research into systems that can create the rules of a game itself as well as other content generators increased in the late 1990s and early 2000s (as cited in Smith, 2021).

Smith (2021) noted that most early PCG systems were developed by practitioners who also served as developers and designers. There was little room for engagement from non-technical users in those systems because the algorithm controlled most of the information generated. A push has been made to encourage user interaction with content generators and let players shoulder part of the game's authoring responsibilities.

Most PCG systems still consider each player's experience in a single-player game, which is how PCG started in digital games as a technique to develop content for a single player. The few PCG-enabled multiplayer games, like Civilization IV, often produce content in a single instance. There are very few games with multiplayer, multi-instance PCG, which allows players to interact with original, real-time created content depending on their actions or that can affect what other players view in their version of the game (Smith, 2021).

The three most frequent aesthetic outcomes of PCG use in games, according to G. Smith (2014), are **challenge**, **discovery**, and **fellowship** (as cited in Smith, 2021). Using PCG to create personalised and unexpected surroundings is a **challenge**. PCG encourages **discovery** by offering new settings for players to discover or new systems to get familiar with gradually. Finally, **fellowship** is a style that has only recently appeared due to the creation of a structure that gradually takes shape to encourage player participation outside the game.

Video games are continuously getting more complicated as a result of their evolution. In addition, although it is already expensive, manually developing video game content costs rise. Through automatic game content generation, PCG-G has gained popularity as a solution to the problem of increasing time and resource consumption.

Games that use PCG do so in various ways and for multiple reasons, highlighting the ability to improve replayability by allowing players to create virtually endless amounts of content automatically. This allows players to return to games repeatedly and strengthen their strategies over time. Smith (2014) contends that this notion of replayability needs to be expanded upon and used as a motivation.

According to Zhang et al. (2022), uncertainty is fundamental to any game since it helps participants understand its goal. Players must select their uncertain course in unpredictable environments since a successful game must offer them effective options and consequences. Game designers have realised that, rather than merely altering the design of game levels, the application of vast and highly influential programming design can further affect the experience the game provides to players. By fusing randomness and rule-set, PCG-G technology generates various levels and types of uncertainty, significantly impacting the game's design.

Various works in this field have been produced over time, where various PCG-G approaches have been employed, each with advantages and disadvantages, categorised differently by various authors. Zhang et al. (2022) classified PCG-G approaches, according to Barriga (2019), into **traditional**, **search-based** and **machine-learning** methods (presented in Figure 2.5). Pseudo-random number generators (the first method used in video games) and other construction techniques are examples of **traditional** techniques. According to Van Der Linden et al. (2013), the key benefits of PCG-G are simplicity and effectiveness. Due to this, it has been commonly utilised to create dungeons and labyrinths (as cited in Zhang et al., 2022). **Search-based** methods can isolate the most evaluated one from the game material continuously generated and evaluated by algorithms. The space representation, the evaluation function, and the search algorithm are typically its three main components. The evaluation function evaluates and scores the generated material once the algorithm has created it. Until the score is high enough, the algorithm will continue to produce game content.When creating visuals, audio, and voices for video games, **machine-learning** techniques train models based on datasets. There are still several effective ways to produce new content using machine learning technology, according to Summerville and Mateas (2016), Jain et al. (2016), Goodfellow et al. (2014), and Snodgrass and Ontanón (2016). These ways include recurrent neural networks, autoencoders, Generative Adversarial Networks (GANs), and Markov Models (as cited in Zhang et al., 2022).

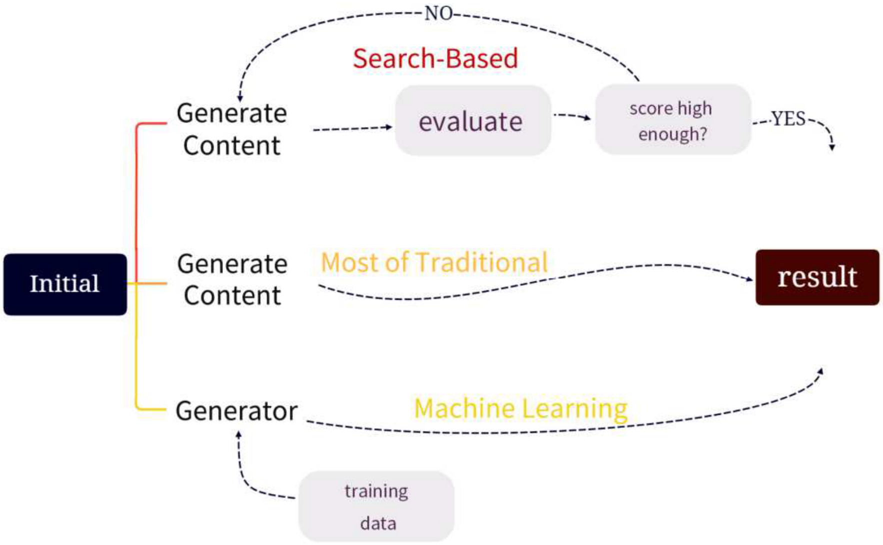


Figure 2.5: Three different PCG-G approaches by Zhang et al. (2022)

Togelius et al. (2011) suggest categorising PCG-G techniques into three groups: a **search-based** algorithm (like the one listed above), a **constructive**, and a straightforward **generate and test** approach (as cited in Moreno-Armendáriz et al., 2022). The phases in the **constructive** approach are designed to create content to warrant a level with the necessary attributes by following a set of criteria. **Generate and test** technique creates material employing a process that typically does not ensure a positive outcome, followed by an evaluation of how well the content performed by specific criteria. If the performance is not achieved, the content is created again (this procedure can be repeated as often as necessary).

Understanding the standards of how a procedural generating method is evaluated is crucial to compare various approaches. According to Yannakakis et al. (2018), a taxonomy might be used to gauge the algorithms' performance in three dimensions. The different results in multiple runs using the same method and settings serve as a proxy for the algorithm's level of **determinism** or randomness, which makes up the first dimension. The algorithm's designer's level of **controllability** over the generated output is the second dimension. The third and final dimension is the quantity of **iterability** or the number of cycles the method requires to produce material. (as cited in Moreno-Armendáriz et al., 2022).

Guzdial and Riedl (2018) presented two additional criteria, **novelty** and **playability**, to supplement the prior metrics for gauging the creativity of generated content. The variability between the reference and output levels, as determined by the algorithm's various levels, is referred to as **novelty**. **Playability** is measured regarding the value generated content (as cited in Moreno-Armendáriz et al., 2022). Table 2.2 presents a side-by-side comparison of the various PCG approaches, ranked from low to high (low → fair → moderate → high) done by Moreno-Armendáriz et al. (2022).

Table 2.2: PCG method comparison by Moreno-Armendáriz et al. (2022)

Technique	**Generation Process**	**D**	**C**	**I**	**N**	**P**
Evolutionary Algorithms	Search-Based	M	M	H	F	H
Reinforcement learning	Search-Based	M	M	H	F	H
Cellular Automata	Gen. and test	F	L	F	M	F
Grammars	Constructive	H	H	L	L	H
Machine Learning	Constructive	H	H	L	L	H
Supervised Learning	Constructive	H	H	L	L	H
Adversarial Learning	Constructive	H	H	L	L	H
Random Generators	Gen. and test	L	L	M	H	L
Rule-based	Constructive	H	H	L	L	H

Measures → D: Determinism; C: Controllability; I: Interactivity; N: Novelty; P: Playability.
Classification → L: Low; F: Fair; M: Moderate; H: High.

We can see from this table that multiple algorithms with a similar generational process react likewise in each of the metrics, and several techniques have advantages that counteract the disadvantages of the others. Hybrid strategies may successfully enhance performance and minimise flaws, according to Moreno-Armendáriz et al. (2022). Togelius et al. (2012) were the ones who first suggested combining PCG algorithms to create hybrid versions where they test the compatibility of a constructive method and an evolutionary algorithm. Algorithm hybridisation offers much potential. Nevertheless, there are still a lot of unanswered research concerns in developing an efficient, fully configurable PCG algorithm that can generate engaging content (as cited in Moreno-Armendáriz et al., 2022).

In their article, Moreno-Armendáriz et al. (2022) proposed a PCG algorithm called IORand, a fusion of two distinct strategies: a random generators strategy using semi-random content creation and a search-based strategy using DQN (deep Q-network) RL (reinforcement learning). Figure 2.6 depicts the IORand algorithm's flow.

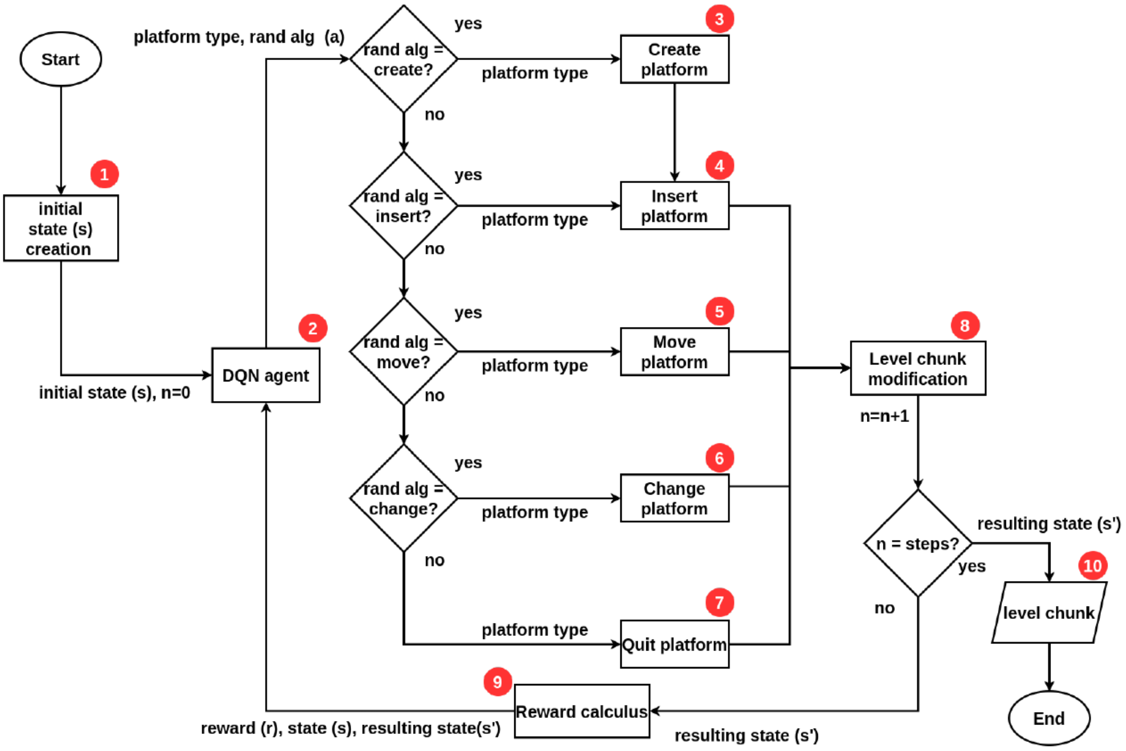


Figure 2.6: IORand flowchart illustration by Moreno-Armendáriz et al. (2022)

As shown in Figure 2.6, the Moreno-Armendáriz et al. (2022) method begins at block 1, with the creation of the initial state (s) and initialisation of the counter (n). In block 2, the agent receives the environment's current state and selects the platform type and semi-random generator that will be used to create the action (a). From this action (a), one of the blocks between 3 and 7 is executed, containing a semi-random content creation mechanism that modifies the surroundings in a way that a real designer would. After computing its environmental impact, the chosen generator is used in block 8, and the counter is incremented. When "n" reaches the value previously specified by the user, the program concludes by saving the resultant level chunk on a file (this is done in block

10). After that, an evaluation of the agent's performance is made regarding how well the gaming experience of the obtained result is. In Block 9, as a result of prior action (s'), the reward (r) along with the environment's current state (s) are the outcomes of this block, together with the prior action (s') itself. The reward and the outcome state (s') are only necessary when the training is in progress. After that, the environment's current state (s) is enough.

The goal behind this hybrid approach, according to Moreno-Armendáriz et al. (2022), is that the RL component directs the content generated through the use of an orchestrator that aims to maximise the objective function value while the random generation component provides a way to broaden the range of outcomes produced by the RL algorithm.

Lastly, to close this topic of PCG in Games, we will discuss two studies by Smith (2014) and Moreno-Armendáriz et al. (2022). Moreover, the conclusions obtained from the case studies and the studied papers will be presented. Smith (2014) provided a framework, illustrated with numerous examples from educational research projects and professional games, for discussing and analysing how PCG works in game design and PCG-based tools. This study's goals included deconstructing the notion of "replayability" and analysing how PCG is applied to influence the UX. Moreno-Armendáriz et al. (2022) created a platform game, "Pingu Run". All the previous measures used to compare the algorithms were employed to gauge how well the suggested algorithm performs: determinism, controllability, iteratively, novelty, and playability.

In the first study, Smith (2014) concluded that the framework had revealed three main processes that enhanced replayability. The first is responding to stunning environments because the game aims to play different material on each attempt to surpass a high score or go further than prior efforts. The second is developing generation methods. Despite increasing replayability, it is also a particular game dynamic that only PCG can provide. The third and final is practising in various contexts due to the content's encouragement of repeating more traditional mechanics as a means of practice in many circumstances.

Table 2.3: Performance of the IORand algorithm by Moreno-Armendáriz et al. (2022)

Technique	**Generation Process**	**D**	**C**	**I**	**N**	**P**
IORand	Hybrid	F	M	H	M	H

Measures → D: Determinism; C: Controllability; I: Interactivity; N: Novelty; P: Playability.
Classification → L: Low; F: Fair; M: Moderate; H: High.

Creating video game content automatically with a focus on providing desired gameplay experiences was a challenge that Moreno-Armendáriz et al. (2022) addressed in their second study. They did this by introducing a novel hybrid PCG algorithm, with the performance presented in Table 2.3, building agents that could reliably finish constructing gameplay-specific level chunks. Combining the evaluation technique with the hybrid PCG algorithm increased the chunks' playability. Furthermore, it was able to raise the amount of originality of RL-based PCG algorithms.

Smith (2021) concluded that developing PCG-based games will involve an in-depth study of novel algorithms and systems and creative game design. When creating PCG-based design tools, it is essential to consider techniques for creating new games that do not rely on templates of game mechanics, as well as player preferences and the game's motives for user-generated content. The terms "replayability" and "adaptability," which are now used frequently, do not adequately convey the capability of PCG-G. If only we had the AI (Artificial intelligence) systems required to create certain games and the capacity for designers to employ those systems, some games would already exist.

In their research, Zhang et al. (2022) concluded that PCG-G is mainly utilised in educational and specialised video games, such as rogue-like games, and has yet to be widely employed in creating game systems and scenarios. PCG-G algorithm should concentrate on creating game material tailored for players based on their actions in the future and make the necessary modifications by learning the different player model types and updating the newly created content while playing. The ability of intricately built deep learning and evolutionary algorithms to smoothly combine player data with in-game resources to build unique aesthetic levels, maps, and scenes based on player preferences still needs additional investigation by PCG experts.

2.3.2 PCG in Educational Games

Educational games are increasingly present to foster students' interest and motivation. However, these games typically present the same sequence of task difficulty, consequently influencing the player's motivation and interest. After some time, players get used to the game flow, getting bored and demotivated, which is caused by this fixed sequence of task difficulties. Hooshyar et al. (2018a) argue that "given the current broad diversity of player background, preferences, and motivations, it is typically difficult to achieve a dynamic difficulty adaptation with any single and fixed progression". According to Eckardt and Robra-Bissantz (2018), when designing educational games, the learning objectives are frequently at the forefront, so focusing on something other than UX may negatively impact the outcomes (as cited in Rodrigues and Brancher, 2019). The UX of playing educational games can give valuable insight into how to improve student's learning by identifying different groups of players and their characteristics.

As mentioned in the previous section, this problem can be easily solved by using a PCG solution to develop the game content through algorithmic means automatically and, if necessary, with the help of a human designer. The use of PCG provides several advantages in comparison to human-generated content: (1) it increases learning content availability, (2) it generates content considering the player's abilities/skills/knowledge, and (3) it creates content on demand. Although PCG is being investigated for educational games, more research is needed to determine which attributes impact UX from interactions with procedural content (Rodrigues and Brancher, 2019).

According to Hooshyar et al. (2018b), the player and designer's involvement in an iterative and interactive content development procedure is crucial. In addition to configuring the PCG parameters, designers also monitor player performance to develop new content that aligns with the pedagogical learning objective (PLO) or reconstruct existing content based on player performance.

In Figure 2.7, it is possible to observe the content generation framework designed by Hooshyar et al. (2018b), which begins with the designer's definition of the domain knowledge, PLO intensities, and an estimated player's skill level. These three parameters are directly related. Therefore, a set of pedagogical learning objectives is chosen after selecting the domain knowledge. On each PLO, the game will generate content according to the player's skill level taking into account that level. This is advantageous because it allows collecting data on player performance over different sessions. Thus, using this information, the designer can assess the efficiency of the learning process and begin a new round of content creation.

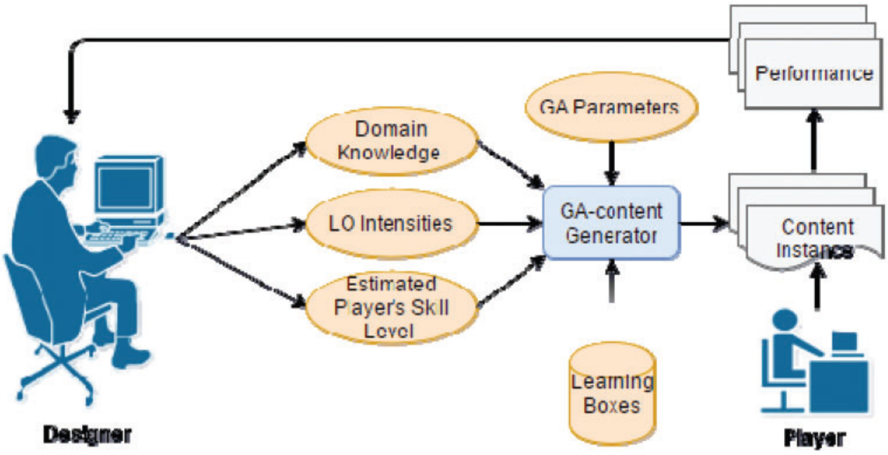


Figure 2.7: Framework of the proposed content generation by Hooshyar et al. (2018b)

The prior approach used a genetic algorithm (GA)-based heuristics search to identify the ideal learning box sequences in alignment with the designer's inputs (Hooshyar et al., 2018b). GA resolves limited and unconstrained optimisation issues based on natural selection, which propels biological evolution. It evaluates and modifies each component of a randomly formed population using a predefined fitness function. The weak ones are swapped out in every iteration, while the strong ones are kept. This algorithm completes when either of two requirements is met: a person of the specified quality is found or a predetermined time has passed. Figure 2.8 illustrates an iterative GA flowchart.

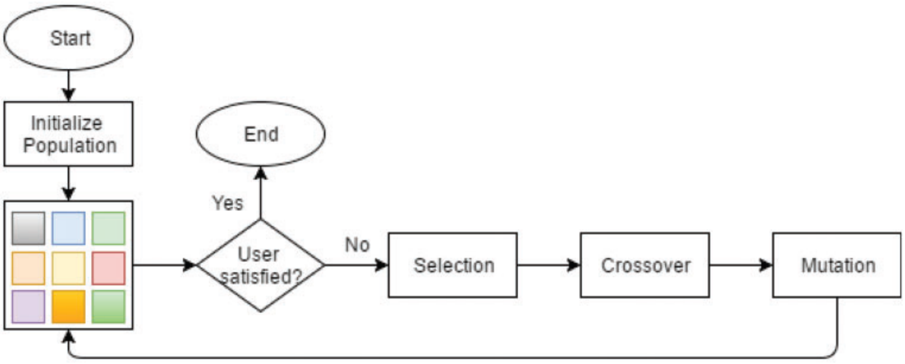


Figure 2.8: Flowchart of the genetic algorithm by Hooshyar et al. (2018b)

Evolutionary algorithm (EA)-based search is another method that can be used for PCG. The content is iterated through several evaluation functions from a population of randomly generated game-content instances. The definitions of fitness functions in EA-based PCG are challenging "because this function must both capture and evaluate a human player's experience" (Hooshyar et al., 2018a).

A data-driven method is most appropriate in EA-based PCG, considering the fitness function requirements, as the evaluation function is built using data on the real impact of a game's content (Hooshyar et al., 2018a). Because it is not dependent on a designer to modify the content following users' specific capabilities, it can more easily adapt to new forms of game content.

Figure 2.9, Hooshyar et al. (2018a) provide an example of data-driven content development. This approach is composed of three modules:

- **Content Design:** helps the designer when creating content. The resulting content consists of shared content, learning objectives, materials, and instances specific to a domain.
- **Data Training:** a game with data-driven language learning is employed to train SVMs.
- **Content Generation:** To evaluate the fitness of the content, the trained SVM is applied to the GA-based content generator. This content generating module may create content that satisfies the learning target based on the student's capabilities and learning objective.

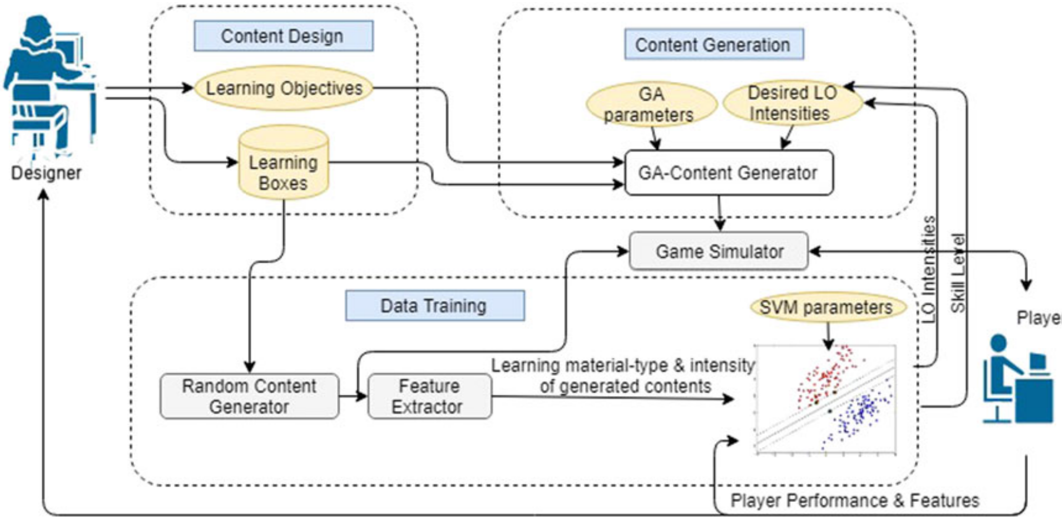


Figure 2.9: Framework of the proposed data-driven content-generation approach by Hooshyar et al. (2018a)

Lastly, to close this topic of PCG in Educational Games, we will discuss three studies, two done by Hooshyar et al. (2018b) and Hooshyar et al. (2018a), where both used their framework to develop early English reading skills of young children with an academic language learning game and one done by Rodrigues and Brancher (2019). Hooshyar et al. (2018a) also compared their solution to a heuristic-based approach.

Rodrigues and Brancher (2019) conducted an experiment in which 256 participants performed it in their classes to determine the factors influencing players' curiosity. A basic math practice game called SpaceMath, which uses PCG to generate the game content, was introduced first, as well as the investigation itself. Unlike most educational games, the objective of the game is to motivate students to do these problems as frequently as possible rather than to instruct them on how to perform basic arithmetic operations (an approach, according to Ke (2008), based on repeated learning where players must continually complete an educational activity). Participants then played the game through exactly 20 levels to ensure that every interaction was similar. In this experiment, two attribute categories were used: in-game measurements and personal data inserted during the game's registration process (all attributes of both types can be seen in Table 2.4). To measure players' curiosity, a modified version of the Wouters et al. (2011) questionnaire, which consists of seven items to be responded to on a Likert scale, was employed at the end of the experiment.

Table 2.4: All measured attributes in the experiment by Rodrigues and Brancher (2019)

Attribute	Definition
Participant's Age	Current age of the participant.
Participant's Genre	Regardless of the participant's gender.
Has Internet Connection	If the participant's home computer has internet connection.
Participant's School Type	The participant's type of school (private, public, or other).
Participant's School Stage	Whether they are in elementary, medium, or a higher stage.
Time Spent Playing	Average weekly playing time for each participant in hours.
Is a Gamer	Regardless of whether the individual identifies as a gamer.
Enjoys Math	On a Likert scale of 1 to 5, how much do you enjoy math?
Understands Math	Self-assessed arithmetic proficiency on a Likert scale (1 to 5).
Average Score	Per-level average score.
Maximum Score	Total of the highest score reached.
Average Shots	Shots fired on average per level.
Average Time	Each level's average time in seconds to be completed.
Total Time	Seconds used to play the game in total.
Highest Win Sequence	The longest run of victories.
Win Rate	Percentage of victories for each played level.

In the first study, Hooshyar et al. (2018b) confirm that when game contents are tailored to students' ability, as opposed to uncustomized games, users' performance greatly improves. According to the results, participants with lower proficiency improved their performance more than those with higher proficiency while playing the customised game.

In the second study, Hooshyar et al. (2018a) reached two conclusions. Firstly, users realised higher performance gains when they played game content customised to their abilities than when they played content that was not tailored. Secondly, data-driven approaches generate content that exactly matches a specific player performance target more effectively than heuristic approaches.

In the third study, Rodrigues and Brancher (2019) deduced from the data that School Stage, Enjoys Math, Understands Math, Age, Average Score, and Win Rate had a small to moderately significant correlation, Average Time, Total Time and Average Shots had a small significant correlation, and being a Gamer had a significant correlation to player's curiosity. All the remaining characteristics, however, had negligible correlations ("School Type", "Time Spent Playing", "Highest Win Sequence", "Maximum Score", "Genre", and "Has Internet Connection"). These techniques allow practitioners to take advantage of these worries and pique players' interest. Real-time adaption systems can be made by designers and developers using these findings. Based on their distinct characteristics, they can use them to modify games and develop player models that would pique students' interest. According to Santos and Isotani (2018), this is essential in overcoming the difficulty of offering personalised educational games.

2.3.3 Object Placement

The inside of every structure, complete with rooms and furnishings, would take too long for designers to manually recreate in an open-world game. However, the interiors of these structures might have been produced automatically using contemporary PCG techniques. This would allow for new gaming experiences while increasing the game's perceived realism. In the realm of games, there should not be much of a distinction between hand-designed and PCG graphics in terms of quality. Compared to material that has been manually developed, current procedural content might appear monotonous and repetitious. Building interiors are a good option for automated approaches, thanks to a number of their properties. For instance, a typical house's interiors frequently have a similar layout. In kitchens, counters and cabinets are frequently placed close to the wall, with either a table or an extended countertop as the island in the middle of the room.

These findings might be turned into rules and processes that automatically create these interior spaces. A layout solution based on rules provided by Tutenel et al. (2009) is particularly well-suited for procedural approaches. Items are fed into the solver according to a plan or method, and it makes an effort to suit them by a set of guidelines unique to those goods and the ones currently in use. They also specify object characteristics to direct the layout or connect them to other objects. Patterns may be seen in practically every space that serves a particular purpose. Numerous of these patterns also have a hierarchical structure.

Hahn et al. (2006) demonstrate a significant benefit of procedurally producing building interiors. Only the rooms that could be generated from the current vantage point were the subject of their investigation. This is a practical method for managing enormous buildings with various rooms. All changes are recorded and tracked to maintain modifications made to the world. The saved modifications are once more applied to the newly formed room when it is briefly erased from memory before being restored (as cited in Tutenel et al., 2009).

The primary goal of the technique proposed by Tutenel et al. (2009) was to determine, given a beginning layout, the potential positions of a new item using a set of criteria for both the new item and the layout's existing items. There are two methods to describe the relationships between items: explicitly (for instance, specifying that the couch must be facing the television as well as the distance between them cannot be greater than 5 meters) and implicitly (using features). Using hierarchical blocks to solve problems is a crucial component of their strategy. These items are integrated and handled as one block when placed together, such as a table with seats surrounding it and a few plates on top. Unwanted overlaps are removed from the detected sites based on the characteristics. They carry out this process for a limited range of angles to accommodate various item orientations. To determine the undesired portions, their system applies the Minkowski addition. When a new item's feature should not conflict with a feature that has previously been introduced, the collection of potential locations is reduced, taking out the Minkowski addition of the old and new features.

According to Tutenel et al. (2009), it is evident from the technique above that the number of features included in the design may result in a significant performance bottleneck. The time it takes to prune potential item positions based on layout characteristics increases as more items and all their features are added to the layout. The backtracking rule is essential in their planner. Backtracking and selecting a new placement for previously put items might help when a stage of the layout process has been reached when a vital component can no longer be placed. These items could have blocked the deployment of the new item at their previous location. Using the approach of Tutenel et al. (2009). Figure 2.10 illustrates an example of a lounge.

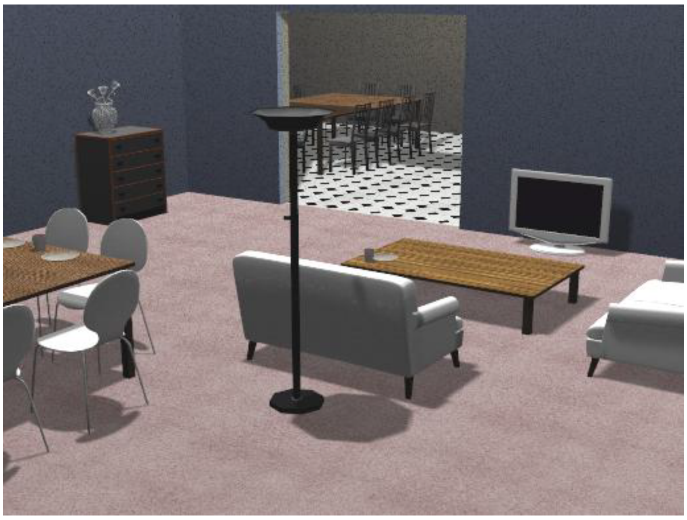


Figure 2.10: Illustration of a lounge design by Tutenel et al. (2009)

A technique presented by Howard and Broughton (2007) adds significant pieces of furniture by hand, while a genetic algorithm adds less significant ones (as cited in Taylor and Parberry, 2011). Five essential factors for any PCG system are listed by Doran and Parberry (2010): controllability, interest, structure, novelty, and speed (as cited in Taylor and Parberry, 2011). The procedural clutter (non-architectural room contents) generator presented by Taylor and Parberry (2011) employs a hierarchical Petri net with inhibitor edges to maximise controllability without sacrificing the other four requirements.

According to Taylor and Parberry (2011), a cluttered space typically has a tangle of items that seem somewhat random. The layout of objects usually nearly follows specific patterns. They can express this understanding with an anchor point, which designates an approximate location for an item. The anchors in a 2D area would be positioned at the extremities, dispersed across the edges, and organised inside a matrix in the centre of the space. The matrix size and edge spacing vary depending on the type of room. The designer can set these initial anchor points either computationally or manually. Objects with multiple anchor points are arranged using their clutter generator. A table, for instance, could have anchor points for seats, as well as points for the dinnerware on top, a centrepiece, and other clutter. The items being put have a Gaussian displacement about their normal surface, as determined by the standard deviations, to prevent everything from sitting perfectly in alignment with everything else.

The chances of things colliding when put randomly are high. According to Taylor and Parberry (2011), when two objects collide, they produce new random displacements for each object's location and orientation, repeating the process if required several times. It is frequently safe to toss the item away if this does not work. However, it is possible that certain crucial game-related objects will not generate. In this instance, they recommend that a different random room arrangement be created again after a limited number of trials. If it does not work, it is best to rethink the restrictions. Utilising the methodology of Taylor and Parberry (2011). Figure 2.11 depicts a few created rooms.

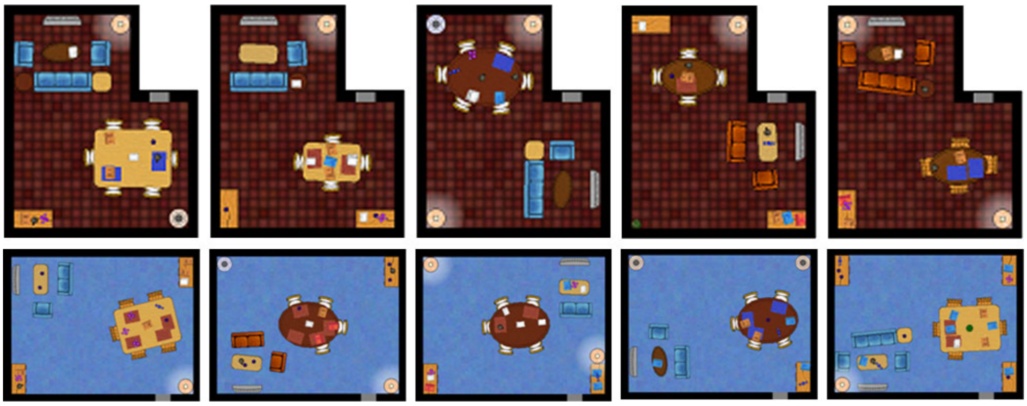


Figure 2.11: An example of 10 rooms created with the same object set and Petri net by Taylor and Parberry (2011)

Atmaja and Sugiarto (2020) investigated PCG's use to produce placement of items representing the educational game's learning material. The game's internal mechanics may be moderately altered by assigning components of various learning materials to game objects. The primary driver of medium coupling's ability to reduce costs is how easily learning materials can be replaced. The game creator must guarantee only the proper placement of the game items. The arrangement defines the game's level of difficulty and enjoyment, and poor performance may even render the game unplayable. Theoretically, the game creator may use PCG to automate the layout process to save resources.

Because the generation process will strive to arrange the items unpredictably while necessitating player exploration to produce placements of things in a game map, Atmaja and Sugiarto (2020) used a three-stage PCG in their study: (1) The first step produced the fewest possible CAEOs (correct answer element objects); (2) The minimum CAEO set was duplicated in the second step; (3) The final step covers the remaining area on the map with WAEOs (wrong answer element objects).

To guarantee the CAEOs' proper dispersion and prevent them from concentrating in one place, the first CAEOs were generated first, and then the subsequent CAEOs. They defined two conditions for their algorithm related to player exploration and unpredictability. The first was that the player should, on average, travel a significant distance to gather answer element objects. The objects' positioning should not only favour certain map sections, according to the second requirement.

The evolutionary search method presented by Togelius et al. (2016) was the foundation for each generation process stage, according to Atmaja and Sugiarto (2020), with certain modifications. To start, they kept the population the same. Second, a modified element replaced its original self in the population rather than producing progeny. Each stage's evolutionary algorithm ran for 100 iterations using 40 game maps as its population, each containing several pertinent items. Using the method of Atmaja and Sugiarto (2020), an illustration of object placements on a map is shown in Figure 2.12 (initial and subsequent CAEOs are represented by blue and black numerals, respectively, whereas WAEOs are represented by red squares).

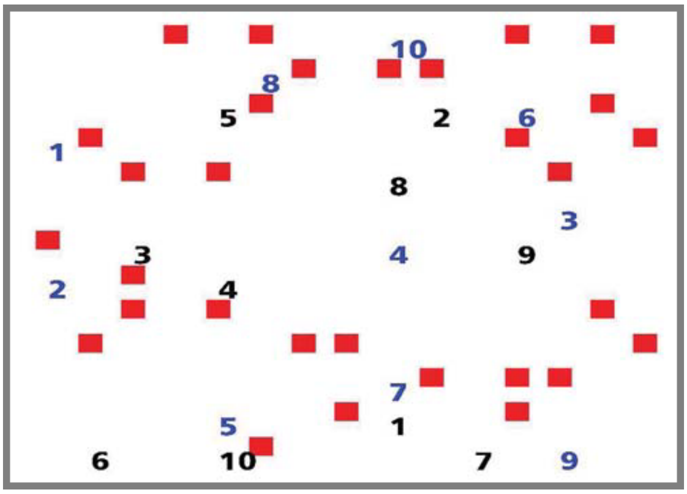


Figure 2.12: An illustration of answer element objects distribution in a map by Atmaja and Sugiarto (2020)

Lastly, to close this topic of Object Placement, we will discuss one experiment by Atmaja and Sugiarto (2020). Moreover, the conclusions obtained from the experiment and the studied papers will be presented. In their study, Atmaja and Sugiarto (2020) investigated PCG, a cost-reduction strategy for pedagogical games with medium coupling. They have applied this strategy to place objects that reflect the right and wrong responses to particular learning content. The PCG approach was used in their experiment to produce and randomly deploy CAEOs and WAEOs on game maps that had position markers scattered across them.

According to Tutenel et al. (2009), their method can produce room interiors quickly enough to be useful at runtime. In this manner, a sizable metropolitan gaming world might be expanded to include the capability of accessing buildings without requiring designers to spend weeks or months meticulously drawing out each architecture. However, designers retain influence over the creation process by producing blueprints that specify which pieces make up the arrangement and by the guidelines that must be followed while placing these items. They conclude that their method may be applied to automatic and user-assisted design and can address many layout issues frequently encountered while building gaming worlds.

According to Taylor and Parberry (2011), their generator creates beautiful clutter for spaces and is adaptable enough for a designer to manage the output to make the right clutter for various rooms. Despite the random nature of the room's contents, they argue that its material is fresh because of how it is organised. Their method is as quick as Tutenel et al. (2009)'s, and Petri nets' Turing completeness gives improved controllability. The approach Howard and Broughton (2007) uses fails in both areas and is tailored towards a subset of clutter production.

Atmaja and Sugiarto (2020) contend that their algorithm was general, stochastic and automatic, followed the generate-and-test approach based on the PCG taxonomy in Togelius et al. (2016), and generated the necessary content for the game. Their investigation of the object distribution patterns of the algorithm produced insightful results. One multiplier value yields the optimal object dispersion for any specific amount of created objects. The multiplier is independent of other factors on maps with various position marker numbers and spacing between markers. The generation technique may produce and distribute items uniformly around the game map while guaranteeing the distribution's unpredictability by using a specified multiplier for the fitness function's standard deviation. The amount of depicted items on the map, as opposed to the map itself, determines the appropriate multiplier value. Using curve-fitting, they calculated the function that converts produced object numbers to the applicable multipliers. They discovered that combining various multipliers for the three phases helps create complete CAEO and WAEO sets with good distributions. Additionally, the generating procedure guarantees the player will have long enough and consistent average traversal distances to acquire CAEOs.

2.4 PCG in IEERS

Building an IEER game requires designers' attention. They need to consider the rules and environment when developing an educational game of this kind to achieve a good end product. With the arrival of PCG, a big part of the game designers' work was facilitated. Houdini and Unity3D are two software tools IEER developers can use to create interactive and realistic games with high-quality graphics.

In the previous academic year, the "EdScape" team, composed of a multidisciplinary team, developed multiple dissertations to build an ER-like game using new technologies (such as PCG) for educational purposes. Within these dissertations are included Zheng (2022), Santos (2022), Pinto (2022) and Garcia (2022), which we will be further introduced together with the obtained conclusions.

In her dissertation, Zheng (2022) included three categories of minigames (puzzles, closed-ended questions and answers, and matching pictures and labels), a web-based system of rewards that allows students to view their medals and a database service that records all pertinent game-related occurrences. Basic minigame and survey structures were incorporated into the game's framework (created using Unity3D), and teachers may change their contents to suit their preferences. Finally, a study was used to validate the framework.

Positive findings from the study conducted by Zheng (2022) showed that the created game assisted students in learning about the topic covered. Given that an ER game must have obstacles to overcome, this dissertation helped the "EdScape" project. Another addition was the attractiveness of the medal awarding system, which increased the project's potential and may serve as an external encouragement for the students to play and learn about the subject matter simultaneously.

The dissertation goal of Santos (2022) was to provide the groundwork for a system that might produce pedagogical ERs. This framework was designed with inexperienced teachers and the development of adaptable games in mind. When necessary, teachers can develop and update the games using this framework. The ER games created using this framework are meant to promote the learning objectives incorporated by the teacher while offering an enjoyable and engaging gameplay experience. Two case studies were designed to evaluate various concerns, such as game levels, multi-room, and players' perspectives, in addition to the more detailed study on engineering ethics.

Santos (2022) claims that 16 university students successfully implemented and tested the two study games. The participants' responses were analysed, and it became clear that the game was beneficial in fostering learning. This framework may aid in creating other educational adventure-style games and prove to be an invaluable resource, given that using this kind of game in the classroom has been linked to positive outcomes for students learning.

During his study, Pinto (2022) created a tool for creating an IEER utilising PCG. This programme employed crucial resources to streamline the complex game development process. A prototype of the PCG was created in Houdini for testing reasons, and it was then integrated into a Unity3D IEER game. To create new levels for various reasons, this tool functions as a manager that changes the level, layout, content, non-architectural features, and overall look. One of the main goals was to create an adaptive IEER with a continuously shifting appearance, layout, and pathing that could procedurally adapt to the user's needs.

The PCG technique created for the IEER game, according to Pinto (2022), demonstrated effectiveness in its application as a complement to this pedagogical tool and proved to be a quality that is beneficial for both teachers and students because of its adjustability, versatility, and application in playful learning. The capacity of PCG approaches to mould themselves and offer a unique versatility that no comparable game feature can be verified to adapt to a gaming need. Regarding PCG usability, the created low-fidelity prototype was positively appreciated. Most participants expressed the emotion of intrigue, followed by interest when it was presented that the recognised PCG utility emerged as a critical element that improves game replayability and would undoubtedly shape future game development. In their opinion, Houdini's PCG tool functions effectively as a helpful tool that may be used in video games. The created tool has demonstrated to be a valuable asset considering its adjustability and utility, although it is still in its initial stages of expansion.

Garcia (2022) states that her research aimed to learn more about the impact of world design on immersion and how to manage challenges with epilepsy, colour blindness and photosensitivity in educational games that need flexibility. Two IEER prototypes were created: (1) 2D with an immersive story; and (2) 3D with neither narrative nor feedback. Both were tested, and the information gathered was analysed to understand how IEER should be developed.

Garcia (2022) concluded that the tools utilised to create the chromatic palettes for every prototype were effective and allowed for the application of colour theory, which enhanced user experience. It was discovered that they were not enough to ensure the prototypes were safe and usable. Participants favoured prototype 1 over prototype 2 in practically all criteria because it was more educational, intuitive, and immersive. The findings showed that an immersive story would make prototype 2 more immersive. Only the adaptability measure yielded results that were not conclusive, highlighting the need for more study. Despite the ER not being in 3D, she discovered that most participants favoured the prototype with an immersive narrative, which led her to conclude that immersive narratives encourage users to feel more connected to the IEER and enable a more profound learning experience. All participants, independently of the prototype, thought that IEERs had educational benefits, which allowed her to conclude that there is interest and significance in further researching how IEERs might be utilised in an academic context.

According to Sousa et al. (2022), researchers may experience difficulties adopting PCG in an EER due to concerns about flexibility based on students' demands in subsequent studies. However, EER should employ PCG techniques when working with certain parts that can be classified by complexity utilizing a six-layer taxonomy (bits, space, systems, scenarios, design, and derived content).

PCG in IEERs is a little-studied topic. Still, from what has been mentioned in this and the previous sections, it will undoubtedly positively impact the student's learning process because it is a set of aspects that have been shown to increase learning.

2.5 Summary

Students learning processes have been improved by introducing games and new technologies into educational settings. As we saw, all the previously spoken topics can enhance learning in very different ways awakening the student's interest and making learning more pleasant and attractive.

By analysing what was found in different documents and what was stated by their authors, we can conclude that the application of games in education is effective, increasing students' motivation and willingness to learn and arousing the feeling of enjoyment, preferring this learning method over traditional teaching means. Teachers' engagement and pedagogical decisions are crucial in ensuring student satisfaction in a learning environment, so their interests and motivations must be awakened. Game design and research must properly integrate cognitive, affective, and social approaches to understand what learning opportunities games present properly.

It was also found that relevant gaps exist in immersive learning environments, which will contribute to the need for future research in this area. Using VR to deliver new experiences strengthens the subject's memory, accelerating their learning process. However, when poorly implemented, VR can cause some inconveniences for the users, such as motion sickness, cybersickness and simulator sickness.

The use of PCG as a game content generator has several advantages, highlighting its capacity to generate fresh content in each run. However, it also presents some disadvantages, requiring human intervention to monitor its behaviour. The integration of quality-diversity algorithms together with PCG can improve the content generated from it. Challenge, discovery, and fellowship are the three aesthetic effects of PCG use in games that occur most frequently. Over time, several works have been created using various PCG-G methodologies. Each of these methodologies has pros and cons categorised differently by multiple writers. The three key activities that improve replayability include responding to gorgeous environments, improving generation methods, and practising in different contexts. A hybrid PCG algorithm can provide better game material than a standard one. A thorough examination of novel algorithms, systems, and game design will be required for creating PCG-based games. PCG-G is yet to be widely used in developing game systems and scenarios, primarily in pedagogical and specialised video games, such as rogue-like games.

PCG has shown to be a powerful tool for generating custom game content considering student preferences and abilities, significantly improving student performance. By identifying various player groups and their characteristics, the UX of educational games may provide crucial information on enhancing students' learning. Based on these traits, designers can alter games and create player models that grab students' attention.

PCG can also be used to place objects in a game scene. Based on what was analysed, various PCG algorithms were presented that quickly produce room interiors without requiring designers to spend much time. So, PCG cannot only generate game worlds but also place objects automatically for us, showing once again to be a powerful tool for game development.

According to what has been discussed in this chapter's previous sections, PCG is a relatively under-explored topic in IEERs. However, it can substantially impact the student's learning process since it combines several aspects to improve learning.

Chapter 3

Methodology

This chapter discusses the research methodology adopted, the work plan that will be followed during the dissertation development and how the project was managed.

3.1 Research Methodology

The methodology adopted was the Design-Based Research (DBR), beginning with the problem's analysis, solutions development, and an iterative cycle of testing and refining solutions. This requires constant interaction with the research subjects since their feedback is crucial in developing an excellent final product.

In educational environments, DBR has emerged as a standard and pertinent technique for designing iteratively to acquire knowledge for quality procedures. The capability of DBR is recognised to forecast outcomes, focusing on creating suitable interventions that can influence the context of the study. Numerous writers have noted this approach's potential to help generate research procedures with significant practical relevance, according to Tinoca et al. (2022). Due to its iterative nature, researchers can set up their study into iterative research cycles to test new resources and pedagogical techniques while improving goods and processes.

As a result of its focus on bridging the gap between practice and theory, frequently cited as a weakness in educational research, DBR was first developed in architecture and engineering and then initially applied to an educational context during the 1990s. This application underwent a more substantial expansion at the start of the new century. According to The Design-Based Research Collective (2003), DBR examines the complexity of education, seeking a better comprehension of the educational processes associated with learning through a systematic design, a variety of techniques, and research of educational techniques and technologies (as cited in Tinoca et al., 2022). The use of DBR in educational contexts has been lauded by Anderson and Shattuck (2012) due to its capacity to involve fieldwork participants actively and frequently perceived as research subjects (as cited in Tinoca et al., 2022).

Tinoca et al. (2022) state that a DBR approach often calls for quantitative and qualitative techniques, combining empirical research with theory-based designs through repeated design cycles, implementation, analysis, and revision. The DBR data offer evidence to comprehend when, why, and how the educational intervention grows, enhancing results and encouraging data-driven innovations in close cooperation among participants, designers, and researchers.

Research subjects are vital within DBR. They are seen by Armstrong et al. (2022) as “key contributors and collaborators in the research process". In other words, the research subjects, and the researcher, make the research go forward.

Through this collaborative and iterative research process, DBR produces two types of outcomes. There are tangible outcomes resulting from meaningful interventions and practices and intangible outcomes that arise from advancing theory and understanding of how people learn (Armstrong et al., 2022).

According to Reeves et al. (2005), the DBR paradigm of inquiry is pertinent to developing educational technology. Anderson and Shattuck (2012) provided the following summary of this strategy (as cited in Adams and Preez, 2022): (1) Placed in a genuine educational setting; (2) Concentrates on developing and evaluating a meaningful intervention; (3) Adopts various approaches to offer a better direction for improving education; (4) Involves several revisions to arrive at the ideal intervention design; (5) Encourages interaction between scientists and practitioners; (6) Helps scholars and practitioners work together.

In conclusion, DBR recognises the impact of environmental elements on the implementation and results of an educational intervention.

DBR is an iterative process that involves researchers refining and reworking the intervention over time using a variety of approaches. This flexibility puts the result ahead of the process. McKenny and Reeves (2012) emphasised that there are three core steps of DBR to achieve the “Project End”, as we can see in Figure 3.1, which are: “Analyses and Exploration”, “Designs and Construction” and “Evaluation and Reflection” (as cited in Armstrong et al., 2022; Adams and Preez, 2022).

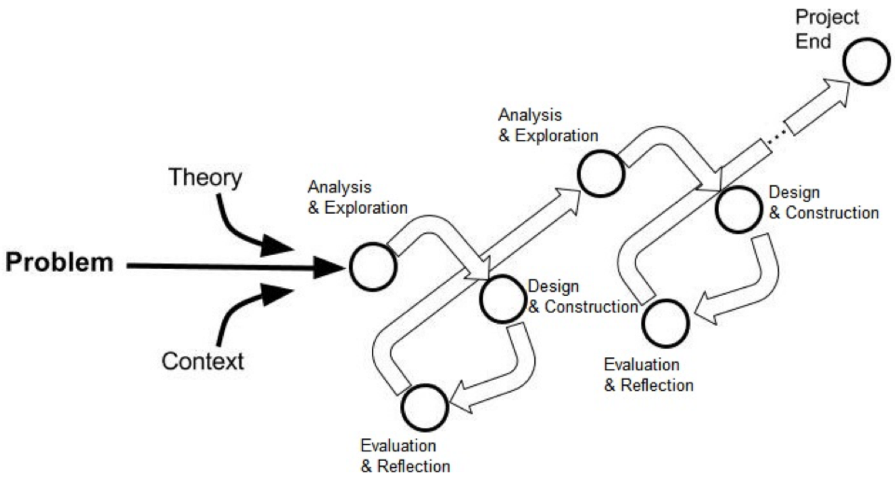


Figure 3.1: The iterative process of design-based research by Armstrong et al. (2022)

These three core steps of DBR can be described as follows:

- **Analysis and Exploration:** DBR begins with analysis and exploration of theory and research context, which should be conducted consistently throughout the process. Identifying and understanding the problem to be addressed is crucial at the beginning.
- **Design and Construction:** following analysis and exploration of the problem, interventions are designed and constructed, including technologies or more abstract elements such as activity structures, institutions, scaffoldings, and curriculums.
- **Evaluation and Reflection:** constant and careful evaluation of each iteration is crucial to make improvements. Reflection helps researchers connect actions with results. To benefit theory and practice, researchers must analyse the changes that resulted in success or failure (Armstrong et al., 2022; Adams and Preez, 2022).

Easterday et al. (2014)'s method is another option for McKenney and Reeves' (2012) strategy. The six iterative steps make up one cycle of the DBR process (which can be observed in Figure 3.2): (Focus) concentrate on the issue, (Understand) comprehend it, (Define) specify design objectives, (Conceive) come up with a rough sketch of the solution, (Build) create it, (Test) and test it (as cited in Lyons et al., 2021).

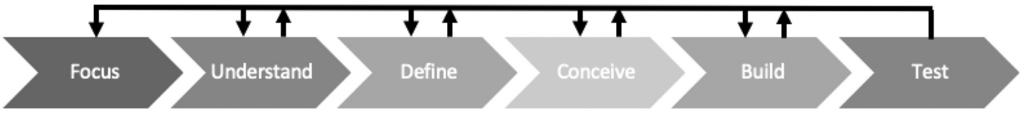


Figure 3.2: DBR's six iterative stages based on Easterday et al. (2014) by Lyons et al. (2021)

According to Lyons et al. (2021), the early stages of the problem-solving process should involve completing a needs analysis, qualitative investigations, literature reviews, and developing local theories. Design objectives and specified design principles must be created after being thoroughly aware of the present state of knowledge and practice about the issue. McKenney and Reeves (2019) define design principles as prescriptive theoretical guidelines that govern the creation of a solution that is founded on problem-solving theories that are descriptive, predictive, and explanatory (as cited in Lyons et al., 2021). A conceptual solution to achieve the design objectives is created during the conception phase and takes the shape of new design propositions and existing design principles. According to McKenney and Reeves (2019), design propositions are design concepts that must be verified and improved via intervention testing and reflection (as cited in Lyons et al., 2021). The test phase, which looks at the intended solution's learning procedures and results, starts after the solution has been built. Lastly, McKenney and Reeves (2019) state that reflecting on the test phase outcomes is necessary to change design concepts, design features, and implementation procedures (as cited in Lyons et al., 2021). Then, all of these processes can be repeated during the available cycles.

3.2 Work Plan

The work plan, developed over 20 weeks, was as follows: [Task 0] State of the art, research and exploration of background, related work, existing technologies, and approaches to the same or related problems regarding the main topics of this dissertation, such as IEERs, VR and PCG (previously developed in Dissertation Preparation subject (PDIS)); [Task 1] Selection of an ontology, based on JSON, to specify learning paths to be transformed in an IEER. This ontology is able to accept different challenges that the teacher wants to make to their students, for example, multiple choice and true or false questions and external links to challenges available on the web. It is also possible to specify the name of the learning path (working like an id to distinguish from the others), its difficulty level and thematic (1 week); [Task 2] Development of an IEER framework in Unity3D to collect the information of the learning path, specified using the previously selected ontology, and generate the IEER according to what was read. This framework connects the file(s) containing the learning path information and the Houdini framework, creating the IEER (4 weeks); [Task 3] Exploration of Houdini for procedural room generation. Start by doing some tutorials to develop the knowledge of the Houdini tool and how it can procedurally generate content with it, explicitly generating IEER rooms on-demand with predefined parameters (1 week); [Task 4] Developing the IEER procedural generation prototype based on the selected ontology. After choosing the ontology and developing the IEER framework in Unity3D, a prototype must be created to test further and evaluate the achieved solution. The framework was created and used examples of learning paths to generate IEERs (4 weeks); [Task 5] Development of a backoffice, in the Unity3D environment, for a manual definition of learning paths to give the teacher a more practical and intuitive solution when building their learning paths. We chose Unity3D for the backoffice development because the IEER game will also be implemented in Unity3D (fewer technologies will be easier to adopt and more practical) (4 weeks); [Task 6] Testing and validating the developed solution, obtaining feedback, and improving it. In this task, the created prototypes are tested and validated with a questionary which will be carefully analysed to extract all the relevant information that could improve our solution (2 weeks); [Task 7] Writing of the dissertation. After the investigation, selection of an ontology, development of the IEER framework and backoffice, and test and validation of the developed prototypes, it's time to include all information gathered inside this document (4 weeks) (see Figure 3.3).

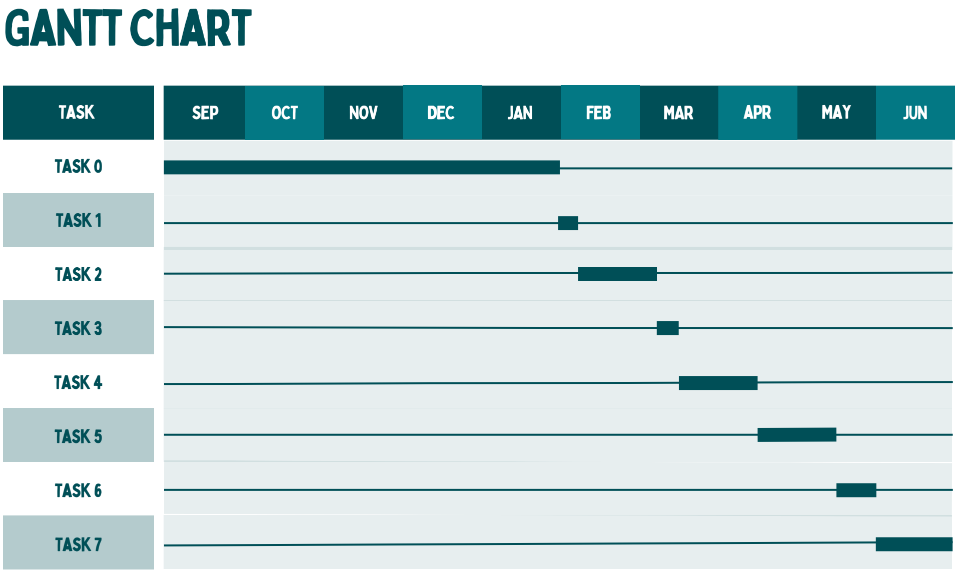
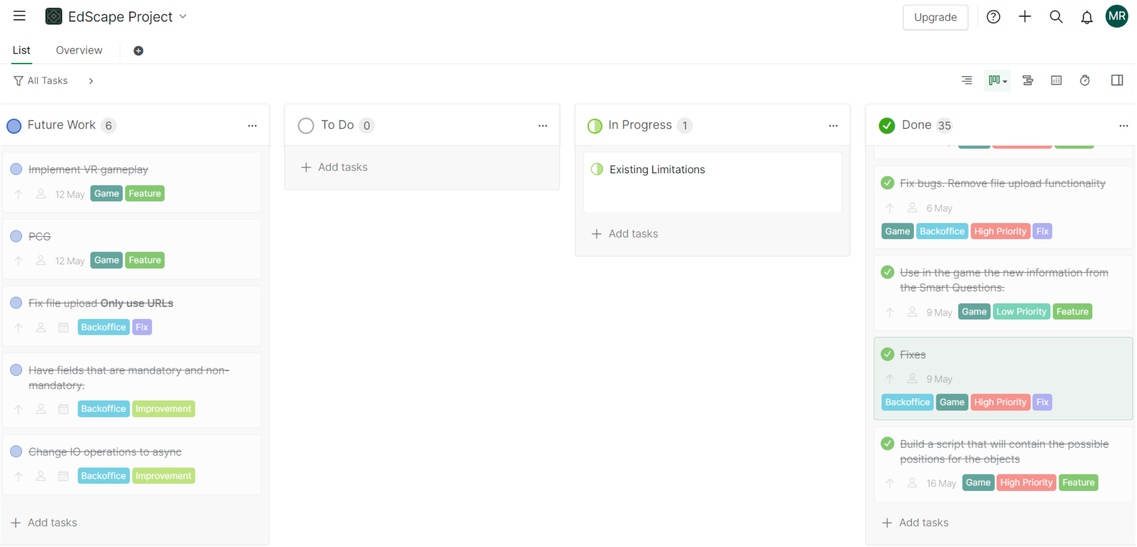


Figure 3.3: Gantt chart

3.3 Project Management

Various tasks were planned, scheduled, and performed throughout the project development to reach the desired final product. To help us with this task management, a web application called Quire was used. Quire has three initial lists ("To Do", "In Progress", and "Done") where we can create new tasks and, according to their status, move them from one list to another. An additional list, called "Future Work", was added and used to put all the planned tasks that were impossible to accomplish due to lack of time. Figure 3.4 shows the Quire board with all the previously mentioned lists and some assigned tasks. Between these tasks, there is still one in the "In Progress" list because it is used to note all the existing limitations that have been found.



Future Work 6

~~Implement VR gameplay~~
12 May Game Feature

~~PCG~~
12 May Game Feature

~~Fix file upload **Only use URLs**~~
Backoffice Fix

~~Have fields that are mandatory and non-mandatory.~~
Backoffice Improvement

~~Change IO operations to async~~
Backoffice Improvement

To Do 0

In Progress 1

Existing Limitations

Done 35

~~Fix bugs. Remove file upload functionality~~
6 May
Game Backoffice High Priority Fix

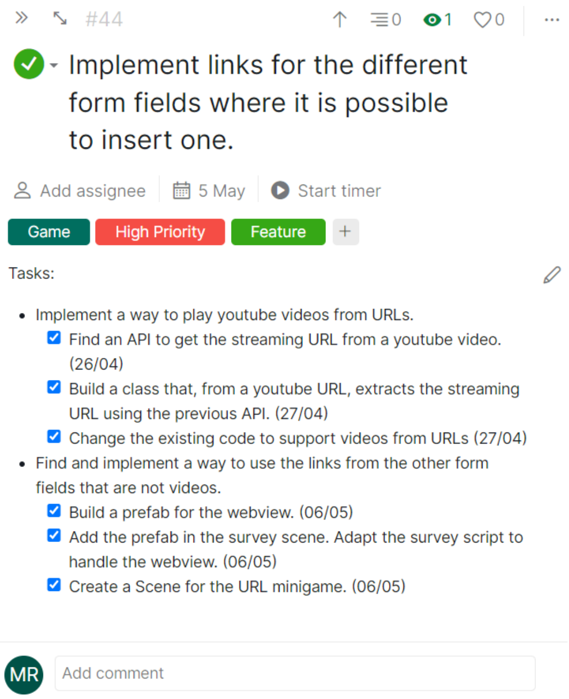
~~Use in the game the new information from the Smart Questions.~~
9 May Game Low Priority Feature

~~Fixes~~
9 May
Backoffice Game High Priority Fix

~~Build a script that will contain the possible positions for the objects~~
16 May Game High Priority Feature

Figure 3.4: Task board in Quire

You probably wonder why so few tasks are on the "Done" list. This happens because most tasks are "Epic", meaning they can be and are divided into subtasks. By accessing their detailed information (see Figure 3.5), it is possible to see that a group of checkboxes exist for each "Epic". Each checkbox therein represents a subtask to perform, and at the end, it is possible to find a date indicating its completion. Tasks can be easily created in one of the available lists, where only the task title is mandatory. Still, it is also possible to add a more detailed description, set the expected end date, assign collaborators (in the case of more than one collaborator in the same project) and add tags. Figure 3.5 shows how the task's detailed information is displayed.



#44 ≡0 1 ♡0 ···

✓ Implement links for the different form fields where it is possible to insert one.

Add assignee 5 May Start timer

Game High Priority Feature +

Tasks:

- Implement a way to play youtube videos from URLs.
  - ☑ Find an API to get the streaming URL from a youtube video. (26/04)
  - ☑ Build a class that, from a youtube URL, extracts the streaming URL using the previous API. (27/04)
  - ☑ Change the existing code to support videos from URLs (27/04)
- Find and implement a way to use the links from the other form fields that are not videos.
  - ☑ Build a prefab for the webview. (06/05)
  - ☑ Add the prefab in the survey scene. Adapt the survey script to handle the webview. (06/05)
  - ☑ Create a Scene for the URL minigame. (06/05)

MR Add comment

Figure 3.5: Detailed information of the task

Tags can be added by clicking the add tag button, writing the tag name, and pressing “Enter”. Figure 3.6 shows how the tags system works. In total, eight tags were created and can be divided into three groups as follows:

1. **Subject:** (a) Game, (b) Backoffice.
2. **Priority:** (a) High Priority, (b) Medium Priority, (c) Low Priority.
3. **Type:** (a) Feature, (b) Improvement, (c) Fix.

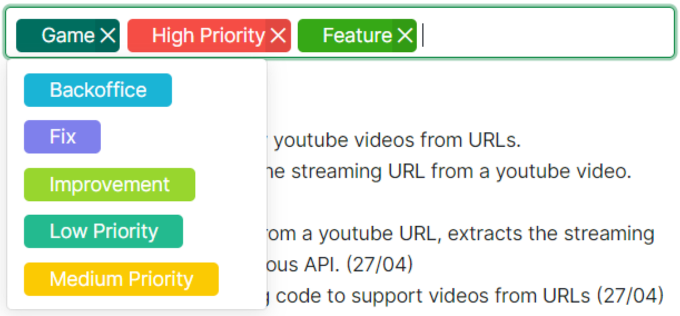

Game ✕
High Priority ✕
Feature ✕
Backoffice
Fix
Improvement
Low Priority
Medium Priority
youtube videos from URLs.
ne streaming URL from a youtube video.
rom a youtube URL, extracts the streaming
ous API. (27/04)
code to support videos from URLs (27/04)

Figure 3.6: Tag system in Quire

The order that appears in each task is the same as presented above. First, we have the “Subject”, where we can choose one of the values or have both simultaneously (in the case of bug fixes). Second, we have the “Priority” that can only assume one of the three available values. And finally, we have the “Type” that, just like the priorities, can only take one value out of the three available.

3.4 Summary

A DBR methodology was applied to develop this dissertation, starting with a problem analysis, solution development, and iterative cycles of solution testing and refinement, producing two outcomes: tangibles and intangibles. Research subjects are vital within it, playing an essential role during the iterative process, helping to refine and rework the solution to reach a good end product. Researchers can organise their research into iterative cycles to test new resources and teaching methodologies while enhancing outcomes and processes due to their iterative character.

Due to its ability to actively integrate fieldwork participants (typically viewed as research subjects) in educational settings, DBR has received high praise. The DBR statistics provide proof to understand when, why, and how the educational intervention is expanding, increasing results, and encouraging participants, designers, and researchers to work together closely and implement data-driven ideas. Finally, DBR understands that environmental influences can affect how an educational intervention is carried out and how well it works.

The work plan of this dissertation was divided into eight tasks, one that was performed in the first semester (state of the art and initial structuring of the document) and seven in the second semester (selection of an ontology, exploration of Houdini for PCG development, creation of a Unity3D IEER framework and backoffice, development of IEER prototypes based on the selected ontology, test and evaluate prototypes and finalise the writing of this document), totalling twenty weeks of work.

We utilised a web tool called Quire to schedule and plan the tasks as the project was developed to assist us with task management. The majority of the tasks are "Epic," which implies that they may be and are divided into subtasks. By examining their full details, it is possible to see that a collection of checkboxes exists for each subtask. Tasks must have a title but may contain a description, collaborators, an end date, and tags. These tags are shown in a particular order, starting with the subject(s), moving on to priority, and concluding with type.

Chapter 4

Design and Development of a PCG solution for IEER

This chapter addresses the design aspects of the virtual environments' authoring tool and their implementation details. Furthermore, the proposed prototype will be presented.

4.1 Introduction

A tool to help the user build their IEER needed to be developed because the existing game required knowledge the user may not have, demanding the creation of JSON files with different and complex structures. Therefore, the backoffice arises to fill the existing gap and difficulties in using this tool in educational contexts.

In this chapter, we present the framework's requirements and architecture, the selected ontology to specify the learning paths to be further converted into IEERs, the backoffice design and implementation, new accessibility and adaptability features, the object placement strategy, and the developed prototype. In the backoffice section, we abord the initial constructed drafts and the implementation details. After that, we present accessibility and adaptability, where we demonstrate and describe the two implemented features designed to help this application reach a bigger group of users: (1) Text-to-speech; (2) Zoom. The object placement was an essential part of this project because we needed it to finish the prototype, which is presented before the prototype section. Lastly, we have the prototype, where we demonstrate the developed prototype with some screenshots together with a brief description of them.

In the last section of this chapter, we summarise the most vital aspects retained from the design and implementation of this virtual environments' authoring tool.

4.2 Requirements

A thorough list of general requirements was defined at the beginning of the framework implementation. Table 4.1, which lists these conditions and shows whether they have been satisfied, illustrates the framework's current position in the context of this dissertation.

Table 4.1: List of requirements

Requirement	**Status**
The teacher can view all the information about an IEER.	Done
The teacher can edit all the IEER information.	Done
The teacher can create new IEERs.	Done
The teacher can delete existing IEERs.	Done
The teacher can access detailed information about each input field through an information icon.	Done
The teacher can introduce links for Youtube videos, surveys and minigames.	Done
The teacher can introduce links from any source for the videos.	Future Work
The teacher can create multiplayer games.	Future Work
The teacher can create IEERs compatible with VR.	Future Work
The teacher can access a preview of the IEER on the EdScape server.	Future Work
The student can play the available IEERs.	Done
The student can view and interact with a web page inside Unity3D.	Done
The student can hear the TTS reading the displayed content.	Done
The student can zoom in and out inside the game while exploring the room.	Done
The framework can generate from a list of languages the corresponding IEER versions.	Done
The framework can generate new rooms and place interactive objects using PCG.	Future Work
The framework can place interactive objects randomly in a static room.	Done

4.3 Architecture

The primary structure of the framework comprises a Unity3D game integrated with the EdScape web interface, along with two external APIs. These components are interconnected, as illustrated in Figure 4.1. Additionally, Figure 4.2 depicts the essential use cases available to both educators and students.

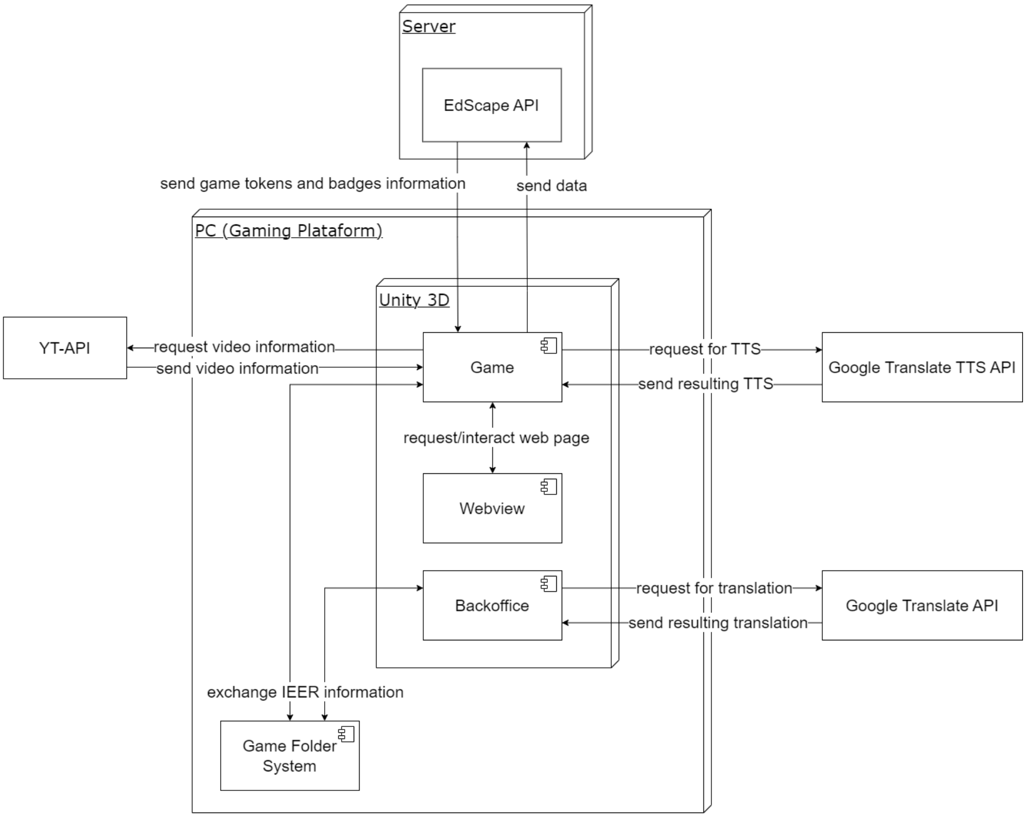


Figure 4.1: Framework architecture

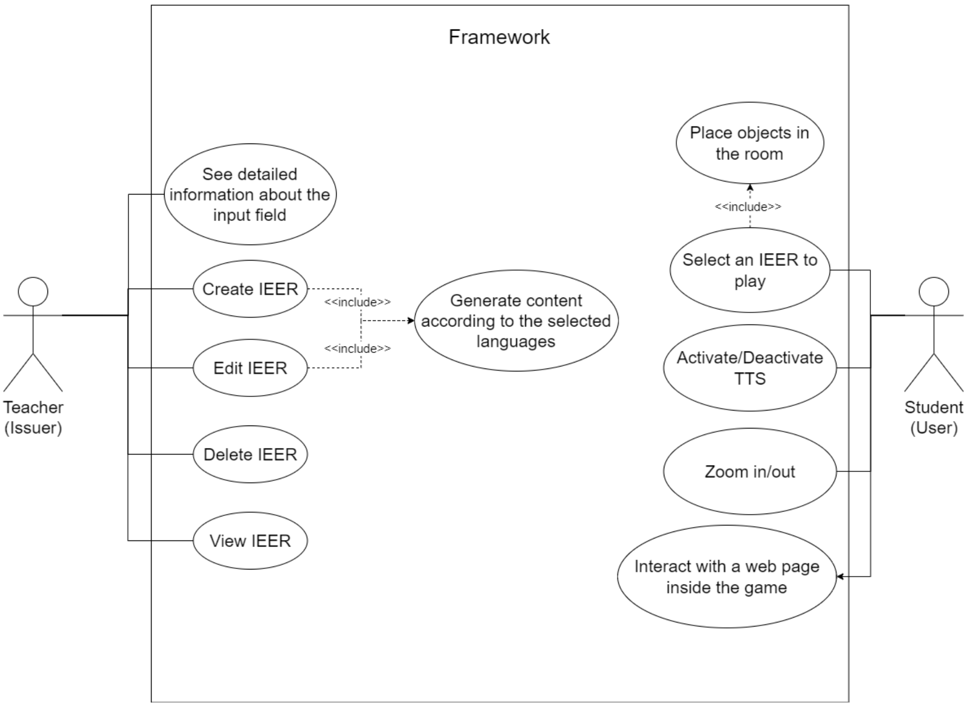


Figure 4.2: Framework use case diagram

4.4 Ontology

An ontology based on JSON was selected to specify learning paths to be further transformed in IEERs. The creation of learning paths is aborded in the following topic about backoffice. Still, the backoffice collects the required information from the user to create a learning path that is converted into the selected ontology. In this topic, we describe in detail the chosen ontology, answering the following questions: (1) How is the information stored and organised?; (2) What information is being stored?

To build this prototype, two sources were consulted: (1) the ontology developed by a colleague, Santos (2022), from the previous year, which developed and wrote a dissertation about IEERs; (2) an article from Fotaris and Mastoras (2022) about design thinking principles to create EERs, which contained a table with SMART questions that help create EERs. In the first source, like with the second one, only the information that we considered relevant was used, removing redundant or unsuitable fields for this problem. Table 4.2 contains the main attributes from the ontology that Santos (2022) developed, with the redundant and unnecessary fields removed, and an explanation of what that field will store. In Appendix A, it is possible to find all the information about the fields that compose the selected ontology.

Table 4.2: Main attributes from the ontology developed by Santos (2022)

Attribute	**Definition**
name	Name that will be given to the IEER.
test	Flag that indicates if the IEER is still being tested.
initialTime	Initial time of the countdown of the escape room in seconds.
textOptionIDs	List containing the available game languages.
infoIntroductionTitle	Title that appears before the game starts and is always available to the player.
infoIntroductionText	Text that appears before the game starts and is always available to the player.
introductionVideoPath	Video that follows the introductory information.
startText	Message that appears after the video and before the game starts.
timeEndedText	Message that appears when the game time runs out.
congratulationsText	Message that appears when the game finishes successfully.
endAfterSurveyText	Message that appears after answering the final survey.
initialSurveyPath	Initial survey that appears before the introductory information.
finalSurveyPath	Final survey that appears after the game finishes.
clueObjs	List of clues that appear to the user when playing.
thoughtObjs	List of thoughts that appear to the user when playing.
interactiveObjs	List of interactive objects in the room when playing.

Below, in Table 4.3, the new attributes and the SMART questions of Fotaris and Mastoras

(2022) that generated them, pertinent to this problem, are presented. The application is not currently using all these fields. The idea is to use them in future work to quickly perceive the goal of an existing IEER, which will take place on a personal page for each user where he can consult this information.

Table 4.3: New attributes generated from the SMART questions of Fotaris and Mastoras (2022)

Attribute	Definition
ieerPurpose	What is the overall purpose of the EER?
ieerLearningObjectives	What are the learning objectives this EER is going to support?
requiredKnowlage	What knowledge is required to succeed in the game?
numOfParticipants	How many participants need to play at the same time?
ieerCoursePosition	Where will the EER be positioned in the course curriculum?
ieerStoryType	Will the story be stand-alone like a full movie or framed as an episode with a continuous narrative arc?
ieerForEvaluation	Will the EER be used as a formative or summative assessment tool?
languageBarriers	Are there any language barriers that may prevent non-native speakers from playing the game?
differentLevelsOfImpairments	Are there any tasks that may prevent participants with differing mobility levels or sensory impairments from playing the game?

4.5 Backoffice

Giving the educator a more practical and intuitive solution is crucial when building their learning paths since it's easier to adapt and learn. Considering this, a backoffice for the manual definition of learning paths was developed. This manual tool was built on top of the Unity3D environment because the game was already built on it, avoiding using other technologies (fewer technologies equals a good learning curve). In this section, we explain how we reached the current version of the backoffice, step by step.

4.5.1 UI Design

At the beginning of a project is essential to start by making some drafts. We need to understand the client's needs before starting to plan how we will build the framework/tool/application to meet their needs. Both sides, the client and development team, win since the time wasted is reduced, and in our case, we are now closer to what the client wants as a final product. It is a good starting point, and using the DBR research methodology makes contact with the client more frequent.

We started by making a wireframe for the backoffice where it is possible to know, from the available windows, the paths a user may take while using the application. Four windows were presented in the wireframe, where one was already present in the EdScape game (homepage), while the three remaining, responsible for the IEER management, are new. All these windows contain new buttons that trigger different actions. Figure 4.3 shows the mentioned wireframe followed by a bullet point, and Figure 4.4, where the pages, buttons and popups are explained in detail.

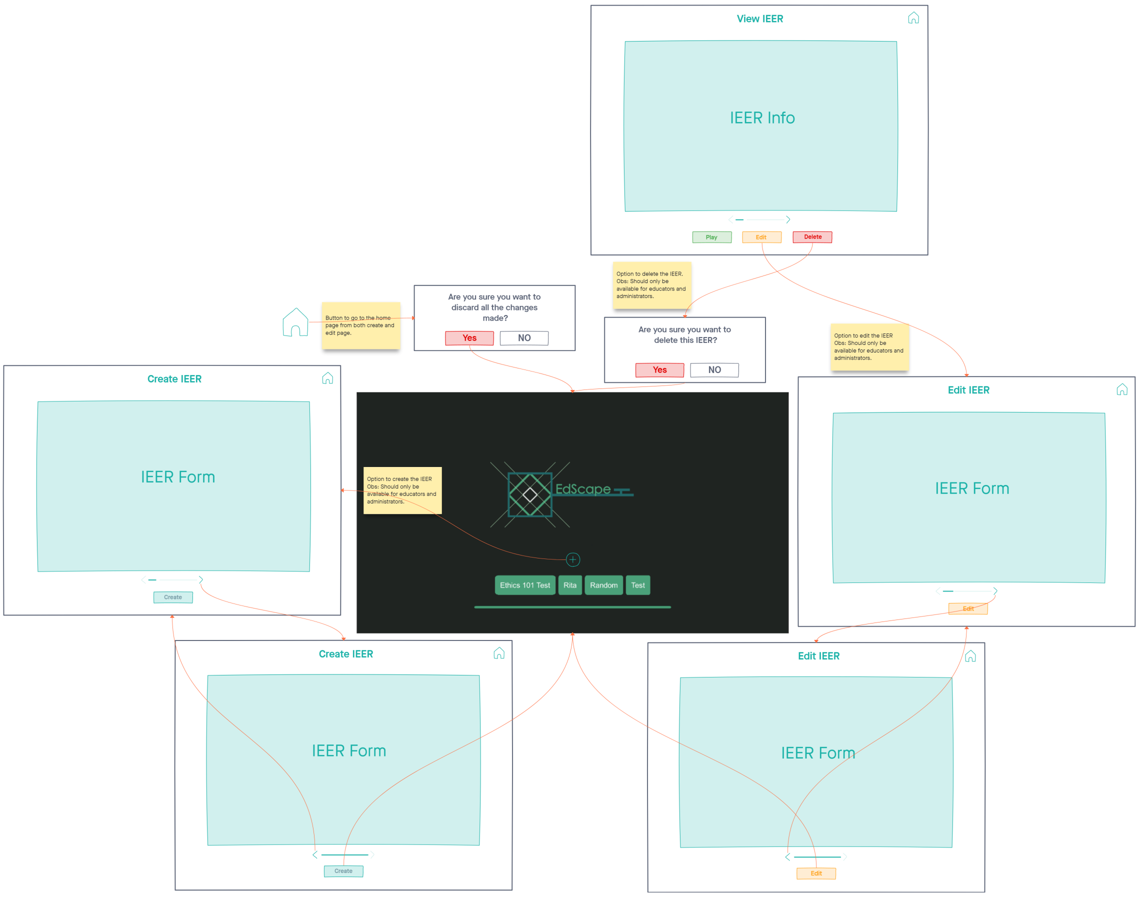
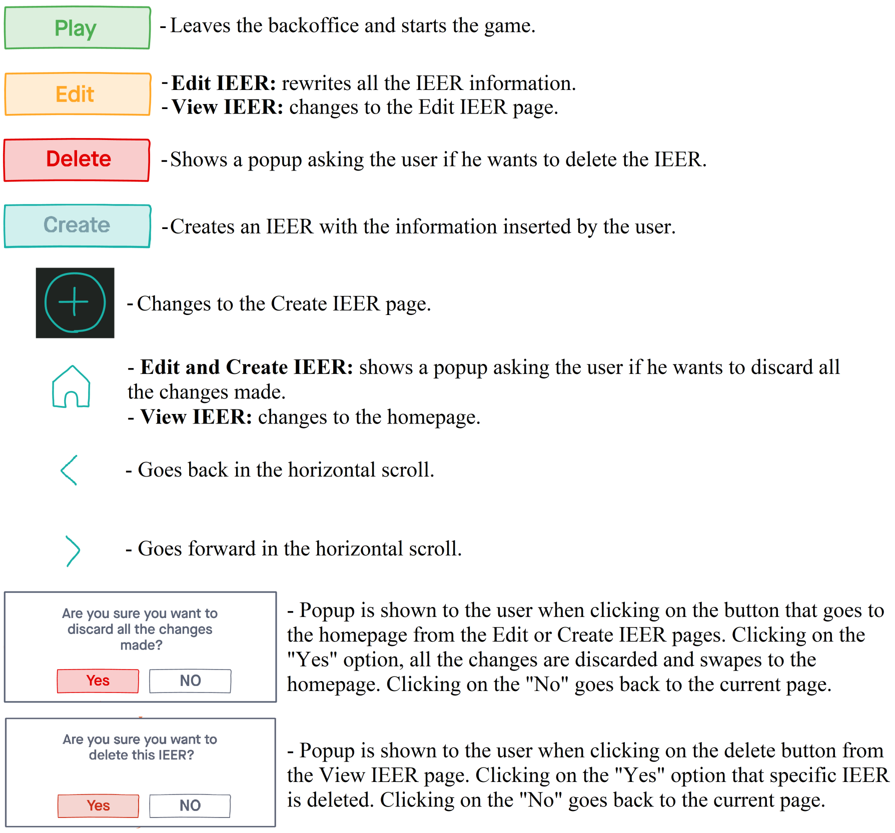

View IEER
IEER Info
Play
Edit
Delete
Option to delete the IEER. Obs: Should only be available for educators and administrators.
Button to go to the home page from both create and edit page.
Are you sure you want to discard all the changes made?
Yes
NO
Are you sure you want to delete this IEER?
Yes
NO
Option to edit the IEER Obs: Should only be available for educators and administrators.
Create IEER
IEER Form
Create
Option to create the IEER Obs: Should only be available for educators and administrators.
EdScape
Ethics 101 Test
Rita
Random
Test
Edit IEER
IEER Form
Edit
Create IEER
IEER Form
Create
Edit IEER
IEER Form
Edit

Figure 4.3: Backoffice wireframe

- **View IEER:** window where the user can observe all the information about that specific IEER. This page contains six buttons, two used to control the horizontal scroll and the remaining to play, edit, delete the IEER and go to the homepage.
- **Create IEER:** window where the user can create a new IEER. This page contains four buttons, two used to control the horizontal scroll and the remaining to make the IEER and go to the homepage.
- **Edit IEER:** window where the user can edit all the information about that specific IEER. This page contains four buttons, two used to control the horizontal scroll and the remaining to edit the IEER and go to the homepage.
- **EdScape Homepage:** window where it is possible to see the available IEER games. This page contains one button to create an IEER and a horizontal scroll with buttons to view the available IEERs.


Play
Edit
Delete
Create
Are you sure you want to discard all the changes made?
Yes
NO
Are you sure you want to delete this IEER?
Yes
NO

- Leaves the backoffice and starts the game.

- **Edit IEER:** rewrites all the IEER information.
- **View IEER:** changes to the Edit IEER page.

- Shows a popup asking the user if he wants to delete the IEER.

- Creates an IEER with the information inserted by the user.

- Changes to the Create IEER page.

- **Edit and Create IEER:** shows a popup asking the user if he wants to discard all the changes made.
- **View IEER:** changes to the homepage.

- Goes back in the horizontal scroll.

- Goes forward in the horizontal scroll.

- Popup is shown to the user when clicking on the button that goes to the homepage from the Edit or Create IEER pages. Clicking on the "Yes" option, all the changes are discarded and swapes to the homepage. Clicking on the "No" goes back to the current page.

- Popup is shown to the user when clicking on the delete button from the View IEER page. Clicking on the "Yes" option that specific IEER is deleted. Clicking on the "No" goes back to the current page.

Figure 4.4: Detailed explanation of every button and popup

After reaching the desired backoffice wireframe was time to start creating drafts for the input fields that would receive the data from the user. From the ontology mentioned in the previous section, we build a possible input field draft for each attribute that compose it. Appendix B shows all the constructed input field drafts. Some of those drafts may not correspond to the actual input field or may not be used since some of those were discarded.

4.5.2 Implementation

The work could begin once the ontology and necessary information were in place. Some code already existed from the previous year, developed by two colleagues that finished their master's in informatics and computing engineering. Therefore, we needed to get everything set up before starting coding, downloading the required Unity Editor version (2020.3.30f1), cloning the GitLab repository and checking if we had any issues opening and running the project. No problems were detected, so we started by analysing all the existing code, trying to figure out what was happening behind the scenes to have a good idea where the tool we were about to start developing could be inserted, finding gateways for potential near future merge.

The goal at first was to create an excellent initial infrastructure independent from the existing code to be further merged with it. A scene with an initial panel was created for the backoffice operations, containing an area to insert the page title, a button to go back to the homepage, a content area where the form fields will be displayed, horizontal and vertical scrolls to control the visualisation of the content, being horizontal to proceed to the next or retreat to the previous page while the vertical to scroll the content from the same page, a progress bar and two buttons connected to the horizontal scroll as a way to send feedback to the user about their location in the scroll and finally a footer area for buttons placement (see Figure 4.5).

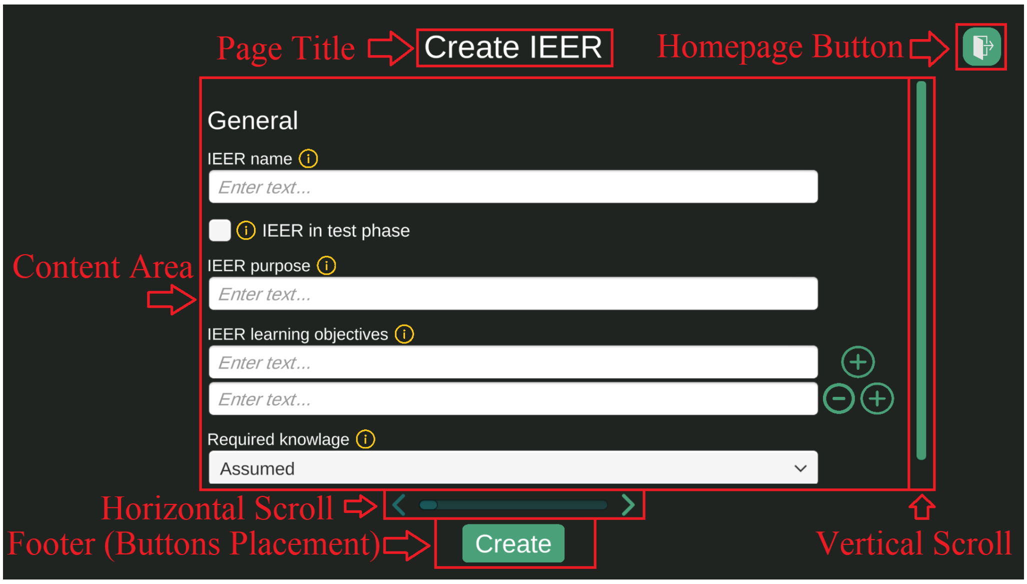
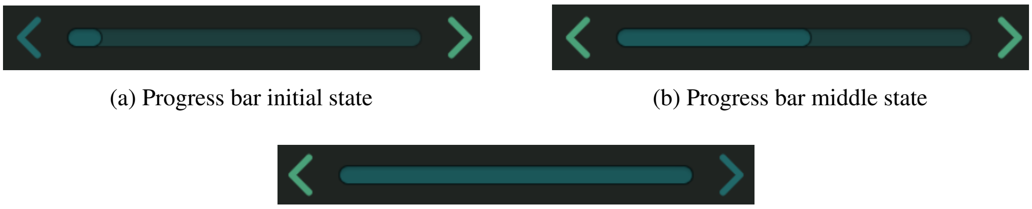


Figure 4.5: Backoffice panel with its different regions labelled

The panel was brought from the draft to a Unity3D prefab without any implemented functionality except for the horizontal and vertical scrolls, the progress bar and the two buttons. With the help of a C# script, it was possible to connect the horizontal scroll with the progress bar and the two buttons. When the user presses one of the two buttons, the script updates the progress bar state considering the horizontal scroll value (see Figure 4.6).



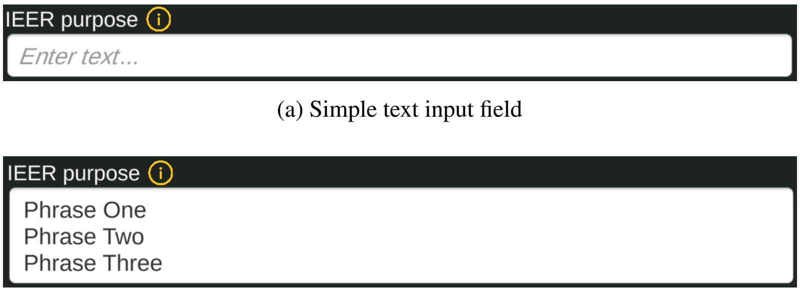
(a) Progress bar initial state

(b) Progress bar middle state

(c) Progress bar final state

Figure 4.6: Three different progress bar states

The next thing to do was the content area, creating prefabs for the primary inputs considering the developed drafts: text, multi-text, dropdown, integer, decimal and toggle. Unity3D already provided some UI objects for those inputs, but we wanted something else. In all the mentioned input fields, except text and multi-text, it was only necessary to add a label, while the text was necessary to create a resizable text input field prefab (a field that automatically increases its size vertically when the phrase is too big or when a newline is inserted, see Figure 4.7) and the multi-text was required to build a script to add and remove text fields. When the user interacts with the plus and minus buttons, the script inserts or removes a new text field under the area next to the controls that the user interacted with.



IEER purpose ⓘ

Enter text...

(a) Simple text input field

IEER purpose ⓘ

Phrase One
Phrase Two
Phrase Three

(b) Input field autoincrement its size to support the inserted text

Figure 4.7: Resizable input field

But while testing these prefabs, something needed to be fixed. The vertical layout group must be updated after adding and removing text fields. The update was executed after some time, but it took too long and needed to be more suitable for the user experience. After some research, we discovered that a method called “LayoutRebuilder.ForceRebuildLayoutImmediate” could force the layout groups to update. According to what is written in the documentation, this method should be used carefully because it is too expensive, concerning the time needed to execute it. Considering this, we built a script that checks every frame if the rectangle containing the objects changes its size, and if so, calls this function, meaning that this was only used when it was strictly necessary.

The prefabs for the initial input fields were created. We needed to start developing scripts to manage them: read, clear, reset, and set the value from the input field and grant or deny the user interaction.

An abstract class called “FormField” was created for the input fields to extend, granting an easier way to reach the same method of all the available input fields, meaning that we can store all of them in a list and loop through them to get their value or make other operations. In this abstract class, it is possible to find three primary methods that are called when accessing one of the three pages (Create, Edit and View), five abstract methods that are indirectly called when calling the three previous methods responsible for getting, setting, and clearing the value inserted by the user and disable/enable the input field.

In addition to the methods, there also exist four fields, “fieldId” used to identify the form field (with its getter), the “valueIsSet” to know if the user set the value, “initialValue” stores the respective initial value and “isInteractable” to know if that field was initially set to be interactable or not.

Figure 4.8 shows the UML class diagram of the “FormField” abstract class. In this figure is possible to see all the classes that extend the "FormField" abstract class, including classes of the prefabs we mentioned before (text, multi-text, dropdown, integer, decimal and toggle) and also the ones that will be further introduced (toggle group, file or URL input, dropdown group, minigame input and multi-input).

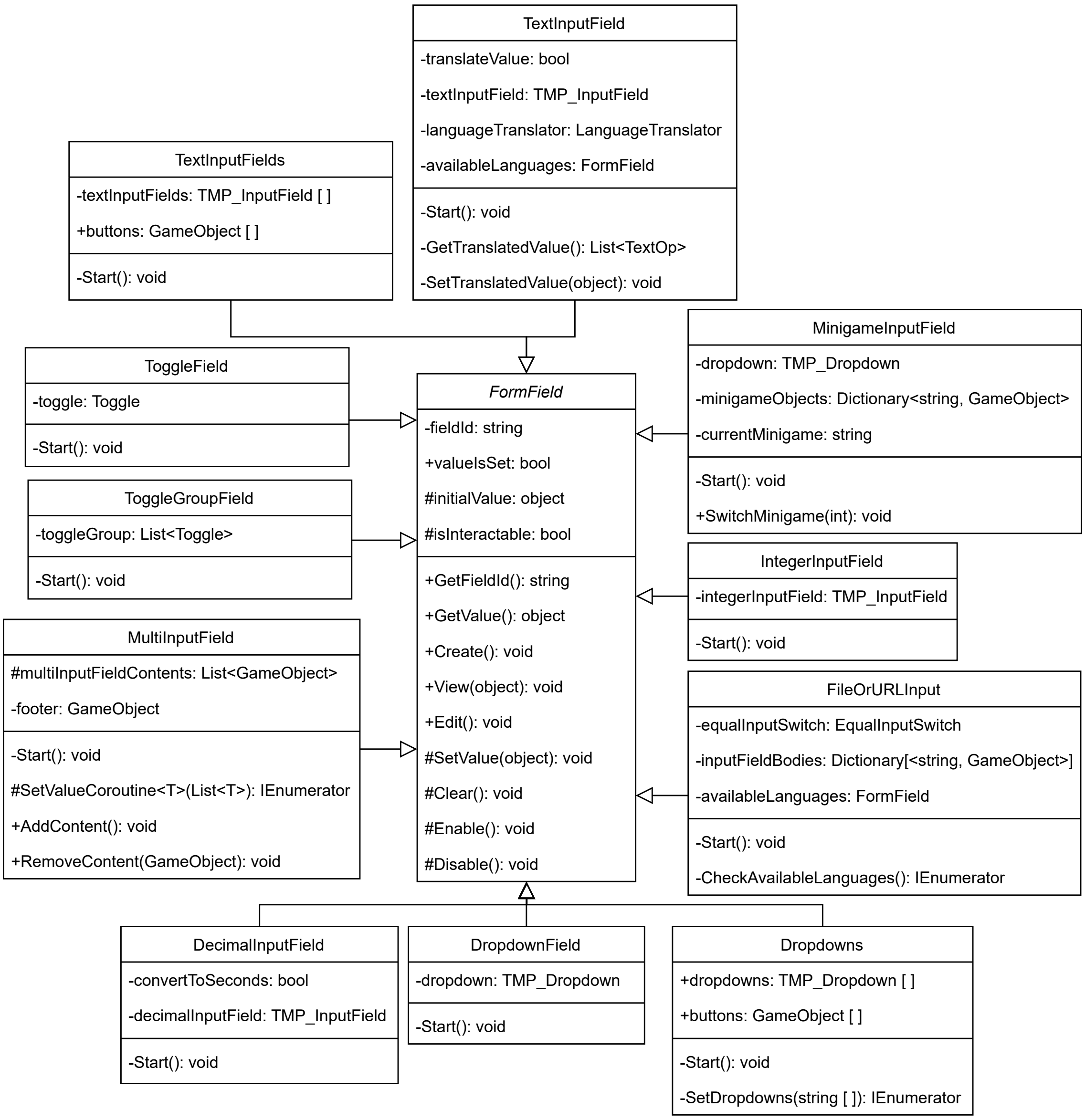


Figure 4.8: UML class diagram of the "FormField" abstract class

Then, we construct a backoffice panel with the first input fields, add their prefab to the backoffice panel prefab, merge both codes and connect the EdScape homepage to the backoffice pages. However, our approach for the backoffice was to have three panels, one for each backoffice page, but it only made our lives difficult. So, we decided to have only one panel with different objects for the title and footer since those were the only ones that differed from page to page.

Developing a C# script facilitated the activation/deactivation of the title and footer objects, considering the page that would be displayed.

Referring back to the merge of both codes, a new button was added to the EdScape homepage to change to the “Create IEER” page, the controls that already existed to play the IEERs were modified to switch to the “View IEER” page and to make this connection, two new classes were created.

The “BackofficeController” stores all the available form fields in a dictionary (dictionary key equals to form field identifier), and is responsible for the activation/deactivation of the title and footer objects when switching between backoffice pages and has all the methods for each button present in the backoffice pages and EdScape homepage. However, this class, considering all the operations available to the form fields, only executes simple things such as clearing all the form fields and activating them for the edition, all the work regarding reading, setting, saving in a JSON file and deleting the JSON file is executed by the “IEER” class. The “IEER”, as mentioned before, is responsible for the work that involves data flow between the form fields and the classes that temporarily hold them before saving their data to a JSON file in the local storage.

In the “IEER” class, the strategy used to easily loop through all the form fields to read or set their value was, in the beginning, challenging to find. Deep research was needed to find solutions to overcome this challenge because the only solution was adding one line of code for each field we wanted to save, and this solution was not friendly and scalable. After trying different strategies that led us to failure, the solution that we were finding was reached.

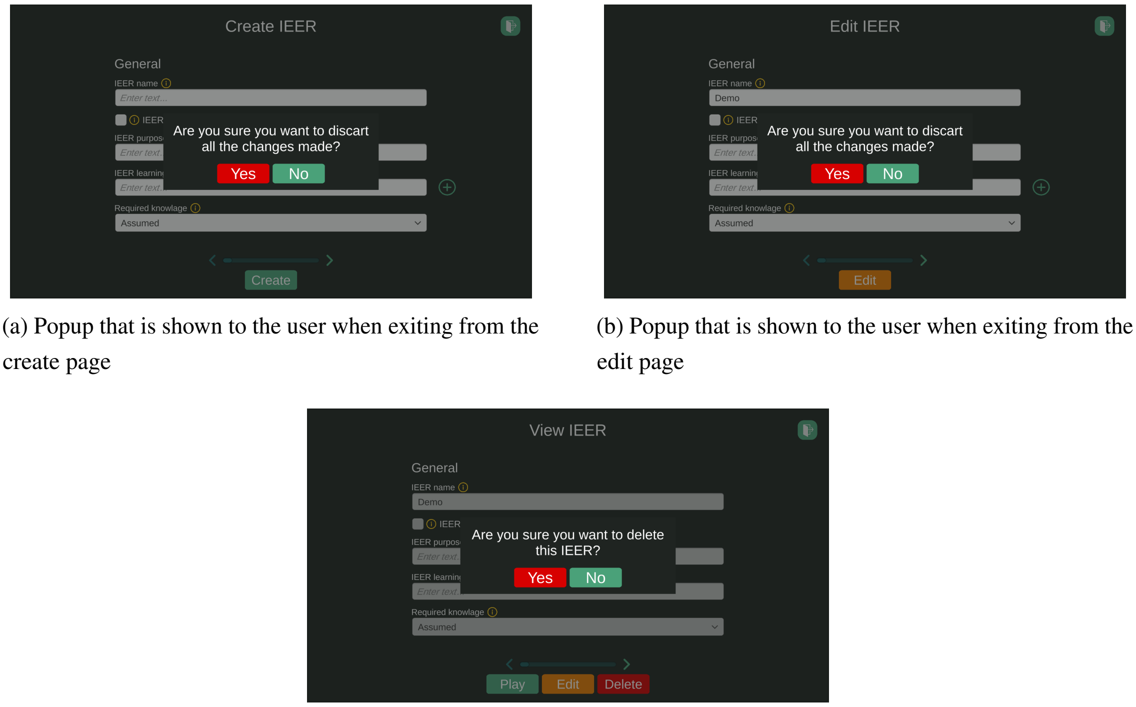
We set the “fieldId” of each form field with the same name as the corresponding class field that would temporarily store its value.

Using the “System.Type” and “System.Reflection.FieldInfo” classes, we can have access to the field of any class by knowing its name, and since we have the form field Ids with the same name as the class fields, we can easily access them and get or set their information. Listings 4.1 shows the part of the “IEER” class code that executes this process.

```
// Gets all the diferent class Types
Type gameSettingsType = typeof(GameSettings);
Type configGameObjType = typeof(ConfigGameObj);

// Assing gameSettings and configGameObj values to formFields
// Loops through all form fields
foreach (var formField in formFields)
{
    // Search for a field in the GameSettings class with the same name as form field key
    FieldInfo gameSettingsField = gameSettingsType.GetField(formField.Key);

    // if was found
    if (gameSettingsField != null){
        formField.Value.View(gameSettingsField.GetValue(gameSettings));
        // Skips to the next frame. This way does not wait for the loop to complete.
        yield return null;
        continue;
    }

    // Search for a field in the configGameObj class with the same name as form field key
    FieldInfo configGameObjField = configGameObjType.GetField(formField.Key);

    // if was found
    if (configGameObjField != null){
        formField.Value.View(configGameObjField.GetValue(configGameObj));
        // Skips to the next frame. This way does not wait for the loop to complete.
        yield return null;
        continue;
    }
}
```

Listing 4.1: Code to get the values from the classes and send them to the form fields

Furthermore, we built a prefab for the popups for exit operations from create/edit pages and when deleting an IEER (see Figure 4.9) and made some fixes and improvements to stabilise the existing code, such as: (1) Remove the functionality from Ana Rita's project, which consists of having more than one IEER per name. This feature randomly chooses an IEER with the same name. This makes it difficult in the backoffice because it is impossible to determine which IEERs with the same name the user would edit or delete; (2) Problems with loading screens. The loading was not being executed because the code was not letting the script responsible for making the loading animation run. The solution was to change the existing code to asynchronous; (3) Add cast check in the form fields. It is essential to check the fields that come from outside. (4) Put the input of integers and decimals with automatic adjustment. The input area should adjust to the size of the number; (5) Reset the horizontal and vertical scroll bars when clearing the backoffice form.



Create IEER

General

IEER name: Enter text...

Are you sure you want to discart all the changes made?

Required knowlage: Assumed

(a) Popup that is shown to the user when exiting from the create page

Edit IEER

General

IEER name: Demo

Are you sure you want to discart all the changes made?

Required knowlage: Assumed

(b) Popup that is shown to the user when exiting from the edit page

View IEER

General

IEER name: Demo

Are you sure you want to delete this IEER?

Required knowlage: Assumed

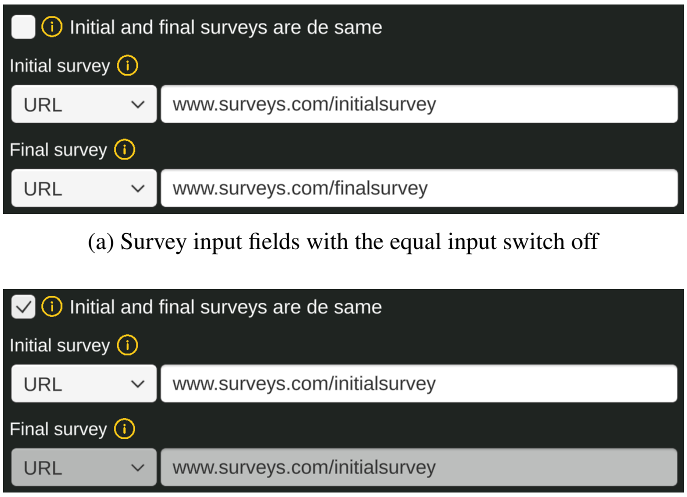
(c) Popup that is shown to the user tries to delete the IEER

Figure 4.9: Three different popups that can be presented to the user

With the main problems and improvements done, we started creating and adding the new and last simple form fields: toggle group, file or URL input, dropdown group and minigame input.

The toggle group, as the name indicates, is a group of toggles where the user can choose more than one option, and it was used for the language selection. The file or URL input is an input field where the user can input a file or a URL. Still, in the current version, it is only possible to insert the URL since the file upload was unfeasible because of the file system policies and because the URL insertion is more accessible and practical to the user. The dropdown group, as well as the toggle group, is a group of dropdowns where the user can select multiple options from the dropdowns. When an option is selected, it will not appear in the other dropdowns.

Finally, the minigame input, as well as the file or URL input, the available minigames present in the existing code turned out to be not feasible because they need to be created by hand and with a specific structure. Concerning this fact, we decided only to use URLs for this feature. In the case of survey input, it was also pertinent to add functionality to indicate that the initial and final surveys would be the same. So, a prefab for the "EqualInputSwitch" was created. Based on the toggle, it is possible to select if both inputs will be the same, and if so, copy what is inserted in the first input and copy to the second one (see Figure 4.10).



□ ⓘ Initial and final surveys are de same

Initial survey ⓘ

URL www.surveys.com/initialsurvey

Final survey ⓘ

URL www.surveys.com/finalsurvey

(a) Survey input fields with the equal input switch off

☑ ⓘ Initial and final surveys are de same

Initial survey ⓘ

URL www.surveys.com/initialsurvey

Final survey ⓘ

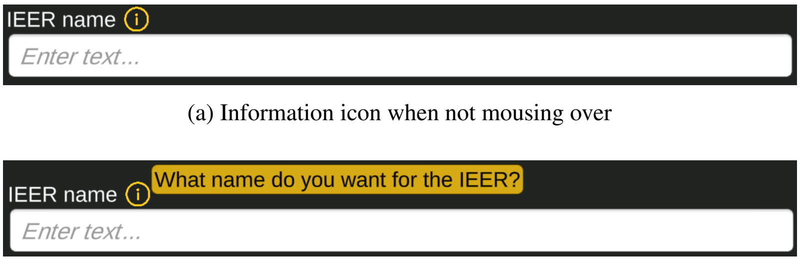
URL www.surveys.com/initialsurvey

(b) Survey input fields with the equal input switch on

Figure 4.10: Equal input switch functionality

The development of all the simple form fields that would be used in the backoffice facilitated the next task of building the composed form fields, a group of different simple form fields, for the clues, thoughts, and interactive objects. With the significant work done, it was only necessary to create the prefab for each composed form field, add simple form fields as desired, and build two scripts, the "MultiInputField" that extends the "FormField" to handle the different clues, thoughts and interactive objects and the "MultiInputFieldContent" to connect to all the form fields within the composed input field.

The "MultiInputField" is responsible for the list of "MultiInputFieldContent", whereas this class contains multiple simple form fields. To help the user understand what was needed to insert in every form field, a prefab for the informative icon was developed, where when the user mouses over the icon, a text box appears with a form field explanation (see Figure 4.11).



IEER name ⓘ

Enter text...

(a) Information icon when not mousing over

IEER name ⓘ What name do you want for the IEER?

Enter text...

(b) Information icon when mousing over

Figure 4.11: Information icon behaviour

Some features in this project were implemented with the help of external resources, and it is important to mention them. Generating the translated versions of the game content was possible using Google Translate API, where we send the text and the language we want to translate, and the API returns the translation. The search for the streaming URL of a Youtube video was possible due to an API called "YT-API" from RapidAPI, whereby sending the video ID returns all the information about the video. We only extract from that information the streaming URL to be used in the Video Player of Unity3D. And finally, access to a web view of the URL inserted by the user could only be possible due to an open-source code accessible in GitHub with the name "unity_browser" from tunerok.

The majority of the planned tasks regarding the backoffice were completed, so it was time to work on its design, change the colour pallet to the same as the EdScape and stabilise the code, fix the problems that were found during the tests and make some improvements such as: (1) Give the possibility to change between minutes and seconds in decimal input. The user should insert the time in minutes, but at the end, it is saved in seconds, and when the information is presented to the user, the time is converted back to minutes; (2) Add new languages to the toggle group and order them alphabetically; (3) Populate the dropdowns with the available options; (4) Check the information of each input field and improve it if necessary; (5) Fix the loading screen when loading the information of the IEER to be displayed. The loading panel was not waiting for the information to be loaded; (6) Sort all the dropdown options using code; (7) Maintain consistency in input labels; (8) Fix existing IEERs to work with the backoffice. The existing IEERs were not optimised to work with the developed backoffice, so it was necessary to change them.

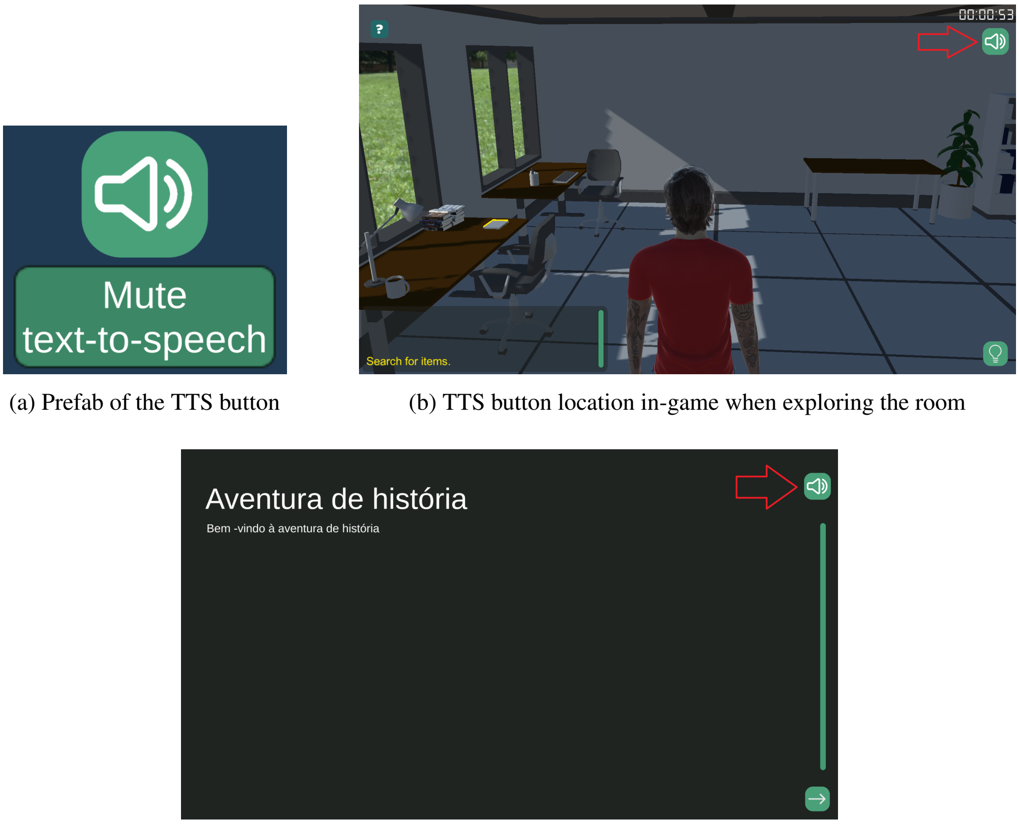
To close this subsection, in Appendix C is possible to see all the prefabs that were built during the project development. There, it is possible to see the "Minigame" and "File or URL" prefabs, and since they are influenced by the languages selected, they present more than one input area for each language.

4.6 Accessibility and Adaptability

When developing a video game, particularly an educational game, is vital to think about the final user and if he can play the game without any difficulty. The game must allow everyone to play and enjoy everything the game offers, even if that person has some impairment. In this section, we abord two features that aim to improve the game accessibility and adaptability: Text-To-Speech and in-game Zoom.

4.6.1 Text-To-Speech (TTS)

The TTS is a feature that helps users with any visual issues, with difficulty reading small text or even users that do not have the patience to read all the text. In all the text content of the game, the TTS feature is available to help the user by reading these contents in the selected language. By default, this feature is always on, but it is possible to turn it off using a button present in the top-right corner of the screen. When the user mouses over the button, a message indicates what this button does. The possibility of switching on/off the TTS is essential because a feature designed to help users can, after some time, be annoying and harm the user experience. Figure 4.12 shows the prefab of the button that interacts with the TTS functionality and some game screenshots locating this button.



(a) Prefab of the TTS button

(b) TTS button location in-game when exploring the room

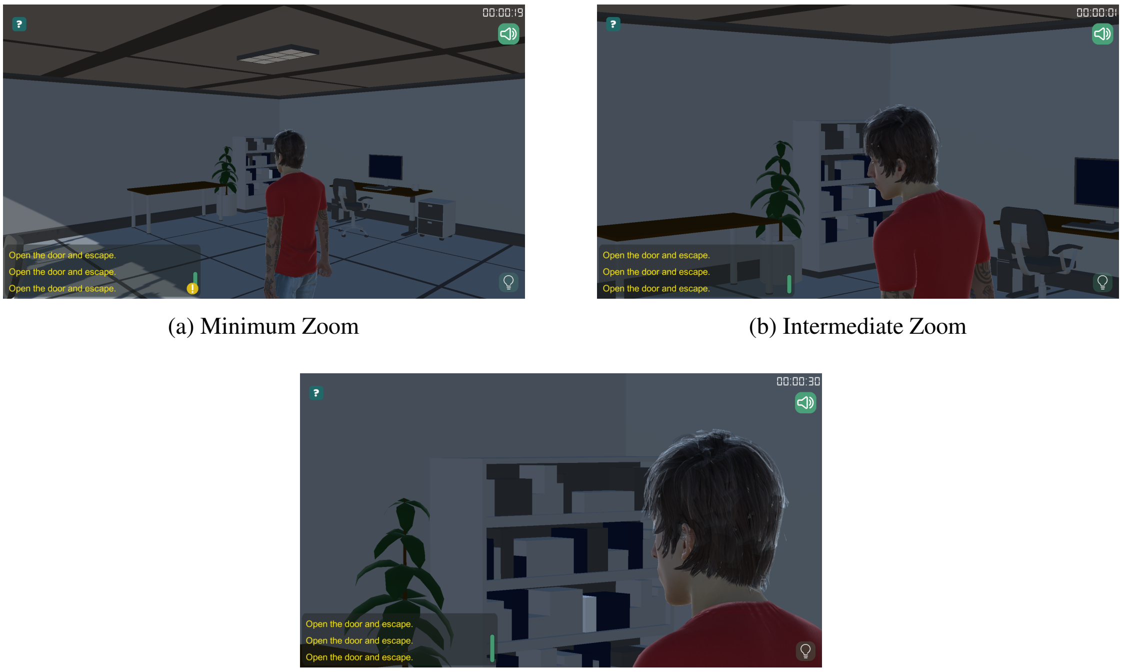
(c) TTS button location in the game introduction

Figure 4.12: Information icon behaviour

To build this feature, the Google Translate TTS API was used. The API returned an audio clip with the pretended content by sending a GET request with the text and the language we wanted. After getting the audio clip, if no error has occurred, it is sent to the audio source and played.

4.6.2 Zoom

As the TTS aborded in the previous subsection, the Zoom feature aims to help visually impaired users. This feature is only available during gameplay and strictly when exploring the room, allowing users to zoom in/out. When the user uses the mouse wheel, it is possible to control the amount of zoom (rotate forward -> zoom in; rotate backwards -> zoom out). We used the camera field of view (FOV) to perform this camera behaviour. By changing the camera FOV, it is possible to create this movement. So, we check the mouse scroll wheel every frame, and given the scroll direction, the zoom in/out is performed. Figure 4.13 shows three screenshots of the minimum, maximum and intermediate in-game Zoom.



Open the door and escape.
Open the door and escape.
Open the door and escape.

(a) Minimum Zoom

Open the door and escape.
Open the door and escape.
Open the door and escape.

(b) Intermediate Zoom

Open the door and escape.
Open the door and escape.
Open the door and escape.

(c) Maximum Zoom

Figure 4.13: Three Zoom states

4.7 Object Placement

Initially, the room generation and object placement were planned to use PCG, but this was impossible due to the lack of time. However, the backoffice required the generation of the coordinates for the position and rotation of the interactive objects. These coordinates are necessary for us to present all the developed backoffice features because the game would assume the placement of the interactive objects in the origin of the referential.

The class “InteractiveObjectPlacer” was developed to randomly choose a position and a rotation for the available objects to overcome this issue. This information about the object’s position and rotation was hardcoded and stored in a dictionary in the same class, where the keys were the interactive objects identifier and the value a list of “PlacementArea”, areas where the interactive objects could be placed. Besides having a dictionary with all the placement areas, this class has a “Random” object to generate random numbers and five methods to choose from the lists, given the random number, the “PlacementArea”, initial position, final position, initial rotation and final rotation.

The “PlacemetArea” class holds information about the area where the interactive objects can be placed. It has eight float arrays to store the minimum and maximum of each position and rotation coordinate (initial and final), four methods to get each coordinate, and one function to generate a random float between two floats. When we want, for example, the initial position coordinate, we call the “GetInitialPosition”, and it randomly chooses a coordinate between the minimum initial position and maximum initial position.

However, this solution could be better since it does not prevent objects with similar placement areas from being placed in the exact location. And when this happens, we can not guarantee that the user can interact with both entities, which could negatively impact user experience and generate endless games.

4.8 Prototype

With all the previously mentioned features implemented, it was possible to build a functional prototype for further tests with users. This section demonstrates the prototype using screenshots and briefly explains them. We divide this section into subsections for the different prototype parts: EdScape Homepage, Create/Edit IEER, View IEER, Delete IEER, Webview, TTS and Zoom. A nine-minute video demonstrating this prototype is available in https://youtu.be/Bt7MNFLqvdM.

4.8.1 EdScape Homepage

This is the first page shown to the user after the login/registration. On this page exist the game logo, a button with a plus symbol and a horizontal scroll with a group of buttons. The button with the plus symbol takes the user to the “Create IEER” page, where he can create new IEERs. The group of buttons in the horizontal scroll takes the user to the "View IEER”, where he can visualise the IEER information and play/edit/delete the IEER. Each button in the horizontal scroll represents an IEER that can be accessed, and when an IEER is created or deleted, this group of buttons is updated (see Figure 4.14).

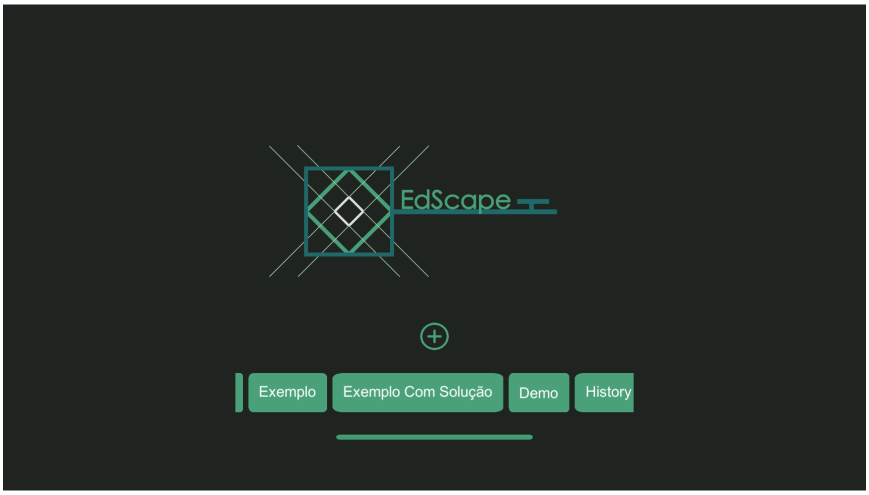
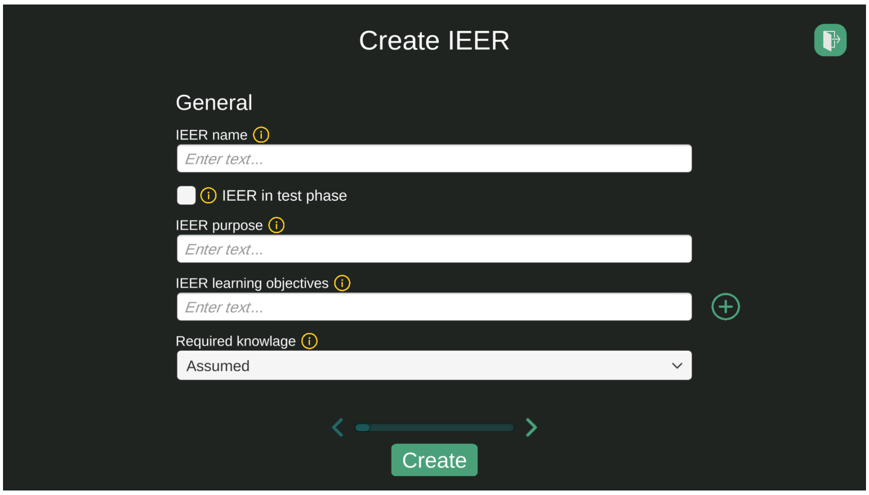


Figure 4.14: EdScape homepage

4.8.2 Create/Edit IEER

Both pages are almost identical, except for the page title and the footer where the buttons appear. In the case of the "Create IEER" page, all the fields are in their initial state, while in the "Edit IEER", some areas may appear filled. Figures 4.15, 4.16 and 4.17 show the first three pages, where we have inputs of general information that can be used to quickly identify an IEER.



Create IEER

General

IEER name ⓘ
Enter text...

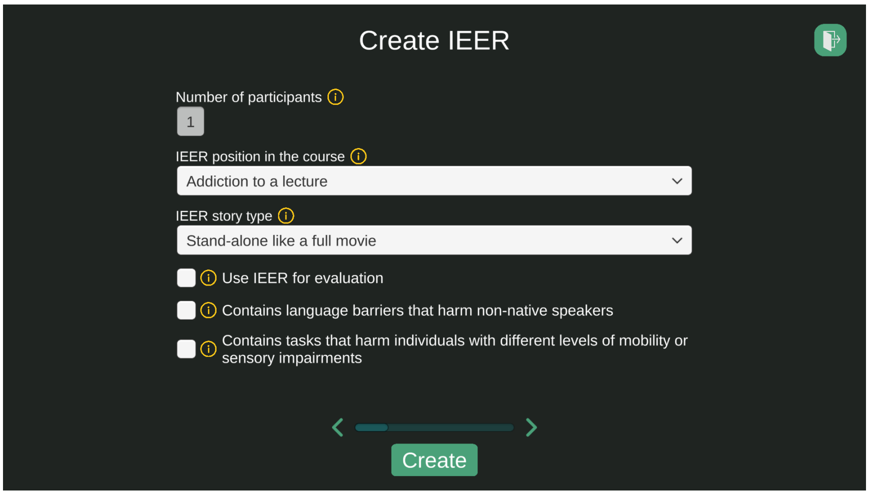
□ ⓘ IEER in test phase

IEER purpose ⓘ
Enter text...

IEER learning objectives ⓘ
Enter text...

Required knowlage ⓘ
Assumed

Figure 4.15: Page 1 of "Create IEER"



Create IEER

Number of participants ⓘ
1

IEER position in the course ⓘ
Addiction to a lecture

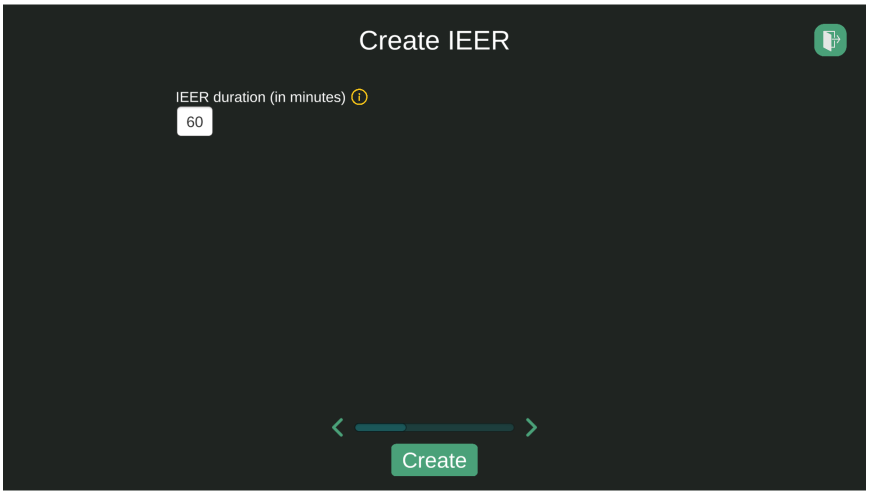
IEER story type ⓘ
Stand-alone like a full movie

□ ⓘ Use IEER for evaluation

□ ⓘ Contains language barriers that harm non-native speakers

□ ⓘ Contains tasks that harm individuals with different levels of mobility or sensory impairments

Figure 4.16: Page 2 of "Create IEER"

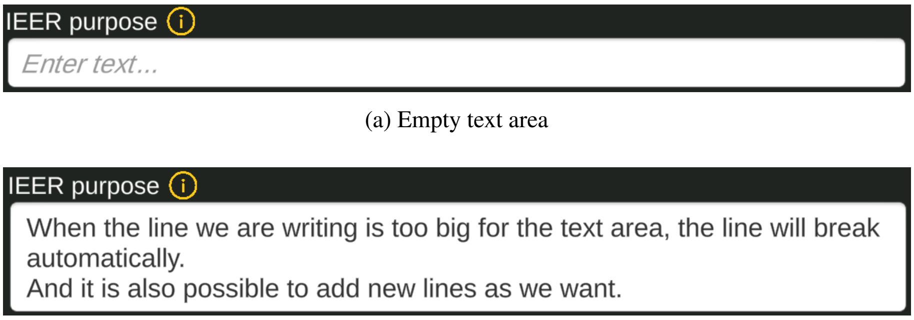


Create IEER

IEER duration (in minutes) ⓘ
60

Figure 4.17: Page 3 of "Create IEER"

On these three pages, it is possible to see simple form fields such as text areas, toggles, multi-text areas, dropdowns and integer or decimal inputs. All of them have a standard behaviour. However, the text area, multi-text area, and integer or decimal input have some different behaviours. The text area originally was an infinite line, meaning there was no possibility of adding new lines. So, we changed that, and now it is possible to add new lines and all the written text is shown inside the box without overflowing the text area (see Figure 4.18).



IEER purpose ⓘ

Enter text...

(a) Empty text area

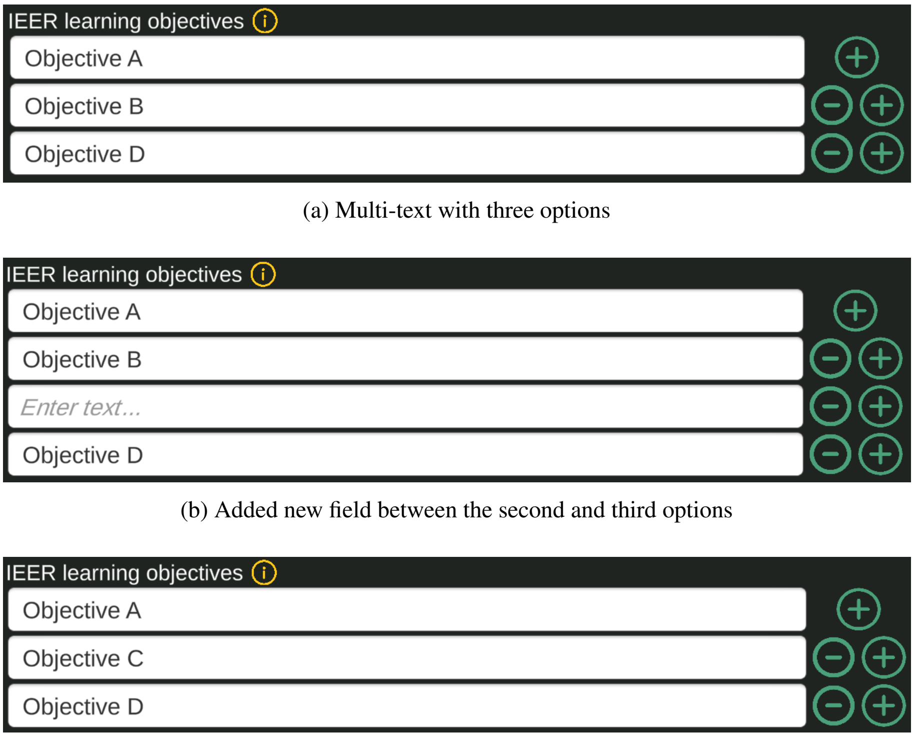
IEER purpose ⓘ

When the line we are writing is too big for the text area, the line will break automatically.
And it is also possible to add new lines as we want.

(b) Text area with the new lines

Figure 4.18: Text area behaviour

The multi-text area was built from scratch, and the idea was to construct an input like a bullet point. This comprises multiple text areas with side buttons to add or remove text areas. It allows the user to add/remove text areas in the middle of the existing ones (see Figure 4.19).



IEER learning objectives ⓘ

Objective A ⊕
Objective B ⊖ ⊕
Objective D ⊖ ⊕

(a) Multi-text with three options

IEER learning objectives ⓘ

Objective A ⊕
Objective B ⊖ ⊕
Enter text... ⊖ ⊕
Objective D ⊖ ⊕

(b) Added new field between the second and third options

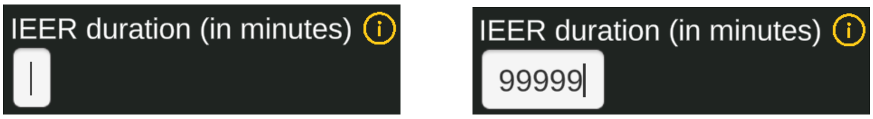
IEER learning objectives ⓘ

Objective A ⊕
Objective C ⊖ ⊕
Objective D ⊖ ⊕

(c) Filled the new field and removed the second one

Figure 4.19: Multi-text input field

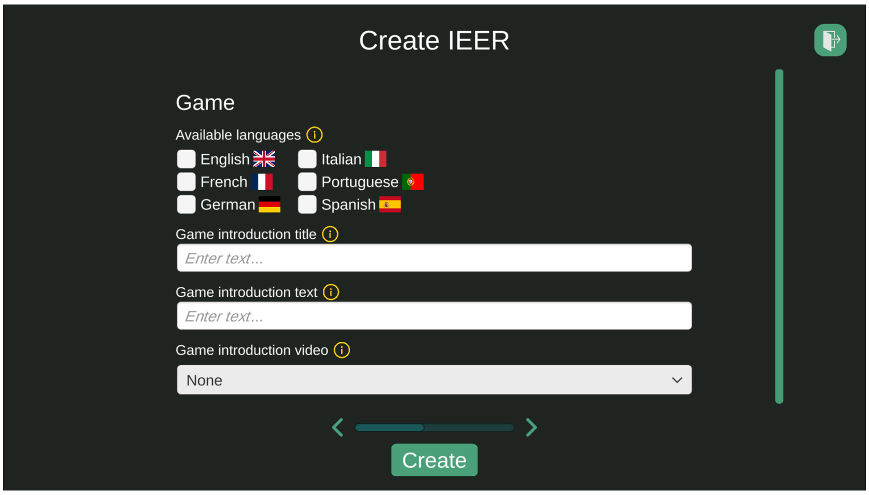
The integer or decimal input has a particular behaviour regarding its input area. Figure 4.20 shows that when the user introduces content, the area size will adapt to the number size.



(a) Integer/decimal input without any value (b) Integer/decimal input with a big value

Figure 4.20: Auto resize of the integer/decimal input area

Next, in Figures 4.21, 4.22 and 4.23, we have three pages regarding the game information that will appear during the game, such as text content, videos and surveys. Here we introduce three new form fields: toggle group, file or URL input and equal input switch.


Create IEER

Game

Available languages ⓘ

□ English □ Italian
□ French □ Portuguese
□ German □ Spanish

Game introduction title ⓘ

Enter text...

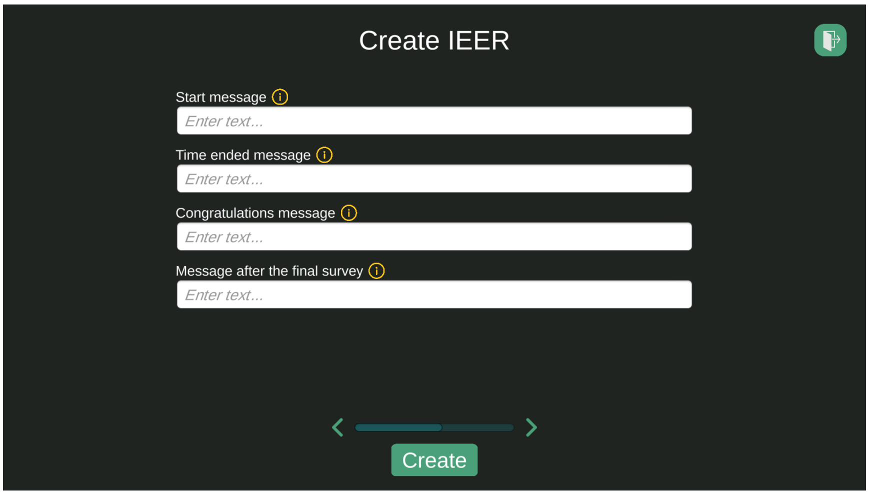
Game introduction text ⓘ

Enter text...

Game introduction video ⓘ

None

Figure 4.21: Page 4 of "Create IEER"


Create IEER

Start message ⓘ

Enter text...

Time ended message ⓘ

Enter text...

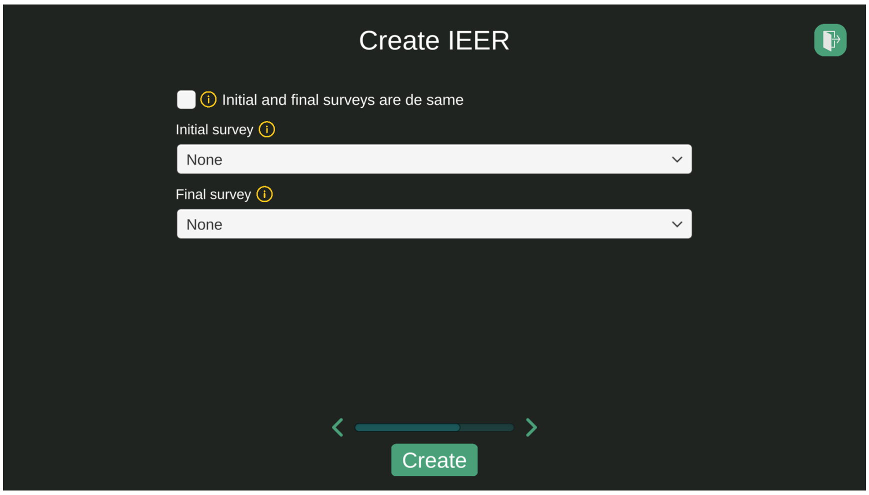
Congratulations message ⓘ

Enter text...

Message after the final survey ⓘ

Enter text...

Figure 4.22: Page 5 of "Create IEER"



Create IEER

□ Initial and final surveys are de same

Initial survey

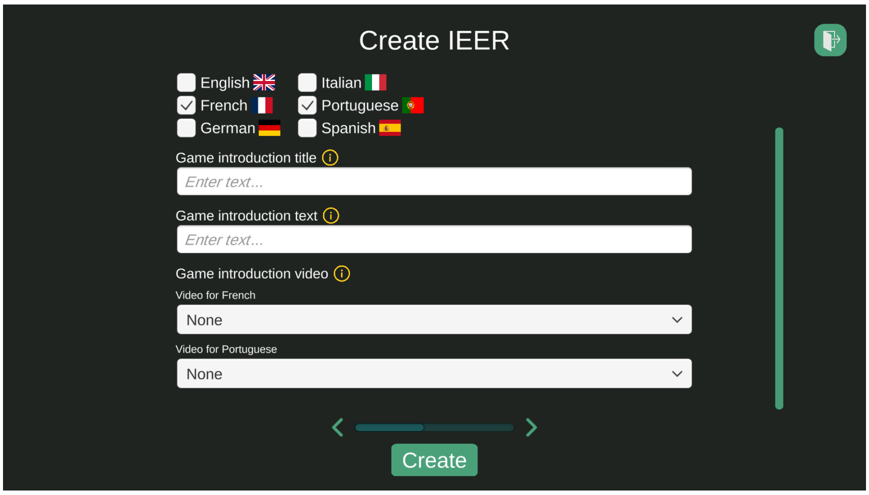
None

Final survey

None

Figure 4.23: Page 6 of "Create IEER"

The toggle group is only used in this part of the backoffice, specifically for the language selection. The file or URL input is used for the videos, surveys and, further on, the minigames. Despite being called a file or URL, it does not allow files because it was considered unfeasible for this project, so the file upload functionality was disabled. This input has a connection with the toggle group responsible for the languages. As shown in Figure 4.24, when one or more languages are selected, new input areas appear for each language because it is impossible to translate links in contrast with the text content.



Create IEER

□ English ☐ Italian

☑ French ☑ Portuguese

□ German □ Spanish

Game introduction title

Enter text...

Game introduction text

Enter text...

Game introduction video

Video for French

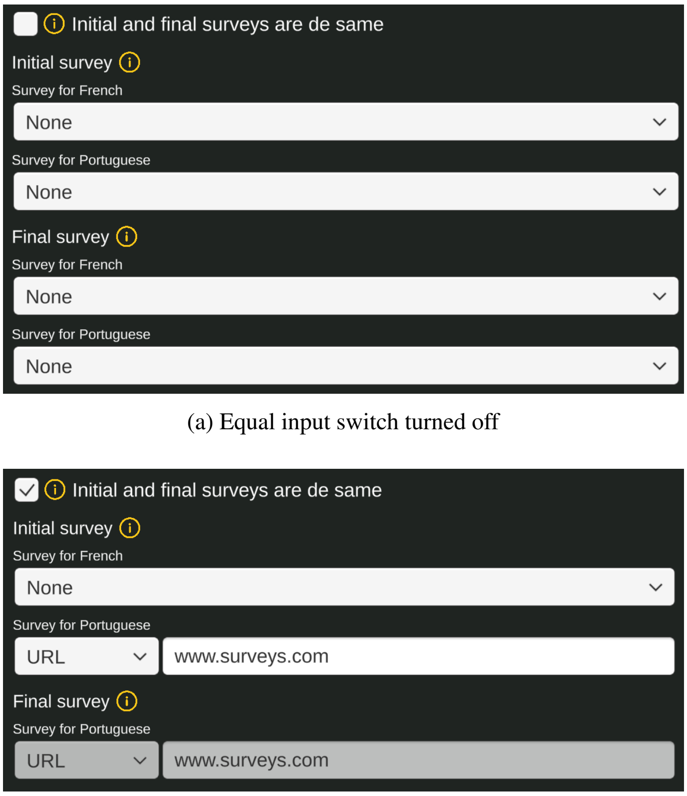
None

Video for Portuguese

None

Figure 4.24: Languages influence in the file or URL input

The equal input switch is also, like with the toggle group, only used in a specific part of the backoffice. Moreover, is used for the surveys because the user may want the final questionary to be the same as the initial, making it easier for the user (see Figure 4.25).



□ ⓘ Initial and final surveys are de same

Initial survey ⓘ

Survey for French: None

Survey for Portuguese: None

Final survey ⓘ

Survey for French: None

Survey for Portuguese: None

(a) Equal input switch turned off

☑ ⓘ Initial and final surveys are de same

Initial survey ⓘ

Survey for French: None

Survey for Portuguese: URL www.surveys.com

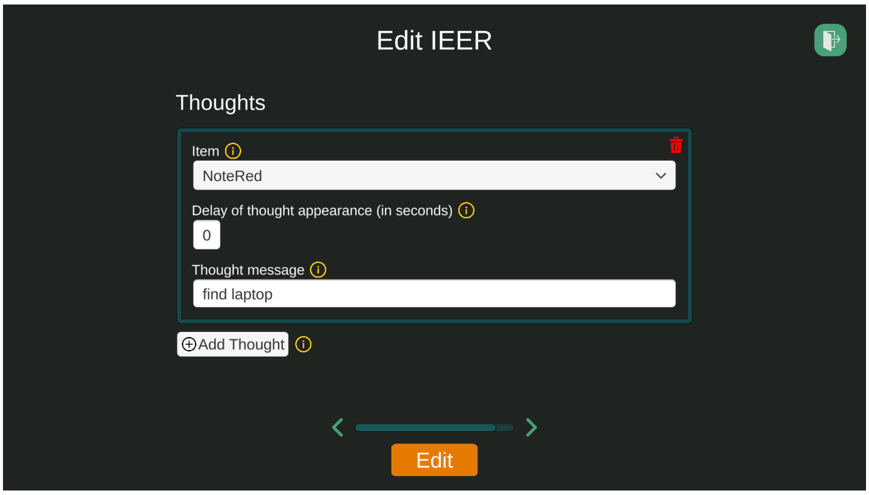
Final survey ⓘ

Survey for Portuguese: URL www.surveys.com

(b) Equal input switch turned on, and with a field filled

Figure 4.25: Equal input switch feature

After the game part, the three last pages appear responsible for the clues, thoughts and interactive objects. All three work similarly, so we will show only the thoughts because it is easier to demonstrate. These pages comprise groups of form fields that can be added and removed. At the top-right corner of each group, a trash button can be found that is responsible for the deletion of that group, and at the end is possible to see a button to add new groups (see Figures 4.26).



Edit IEER

Thoughts

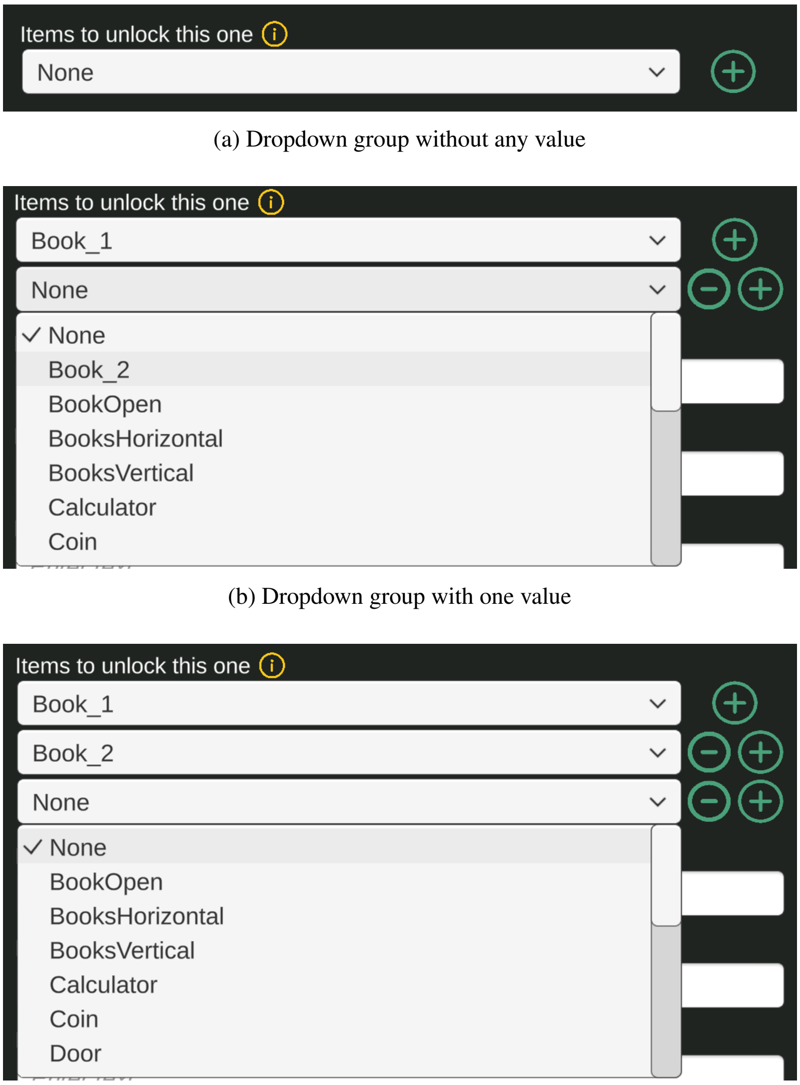
Item ⓘ: NoteRed

Delay of thought appearance (in seconds) ⓘ: 0

Thought message ⓘ: find laptop

Figure 4.26: Page where the thoughts can be managed.

The interactive objects page presents the final form field: dropdown group. The dropdown group works the same way as the multi-text areas. When an option is selected from the dropdown, that option will not be available in the new dropdowns, meaning that if a specific dropdown with a value is removed, that value is readded to the list (see Figure 4.27).


Items to unlock this one: None

(a) Dropdown group without any value

Items to unlock this one: Book_1, None

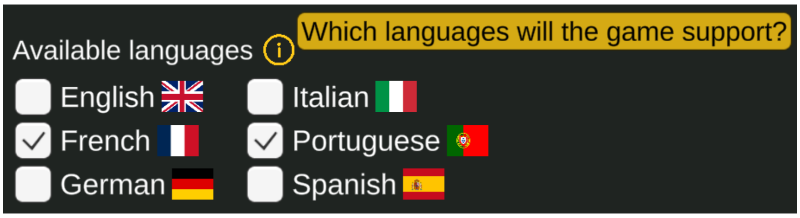
(b) Dropdown group with one value

Items to unlock this one: Book_1, Book_2, None

(c) Dropdown group with two values

Figure 4.27: Dropdown group form field

To conclude this subsection, we demonstrate the informative icon and the popup that appears to the user when leaving the create or edit pages in Figures 4.28 and 4.29, respectively. When the user mouses over this icon, a message appears to explain what is pretended to insert in a specific form field.



Available languages ⓘ

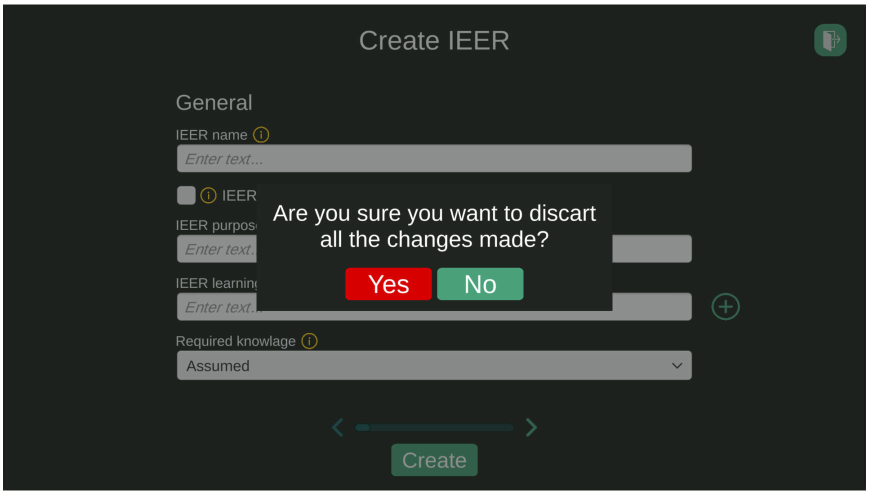
Which languages will the game support?

☐ English ☐ Italian

☑ French ☑ Portuguese

☐ German ☐ Spanish

Figure 4.28: Information that is available in every form field when the user mouses over



Create IEER

General

IEER name ⓘ

Enter text...

☐ ⓘ IEER

IEER purpos

Enter text.

IEER learnin

Enter text.

Required knowlage ⓘ

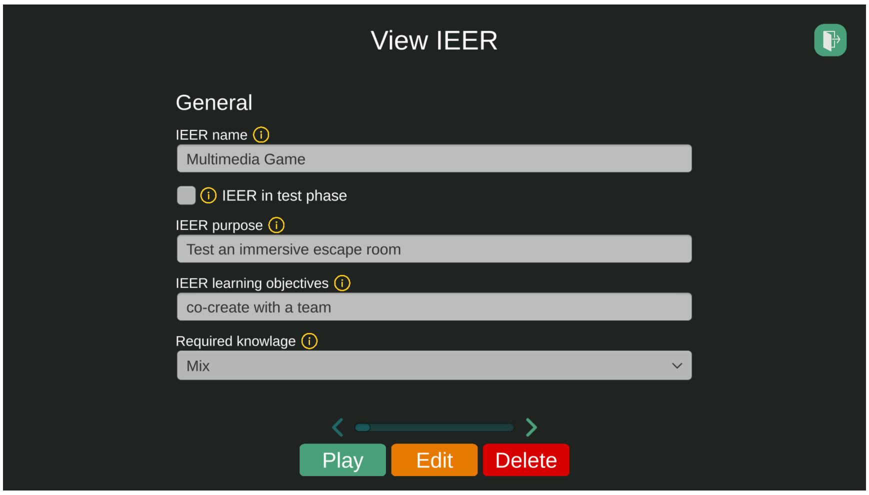
Assumed

Are you sure you want to discart all the changes made?

Figure 4.29: Popup that appears to the user when leaving the create and edit pages

4.8.3 View IEER

View IEER is the page where users can visualise all the information about a specific IEER. This page is like the create and edit pages, except for the title, footer and content area that, in this case, can not be changed by the user. All the form fields are disabled and only show the stored information. At the bottom of the page, three buttons appear: “Play”, “Edit”, and “Delete”. The “Play” takes the user to the game, “Edit” take the user to the “Edit IEER” page, and “Delete” is responsible for deleting the IEER (see the nine following figures).



View IEER

General

IEER name ⓘ

Multimedia Game

□ ⓘ IEER in test phase

IEER purpose ⓘ

Test an immersive escape room

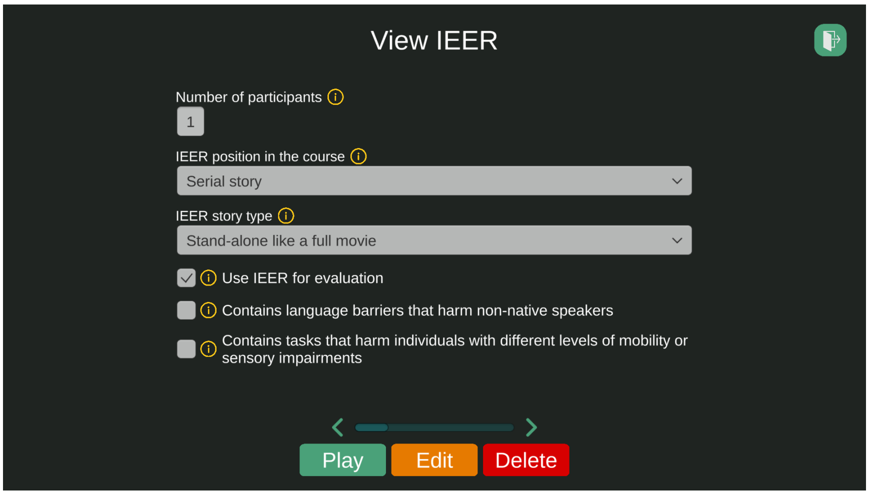
IEER learning objectives ⓘ

co-create with a team

Required knowlage ⓘ

Mix

Figure 4.30: Page 1 of "View IEER"



View IEER

Number of participants ⓘ

1

IEER position in the course ⓘ

Serial story

IEER story type ⓘ

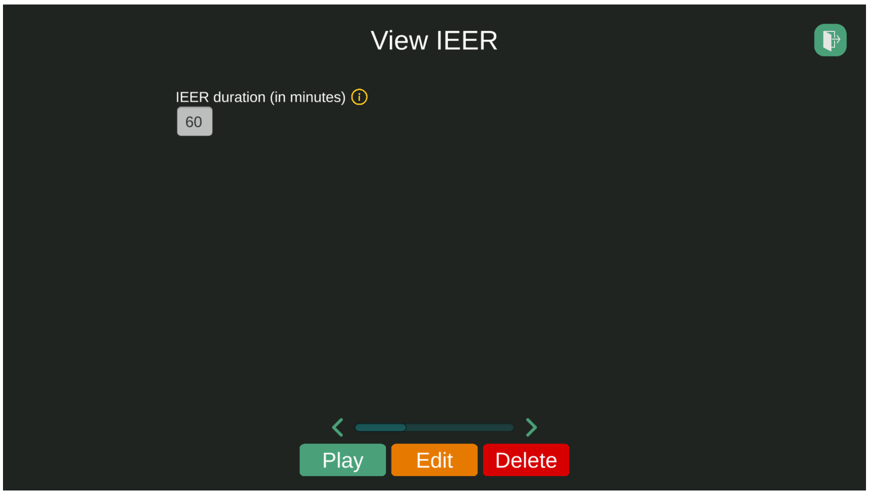
Stand-alone like a full movie

☑ ⓘ Use IEER for evaluation

□ ⓘ Contains language barriers that harm non-native speakers

□ ⓘ Contains tasks that harm individuals with different levels of mobility or sensory impairments

Figure 4.31: Page 2 of "View IEER"

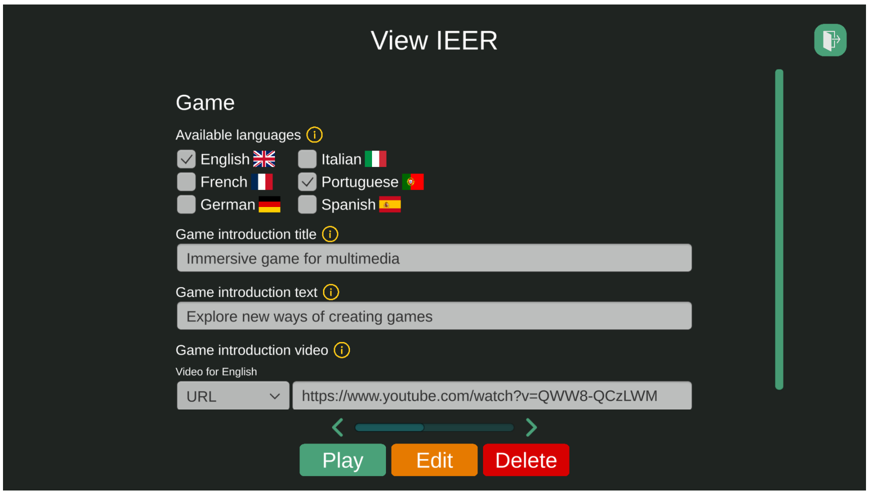


View IEER

IEER duration (in minutes) ⓘ

60

Figure 4.32: Page 3 of "View IEER"


View IEER

Game

Available languages ⓘ

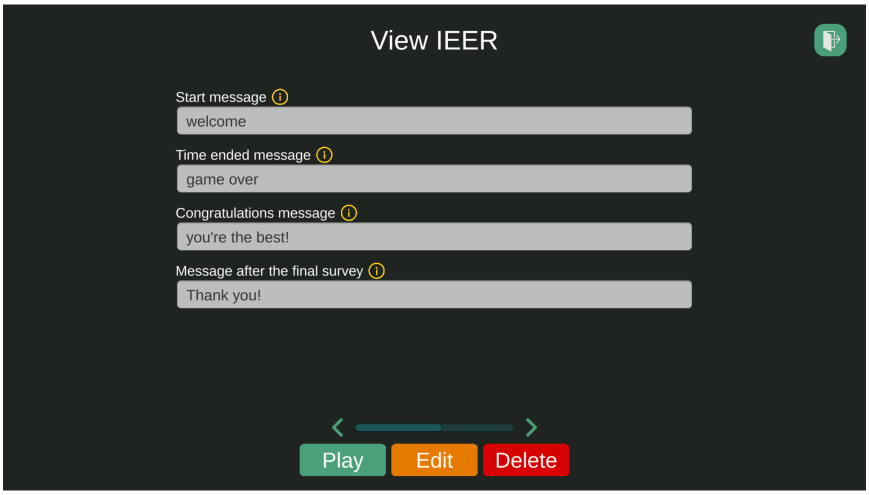
☑ English □ Italian
□ French ☑ Portuguese
□ German □ Spanish

Game introduction title ⓘ
Immersive game for multimedia

Game introduction text ⓘ
Explore new ways of creating games

Game introduction video ⓘ
Video for English
URL https://www.youtube.com/watch?v=QWW8-QCzLWM

Figure 4.33: Page 4 of "View IEER"


View IEER

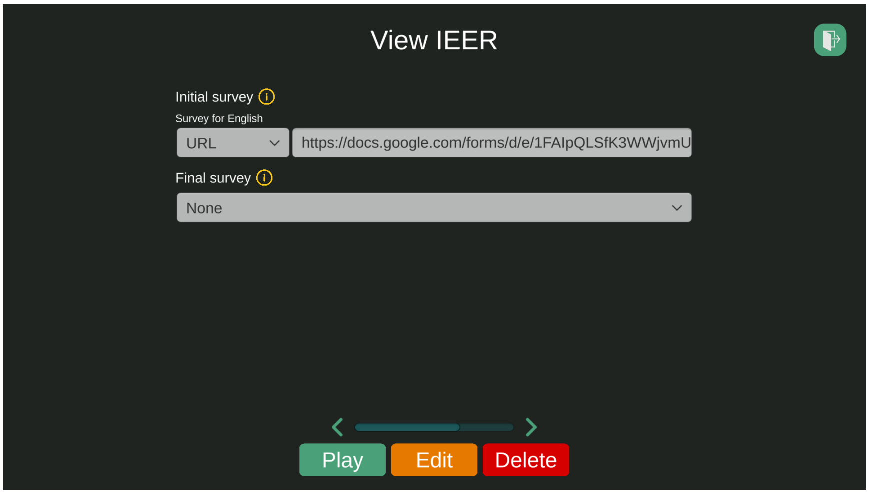
Start message ⓘ
welcome

Time ended message ⓘ
game over

Congratulations message ⓘ
you're the best!

Message after the final survey ⓘ
Thank you!

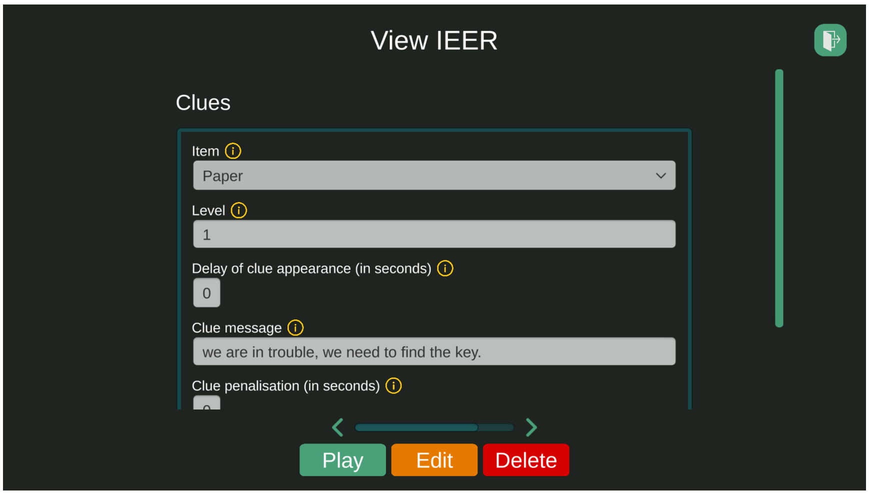
Figure 4.34: Page 5 of "View IEER"


View IEER

Initial survey ⓘ
Survey for English
URL https://docs.google.com/forms/d/e/1FAIpQLSfK3WWjvmU

Final survey ⓘ
None

Figure 4.35: Page 6 of "View IEER"



Clues

Item: Paper

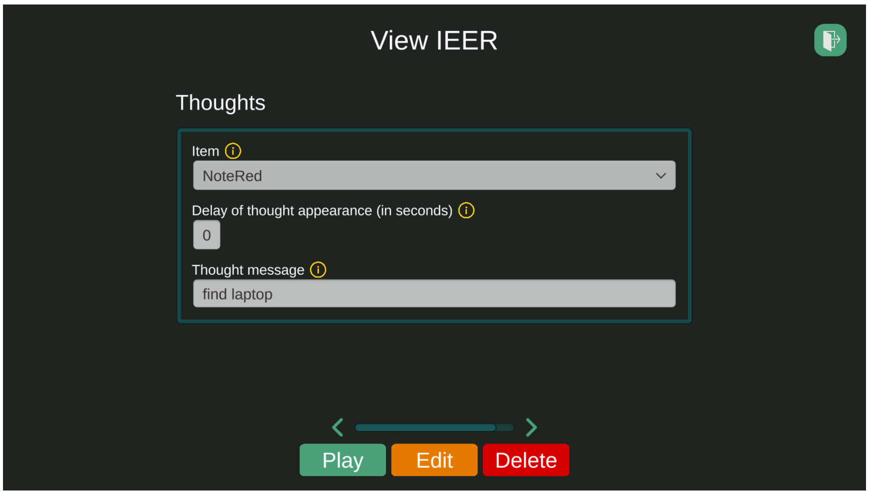
Level: 1

Delay of clue appearance (in seconds): 0

Clue message: we are in trouble, we need to find the key.

Clue penalisation (in seconds): 0

Figure 4.36: Page 7 of "View IEER"



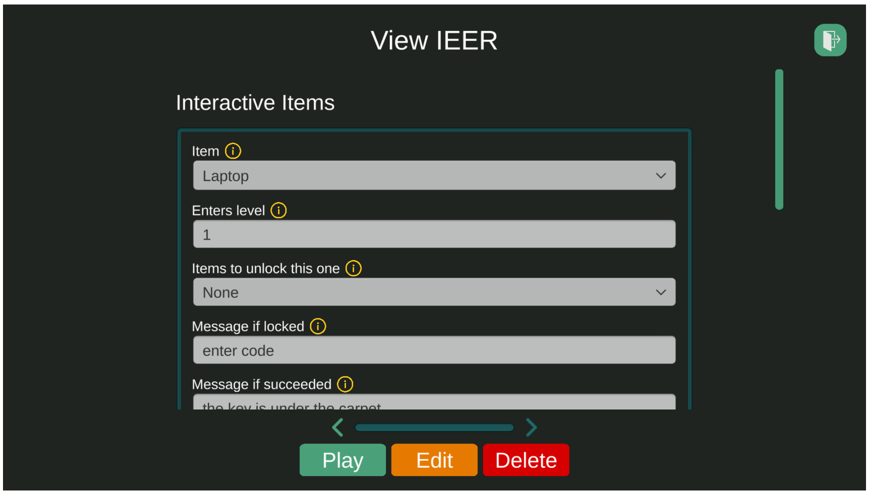
Thoughts

Item: NoteRed

Delay of thought appearance (in seconds): 0

Thought message: find laptop

Figure 4.37: Page 8 of "View IEER"



Interactive Items

Item: Laptop

Enters level: 1

Items to unlock this one: None

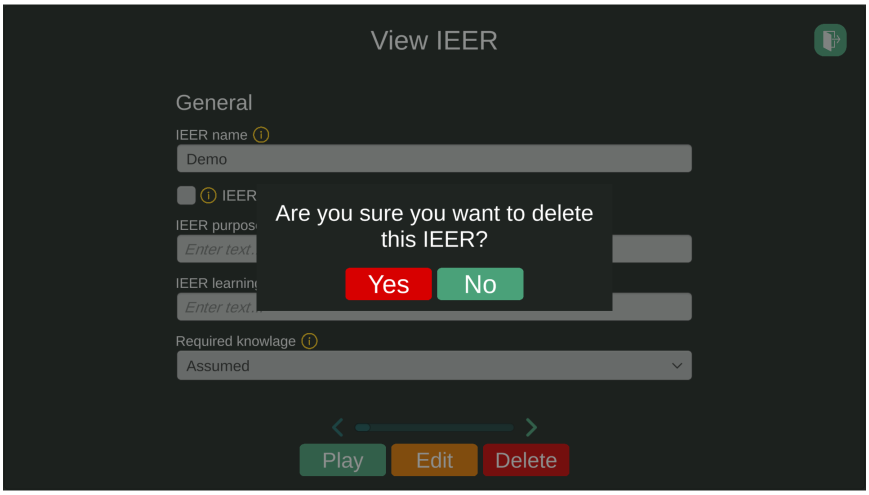
Message if locked: enter code

Message if succeeded: the key is under the carpet

Figure 4.38: Page 9 of "View IEER"

4.8.4 Delete IEER

As mentioned, “Delete” button deletes an IEER the user selects. Figure 4.39 shows that when clicking “Delete”, a popup shows up to confirm if the user wants to delete that specific IEER.



View IEER

General

IEER name ⓘ

Demo

□ ⓘ IEER

IEER purpos

Enter text.

IEER learnin

Enter text.

Are you sure you want to delete this IEER?

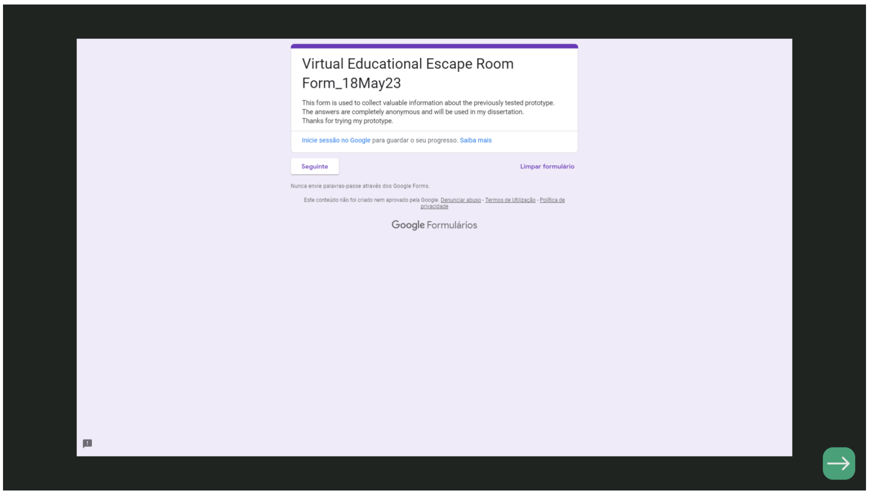
Required knowlage ⓘ

Assumed

Figure 4.39: Popup that appears to the user when trying to delete an IEER

4.8.5 Webview

Using the file or URL input information for the videos was simple because the video player already existed. At the same time, a new solution was needed for the surveys and minigames. This solution involves using a webview to display the web page for the URL inserted by the user. This feature does not only allow the user to see the web page but also to interact with it (see Figure 4.40).



Virtual Educational Escape Room Form_18May23

This form is used to collect valuable information about the previously tested prototype.
The answers are completely anonymous and will be used in my dissertation.
Thanks for trying my prototype.

Inicie sessão no Google para guardar o seu progresso. Saiba mais

Nunca envie palavras-passe através dos Google Forms.

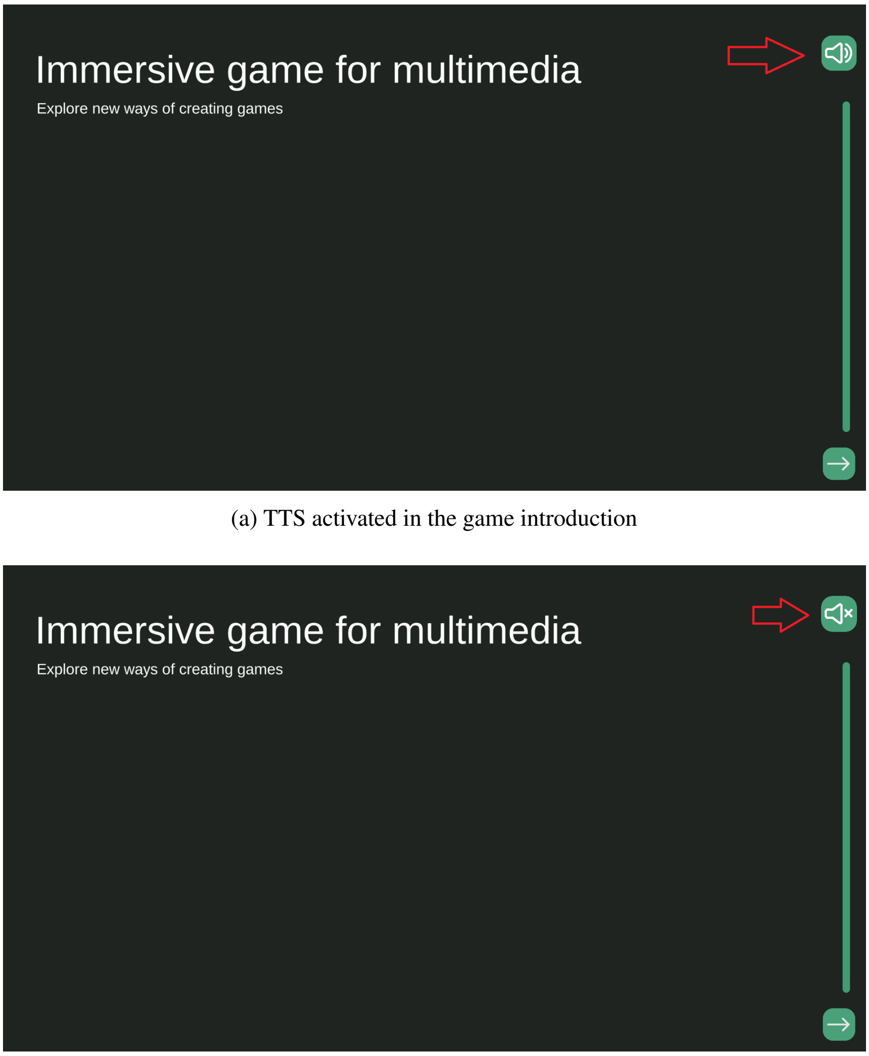
Este conteúdo não foi criado nem aprovado pela Google. Denunciar abuso - Termos de Utilização - Política de privacidade

Google Formulários

Figure 4.40: Webview of the survey URL inserted by the user

4.8.6 TTS

The TTS feature was added to the game to help players with visual difficulties. We can use the TTS to read that content by having the text and the language we want. In Figure 4.41, it is possible to see two images of the TTS in the game introduction, but it is available in other parts of the game.



Immersive game for multimedia

Explore new ways of creating games

(a) TTS activated in the game introduction

Immersive game for multimedia

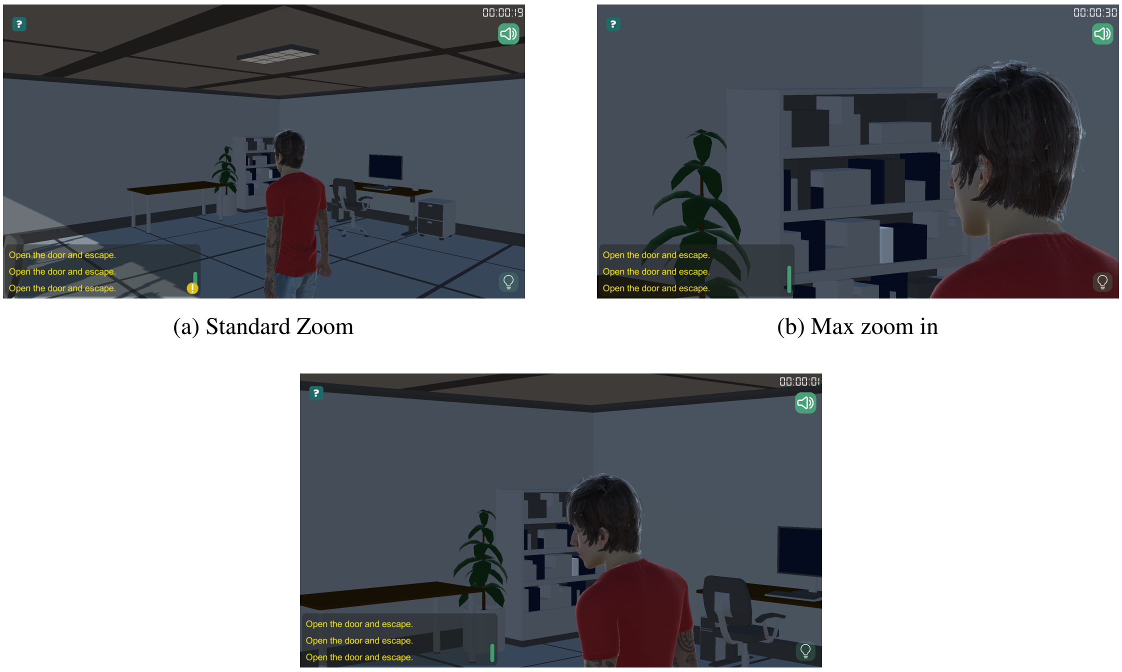
Explore new ways of creating games

(b) TTS deactivated in the game introduction

Figure 4.41: TTS feature in the game introduction

4.8.7 Zoom

In the preceding subsection, we discussed the TTS functionality. Now, let's examine the Zoom feature explicitly designed to assist visually impaired users. This feature is accessible exclusively during gameplay, particularly when exploring the room, allowing users to zoom in or out. By utilising the mouse wheel, users can easily regulate the zoom level: rotating it forward results in zooming in, while rotating it backwards allows for zooming out (see Figure 4.42).



Open the door and escape.
Open the door and escape.
Open the door and escape.

Open the door and escape.
Open the door and escape.
Open the door and escape.

(a) Standard Zoom

(b) Max zoom in

Open the door and escape.
Open the door and escape.
Open the door and escape.

(c) Zoom between standard and max zoom

Figure 4.42: Zoom behaviour

4.9 Summary

Focusing on JSON-based ontology, we defined learning paths for subsequent conversion into IEERs. The backoffice retrieves user-provided information and transforms it into the selected ontology. To construct this prototype, we referred to two sources: a dissertation by Santos (2022), who developed an ontology and explored IEERs, and an article by Fotaris and Mastoras (2022) on design thinking principles in EER creation.

To avoid using different technologies, we developed a backoffice for manually defining learning paths within the Unity3D environment, leveraging the existing game. Before planning the framework/tool/application, it was crucial to start with initial drafts to understand the client's requirements. We created a wireframe for the backoffice, outlining the possible user paths within the application. Once the desired backoffice wireframe was achieved, we designed input fields based on the ontology.

Some code had already been developed by two colleagues who recently completed their master's in Informatics and Computing Engineering. Therefore, we obtained the required Unity Editor version, cloned the GitLab repository, and ensured the project opened and ran smoothly. We carefully analysed the existing code to understand its functionality and determine how our new tool would fit.

We designed an initial panel scene for backoffice operations, featuring elements such as a page title input area, a homepage navigation button, a content area for form fields, horizontal and vertical scrollbars, a progress bar, buttons linked to the horizontal scroll, and a footer area for button placement. Subsequently, we created prefabs for primary inputs based on our drafts, including text, multi-text, dropdown, integer, decimal, and toggle fields. We also developed scripts to manage these inputs, allowing for reading, clearing, resetting, setting input field values and granting or denying user interaction. We created an abstract class for input fields to streamline the process, enabling easy access to the same methods across different input types. Combining the existing code with the newly designed input fields, we constructed a backoffice panel and connected it to the EdScape homepage.

Next, we created and integrated additional form fields, such as toggle groups, file or URL inputs, dropdown groups, and minigame inputs. Due to file system policies and user convenience, only URL insertion was implemented in the current version, as it proved more accessible and practical than file upload. To accommodate the selection of identical inputs, we developed a prefab for the "EqualInputSwitch" based on the toggle input.

Developing all the simple form fields for the backoffice simplified the creation of composed form fields, including clues, thoughts, and interactive objects. We created an informative icon prefab to provide users with clear instructions for each form field. When hovering over the icon, a text box appears, explaining the purpose of the corresponding form field.

Some features of this project relied on external resources. The translation of game content was facilitated using the Google Translate API. The search for YouTube video streaming URLs involved the "YT-API" from RapidAPI. Accessing a web view of the user-inserted URL was possible using GitHub's open-source "unity_browser" code. After stabilising all the code, we focused on updating the colour palette of the developed prefabs to match EdScape's visual style.

When developing an educational video game, it is crucial to consider the end user's experience, ensuring the game is accessible and enjoyable for everyone, including individuals with impairments. Two key features were implemented to address this: TTS, utilising the Google Translate TTS API to assist users with visual impairments or difficulty reading small text, and a Zoom feature to aid visually impaired users.

Initially, we planned to use PCG for room generation and object placement. However, due to time constraints, we opted for an algorithm that randomly assigns positions and rotations to interactive objects. Although this solution addressed the issue, it could be improved to allow for the precise placement of objects with similar requirements. Ensuring both entities can be interacted with is crucial for a positive user experience and for avoiding monotonous gameplay.

With all the features mentioned earlier implemented, we finally achieved a functional prototype ready for further user testing.

Chapter 5

Results and Evaluation

This chapter shows the results obtained in an experimental study with a focus group, followed by its evaluation, limitations and contributions.

5.1 Introduction

Results and evaluation is the chapter where we can evaluate the work carried out, validate the prototype with the users through studies and present the limitations and contributions obtained.

In this chapter, we present an experimental study done with a focus group, describing it and analysing the results gathered from it, showing graphics with the results with a brief analysis. Lastly, we present the limitations and contributions observed.

In the last section of this chapter, we summarise the most vital aspects retained from the results and evaluation of this virtual environment's authoring tool.

5.2 Focus Group

In this research type, an essential step of the DBR methodology is when we contact the users. Therefore, we can get important feedback from them, knowing what needs to be fixed or improved, new features that could be implemented and insights into what would be necessary to reach the perfect product. This section describes the focus group and presents and analyses the results.

5.2.1 Description

The performed study occurred in room B302 of the Faculty of Engineering of the University of Porto, with the participation of a focus group composed of eight education specialists (teachers and trainers) who frequent the subject of Educational Software from the master's degree in Multimedia. This study aimed to test the IEER managing tool and the new accessibility and adaptability features. So, one by one, the eight participants tested the application and played a more complex IEER to enjoy the gameplay and see how far the application could go. Moreover, they were asked to fill out a short formulary with questions about the developed features.

The formulary was divided into four groups: “About you”, “IEER managing tool”, “Accessibility and adaptability”, and “Thank you!”. First, we have the questions to know a little about the user and characterise the gathered data. Further, the questions about the developed features appear to understand the user’s opinion. And finally, we have the acknowledgement part, where we ask if any problem occurred and what can be improved. The questions of each formulary’s group were as follows:

- “About you”
  - Date of birth
  - Sex
  - Have you ever built an IEER (Immersive Educational Escape Room)?
  - What is your experience with IEERs?
- “IEER managing tool”
  - How easy was it to create an IEER?
  - What was it like to create an IEER?
  - Were all the input fields clear?
  - When an input field was unclear, was the info button able to help?
  - An external API is being used to generate IEER content according to the selected languages. Is this a helpful approach?
  - How was your experience using the IEER management options (create, view, edit and delete)?
- “Accessibility and adaptability”
  - For a person who has never created an IEER, how accessible is this IEER managing tool?
  - How do you classify TTS (text-to-speech) functionality?
  - How do you classify Zoom functionality?
- “Thank you!”
  - Did you find any problems during the test?
  - If yes, describe them.
  - What could we improve?

5.2.2 Data Analysis and Results

In this section, we analyse the gathered data by group. Starting with the “About you” and finishing with the “Thank you!”. We show the graphics with the results and then explain what information can be extracted from them.

By analysing the results, our participants are between 22 and 34 years old, being that half of them have 22 years of age, so we can consider that our sample is mainly composed of young adults. They were mostly Female. Out of eight participants, five were Female, two were Male and one preferred not to say.

Most of our participants had never built an IEER before, as seven responded that they had never done so, while one answered the opposite. So, for most of them, this study was the first time they built and played an IEER.

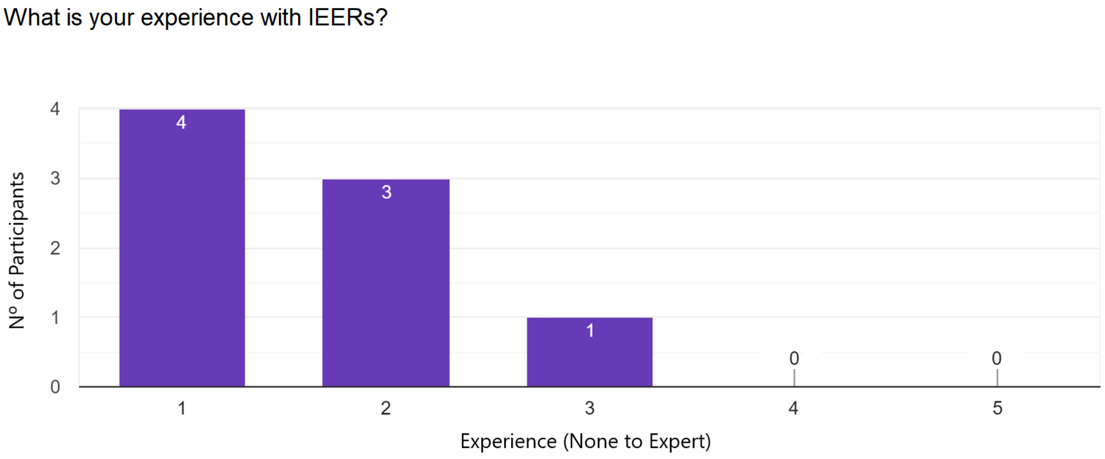


Figure 5.1: Question 4 of the form

Our participants have little or no experience with IEERs, whereas four assumed they have no experience, three have minimal experience, and one considered having some experience with IEERs. Therefore, most participants have never been in touch with IEERs.

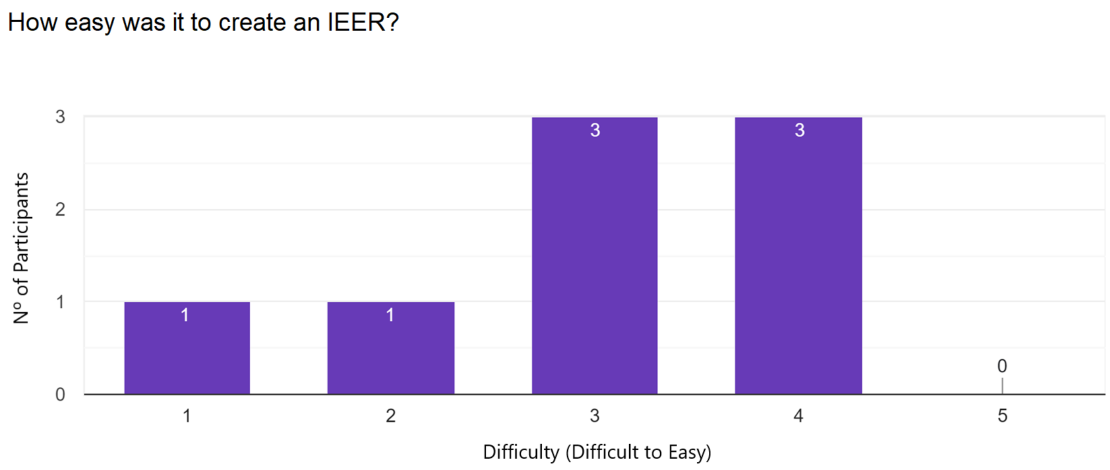


Figure 5.2: Question 5 of the form

With the results gathered, most participants considered creating an IEER straightforward since six answered the second and third highest answers. The remaining two got distributed evenly below the middle option. Considering this, the IEER creation is accessible but has some aspects to improve.

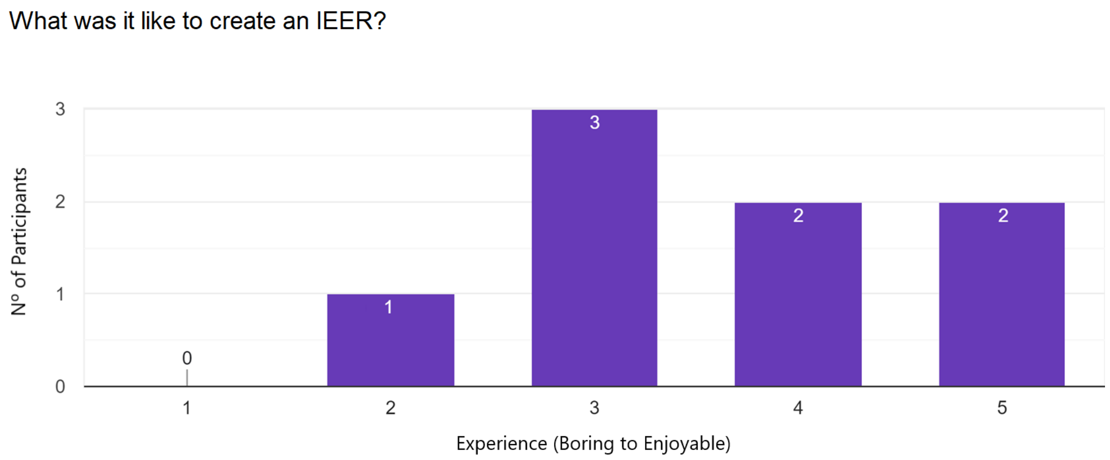


Figure 5.3: Question 6 of the form

The results indicate that two participants considered their experience with IEER creation the highest answer possible and five the second and third highest answers. The remaining one considered the IEER creation a little boring. Therefore, we can assume that the experience of creating an IEER is acceptable and may not bore users.

By analysing the results, six participants assumed that what was intended to insert in each input field was evident, while two disagreed. Despite most participants agreeing with the clearness of the input fields, we consider that some improvements must be made, so all the users can clearly understand what was intended for each input field. We can assume that the information button was handy for the users when the input field was unclear since all participants agreed that this feature helped them.

In the question where we mentioned if using an external API brought any advantage to the user, all participants answered “Yes”. Overall, they liked this approach for translating the game content instead of being them to write their different language versions.

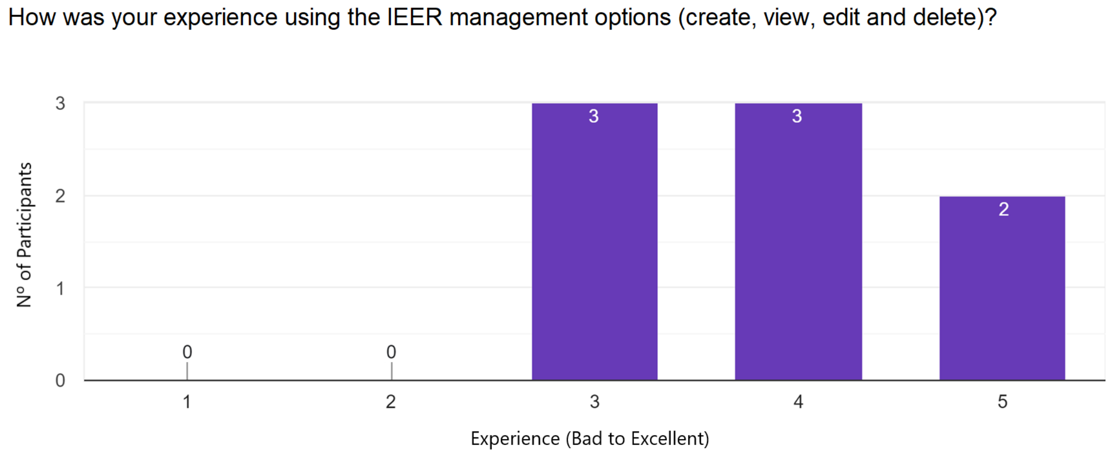


Figure 5.4: Question 10 of the form

All the participants considered that the IEER management options experience was good, where two participants answered the highest option and the remaining six were equally distributed between the second and third highest options.

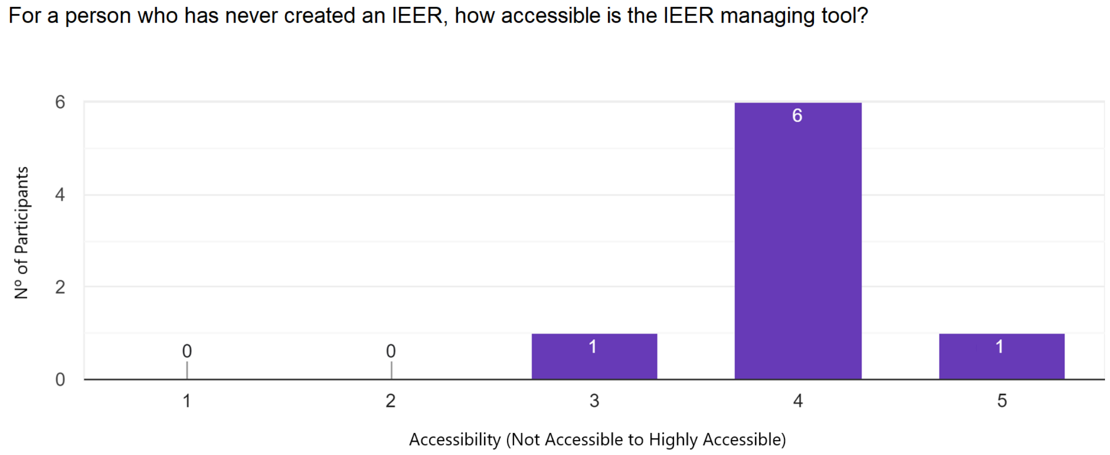


Figure 5.5: Question 11 of the form

The results indicate that six participants scored the IEER managing tool, with the second highest option, one considered highly accessible, and the remaining one answered the third highest option. So, we can assume that our IEER managing tool is accessible to users who never have been in contact with similar applications.

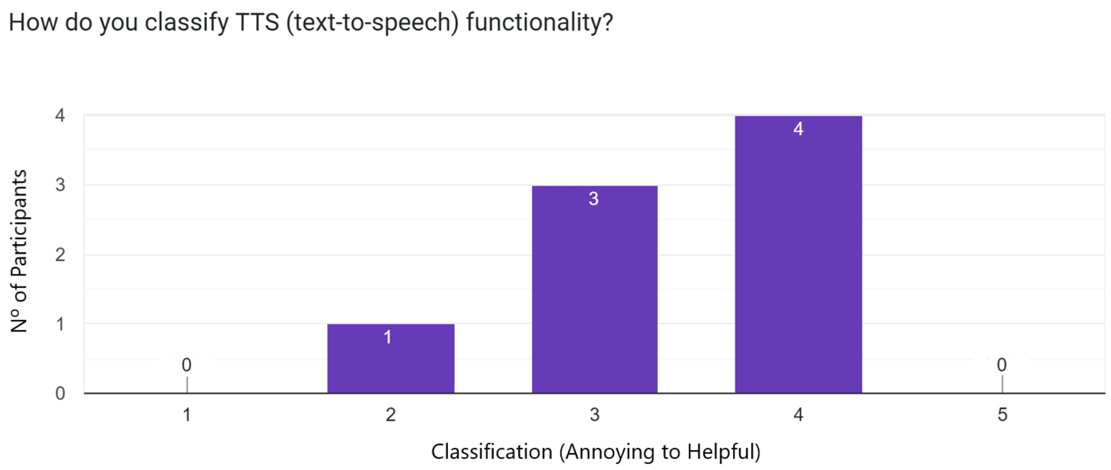


Figure 5.6: Question 12 of the form

Regarding the TTS functionality, seven of the participants answered the second and third highest options, while one did not consider this feature helpful. Considering these results, we can assume that the TTS is a valuable feature.

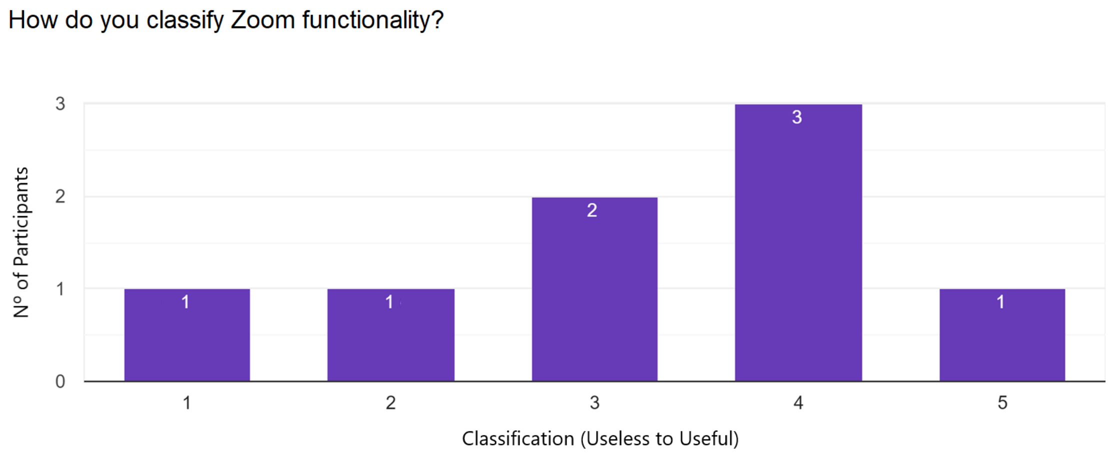
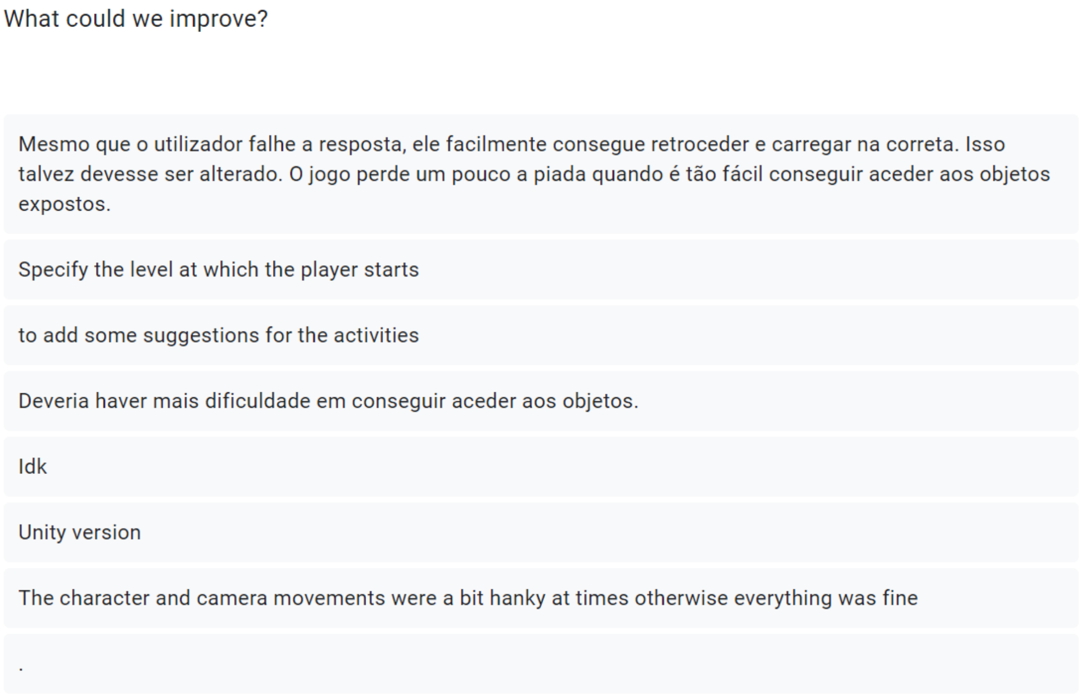


Figure 5.7: Question 13 of the form

The results are more distributed for the Zoom feature. Considering the Zoom functionality helpful, one answered the highest option, three answered the second highest option and two the third best option. The remaining two were equally distributed in the lowest options, considering the

Zoom feature was useless. Considering this result, the Zoom feature does not add much value to the application but can be helpful in certain situations.

Finally, we have the last three questions, where we acknowledge the participants for their participation and ask if something went wrong and what could be improved. In this question, all the participants answered that no problems were found during the experiment.



What could we improve?

Mesmo que o utilizador falhe a resposta, ele facilmente consegue retroceder e carregar na correta. Isso talvez devesse ser alterado. O jogo perde um pouco a piada quando é tão fácil conseguir aceder aos objetos expostos.

Specify the level at which the player starts

to add some suggestions for the activities

Deveria haver mais dificuldade em conseguir aceder aos objetos.

Idk

Unity version

The character and camera movements were a bit hanky at times otherwise everything was fine

.

Figure 5.8: Question 16 of the form

Last but not least, we have the improvements suggested by the participants. From these answers, we can retain the following points: (1) The minigame was always the same even if the user failed the first time, leading to memorising the answer position. The object placement was not good since the objects were too easy to find; (2) The game level was not visible to the user, and when creating an IEER, the user is asked to select the level for the clues and interactive objects. In that specific input field is not specified in each level the game starts, and probably should be an integer input and not a text input; (3) The character and camera movements should be improved.

Overall, we think the focus group went well, there were no problems, and the participants were friendly and accessible. With this study, we gathered important information about the project's current state and what needs to be improved.

5.3 Limitations and Contributions

Several limitations were detected and noted throughout the project development and with the help of the focus group, such as: (1) The file upload system works for the standalone environment, but working with files is not that simple, so this option was disabled and is not accessible by the user. Although this functionality works in the standalone environment, the WebGL environment does not; (2) Reproducing other videos out of Youbute is impossible in the current application version; (3) In the file or URL input, if a value is assigned to a field, for example, the one destinated to the Portuguese language, that field value becomes invisible while the Portuguese language is deselected. So, it requires additional work for the user to keep that value; (4) Show the item images in the dropdowns, allowing the user to see the items instead of just seeing their identifier; (5) It is possible to create empty clues, thoughts and interactive objects, although they will not be used; (6) The data from the new inputs related to the SMART questions are not being used; (7) The number of participants is disabled because the multiplier feature is not implemented yet; (8) The application should work in a standalone and WebGL environment. However, due to a building issue, it was impossible to deploy a WebGL version; (9) In the case of webview, it is impossible to know if the user has already finished the survey or the minigame. An option is always available to proceed in these cases; (10) TTS plays log messages when the log window is not visible. This happens because the game sends messages to the log window even if it is not visible, which should not occur. Fixing this issue will consequently fix the TTS issue.

5.4 Summary

In this DBR methodology, the focus group was crucial. As a result, we can obtain valuable feedback from them, understand what needs to be fixed or improved, identify new features that could be implemented, and gain insights into what would be necessary to achieve the perfect product.

The focus group took place at the Faculty of Engineering of the University of Porto, involving eight education specialists attending Educational Software for the master's degree in Multimedia. This study aimed to test the IEER management tool and the new accessibility and adaptability features. Additionally, the participants were asked to complete a brief questionnaire regarding the developed components.

The questionnaire was divided into four sections: "About You," "IEER Management Tool," "Accessibility and Adaptability," and "Thank You!". From this questionnaire, several assumptions were derived: (1) Our participants are aged between 22 and 34 years; (2) The majority of our participants were Female; (3) Most of our participants have never built an IEER before; (4) The majority of participants have had no prior experience with IEERs; (5) The process of creating an IEER is accessible but requires some improvements; (6) The experience of creating an IEER is acceptable and not likely to bore users; (7) Certain enhancements must be made to ensure that all users can clearly understand the intended purpose of each input field; (8) The information button proved helpful for users when the input field was unclear; (9) The approach of translating the game content, rather than having users write their versions in different languages, was well received; (10) The experience of managing IEER options was considered good; (11) Our IEER management tool is accessible to users who have never interacted with similar applications; (12) The Text-to-Speech

(TTS) feature is highly valuable; (13) The Zoom feature does not contribute significantly to the application but can be helpful in specific situations.

Overall, the study was successful, with no issues encountered, and the participants were friendly and approachable. Through this study, we gained important insights into the project's current state and identified areas for improvement. Several limitations were detected and documented throughout the project development and with the assistance of the focus group.

Chapter 6

Conclusions and Future Work

This chapter presents the conclusions that have been drawn from this work and propose some future topics that would be relevant to explore.

6.1 Conclusions

New technologies and games have improved students' learning processes in educational settings. We have seen that all the topics discussed before can enhance learning in very different ways, awakening students' interest and making it more enjoyable.

After analysing the contents of various documents and their authors' statements, we can conclude that games are an effective way of teaching, encouraging students to be motivated to learn, making them prefer this learning method over traditional approaches. Students' success in a learning environment depends on teachers' engagement. Combining cognitive, affective, and sociocultural perspectives is crucial for game design and research to understand the importance of relevant game features for learning.

Further research is needed in immersive learning environments due to relevant gaps. VR creates new experiences empowering the subject's memory and accelerating their learning process. Still, when poorly implemented, VR causes some inconveniences, such as motion sickness, cybersickness and simulator sickness.

PCG is an excellent game content generator because it generates customized content every time. However, this approach has some drawbacks, including the need for human intervention to monitor the results. The generated content can be improved when Quality-Diversity algorithms are integrated with PCG. The three aesthetic consequences of PCG use in games that commonly occur are Challenge, Discovery, and Fellowship. Various PCG-G techniques have been used to generate several works over time. Each approach has benefits and drawbacks that other authors have classified differently. Responding to beautiful landscapes, enhancing developing techniques, and practising in various settings are the three main actions that increase replayability. A hybrid PCG algorithm may produce better game content than a traditional one. A detailed analysis of unique algorithms, systems, and game design will be necessary for making PCG-based games.

PCG-G is mainly used in educational and specialised video games, such as rogue-like games, and has not yet been extensively used in developing game systems and scenarios.

In this context, we have found that PCG is a powerful tool for generating custom game content, depending on student preferences and abilities, and improving it considering student performance. The UX of educational games may offer vital information on boosting students' learning by recognising different player groups and their features. Based on these characteristics, designers might modify games and develop player models that catch students' interest.

PCG may be used to position items in a gaming scene as well. Various PCG methods were provided based on the analysis that swiftly builds room interiors without designers spending much time on them. PCG has once again shown to be a potent tool for game creation by being able to construct game environments and automatically put objects for us.

Considering what has been presented in Chapter 2, PCG is an area that has been relatively underexplored in IEERs. However, it can significantly impact students' learning process because it combines several factors that enhance learning.

The dissertation was developed using a DBR methodology, beginning with problem analysis, solution development, and iterative solution testing and refinement cycles, resulting in tangible and intangible outcomes. As part of the iterative process, research subjects are essential in refining and reworking the solution to reach a good product. Due to its cyclical nature, research may be structured into cycles that allow testing new tools and teaching methods while improving final goods and procedures.

DBR has won recognition for its capacity to actively incorporate fieldwork participants (traditionally treated as research subjects) into teaching settings. The DBR data show when, why, and how the educational intervention is growing, improving results, and encouraging innovations based on data in close collaboration with participants, researchers, and designers. Moreover, DBR recognises that environmental circumstances might affect how an educational intervention is carried out and its results.

We established learning paths with a JSON-based ontology in mind to later transform into IEERs. The backoffice gets data from the user and modifies it to fit the chosen ontology. We used two sources to create this prototype: a dissertation by Santos (2022), who made an ontology and studied IEERs, and an article by Fotaris and Mastoras (2022), who discussed the application of design thinking concepts to the development of EERs.

We created a backoffice for manually constructing learning courses within the Unity3D environment, utilising the already-existing game to avoid using different technologies. It was essential first to grasp the client's needs to plan the framework, tool, or application correctly. For the backoffice, we drew a wireframe describing potential user pathways inside the programme. We developed input fields based on the ontology once the necessary backoffice wireframe had been realised.

Two coworkers who had recently obtained their master's degrees in Informatics and Computer Engineering had already written some code. As a result, we cloned the GitLab repository, acquired the necessary Unity Editor version, and ensured the project opened and operated without a hitch.

To comprehend the operation of the current code and ascertain how our new tool would fit, we thoroughly examined it.

A page title input area, a homepage navigation button, a content area for form fields, horizontal and vertical scrollbars, a progress bar, buttons linked to the horizontal scroll, and a footer area for button placement are some of the components of the initial panel scene we created for backoffice operations. Then, using our draughts as a guide, we produced prefabs for primary inputs such as text, multi-text, dropdown, integer, decimal, and toggle fields. We also created scripts to read, clear, reset, set input field values, and accept or reject user activity to manage these inputs. To speed up the procedure, we developed an abstract class for input fields that makes it simple to access the same functions regardless of the kind of input. We built a backoffice panel and connected it to the EdScape homepage by fusing the preexisting code with the newly created input fields.

Then, we developed and included other form fields, including toggle groups, file or URL inputs, dropdown groups, and minigame inputs. Only URL insertion was included in the present version due to file system regulations and user convenience because it was more accessible and valuable than file upload. We created a prefab for the "EqualInputSwitch" based on the toggle input to allow for the selection of identical inputs.

The development of all the specific form fields for the backoffice made it easier to create composite form fields that included clues, thoughts, and interactive objects. We designed a helpful icon prefab to give customers precise explanations for each form field. When the user mouse over the icon, a text bubble describing the function of the associated form field displays.

This project's features sometimes depended on other resources. The Google Translate API was used to make it easier to translate game information. The "YT-API" from RapidAPI was used to look for YouTube video streaming URLs. The open-source "unity_browser" code from GitHub also enabled viewing the user-inserted URL's webview. After stabilising all the code, we concentrated on altering the colour scheme of the created prefabs to fit EdScape's visual aesthetic.

The player's experience must be considered while creating a didactic video game to ensure everyone, including those with disabilities, can play and enjoy it. Two essential features were added to solve this: TTS, which uses the Google Translate TTS API to help users with visual impairments or trouble seeing tiny text, and a Zoom option for visually impaired users.

Initially, we intended to utilise PCG to create rooms and position objects. Due to time restrictions, we used an algorithm that randomly distributes interactive objects' locations and rotations. Even though this approach solved the problem, it might be enhanced to enable the accurate positioning of items with comparable needs. Making sure both entities may be interacted with is essential for a satisfying user experience and to prevent games from becoming repetitive.

After including all the previously indicated features, we produced a workable prototype suitable for additional user testing.

A critical component of the DBR process is the focus group. By asking people for their opinion, we may learn what needs to be changed or improved, find new features that might be included, and understand what would be required to create the ideal product.

Eight education professionals enrolled in the Multimedia master's programme at the University of Porto's Faculty of Engineering participated in the focus group. The new accessibility and adaptation features and the IEER management tool were tested in this focus group. In addition, a quick questionnaire about the created components was given to the participants.

There were four sections on the questionnaire: "About You," "IEER Management Tool," "Accessibility and Adaptability," and "Thank You!". Several conclusions were drawn from this questionnaire: (1) Our participants are primarily Female and range in age from 22 to 34. Most of them did not know IEERs and had never constructed one; (2) Although it needed some work, creating an IEER was doable, acceptable, and unlikely to tyre consumers. Specific improvements must be made to guarantee that all users can comprehend each input field's intended usage. However, consumers found the information button helpful when the input field was not clear. Instead of letting players create their versions of the game in other languages, the strategy of translating the game's content was highly appreciated. It appears that managing IEER options went well; (3) Our IEER management solution was usable by people who had never used anything like it before. The TTS function is quite beneficial. Although it does not dramatically improve the programme, the Zoom capability might be helpful in some circumstances.

The focus group was practical. Overall, there were no problems, and the participants were friendly and approachable. We discovered possibilities for development through this investigation and obtained crucial insights into the project's present situation. Several constraints were found and recorded throughout the project's growth and with the help of the focus group.

With the information collected throughout the dissertation, we can now answer the research questions that were proposed in Chapter 1:

- How can a teacher's learning path be translated into a simple graph?
  - The teacher's learning paths can be transformed into a simple graph using a tool like the one developed for this dissertation. Where the teacher can insert every detail of his intended learning path, and the tool does the work. This information is collected using a simple form, stored internally in classes and then in JSON files as a simple graph.
- How to generate an IEER environment from a simple graph?
  - As previously mentioned, the learning path information is stored in a JSON file as a simple graph. Therefore, the application only needs to load that information to the respective classes the game will use, for example, to place objects.
- What is the diversity of IEER generated by PCG?
  - The learning path information and the PCG technology can generate different IEERs. PCG is a game content generator that works on demand, meaning that if we want fresh content, it is only necessary to ask for it. So, every time the user clicks on the play button of a random IEER, the application sends the necessary data to the PCG, and it generates an IEER different from the ones played before for the same learning path.

In short, considering what was mentioned in all previous sections, this dissertation theme is worth exploring since educational games that group VR, PCG and EERs are currently unknown. Moreover, this combination can positively impact education, particularly in student learning.

6.2 Future Work

In this section, we discuss the improvements needed to reach the desired final product, where teachers can introduce their learning paths into the application, and their students can play it, having an enjoyable and enriching experience while playing. Consult their students' performance and have access to other data registered during the gameplay, aiming for a future where learning/teaching can be more attractive for students and teachers. So, we list below a few improvements that should be made: (1) Fix the problems that do not allow the deployment of the game to the WebGL version; (2) Search for a better alternative or improve the existing webview. It is impossible to know when the user completes the survey or minigame. So, the user needs to click on a button to proceed to the next page; (3) Improve the backoffice. Information on the file or URL inputs is temporarily lost when the user changes the available languages because the input areas appear and disappear when the available languages are changed. Show images instead of text when the user is selecting an item. Creating empty clues, thoughts, and interactive objects should not be possible. Set the required fields necessary for a simple IEER to work as mandatory; (4) Fix the log messages because it is affecting the TTS feature. The game sends log messages when the user can not see this log. Consequently, the TTS always plays that text, meaning that we can hear the TTS and not see that specific text; (5) Study and analyse if the input of files for videos, surveys, or minigames would somehow be feasible or discard this option; (6) Search for a better solution than Google Translate API for the text content, aiming for better quality translations; (7) When creating an IEER, allow users to input links to videos on websites other than YouTube; (8) Take advantage of the new information collected in the form, based on the SMART questions, to be used on the website as a quick preview of an IEER created by a teacher. Use that information to quickly know a specific IEER without going into detail; (9) Replace the existing room and object placer algorithm for the PCG. Use the PCG as the primary tool to generate the rooms and objects' position and rotation; (10) Implement VR gameplay. Allow the users to play the game using new technologies such as Oculus VR; (11) Implement multiplayer gameplay. Allow the users to play the game in groups to improve soft skills such as teamwork, communication and leadership.

References

Samantha P. Adams and Ronel Du Preez. Supporting student engagement through the gamification of learning activities: A design-based research approach. *Technology, Knowledge and Learning*, 27:119–138, March 2022. ISSN 2211-1662. doi: 10.1007/s10758-021-09500-x.

Alba Amato. Procedural content generation in the game industry. In *Game Dynamics*, pages 15–25. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-53088-8_2.

Matthew Armstrong, Cade Dopp, and Jesse Welsh. Design-based research: What is DBR, why might one do it, and how does one do it well? In Royce Kimmons, editor, *Education Research*. EdTech Books, 2022. URL https://edtechbooks.org/education_research/design_based_research.

Pratama Wirya Atmaja and Sugiarto. Generating object placements for optimum exploration and unpredictability in medium-coupling educational games. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pages 1–6, 2020. doi: 10.1109/3ICT51146.2020.9311976.

Dennis Beck, Leonel Morgado, and Patrick Shea. Finding the gaps about uses of immersive learning environments: A survey of surveys. *JUCS - Journal of Universal Computer Science*, 26: 1043–1073, August 2020. ISSN 0948-6968. doi: 10.3897/jucs.2020.055.

Michael Blatz and Oliver Korn. A very short history of dynamic and procedural content generation. In *Game Dynamics*, pages 1–13. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-53088-8_1.

Carlos Borrego, Cristina Fernández, Ian Blanes, and Sergi Robles. Room escape at class: Escape games activities to facilitate the motivation and learning in computer science. *Journal of Technology and Science Education*, 7:162, June 2017. ISSN 2013-6374. doi: 10.3926/jotse.247.

David David, Edwin, Edward Arman, Hikari, Natalia Chandra, and Nadia Nadia. Development of escape room game using VR technology. *Procedia Computer Science*, 157:646–652, 2019. ISSN 18770509. doi: 10.1016/j.procs.2019.08.223.

Panagiotis Fotaris and Theodoros Mastoras. Room2educ8: A framework for creating educational escape rooms based on design thinking principles. *Education Sciences*, 12:768, October 2022. ISSN 2227-7102. doi: 10.3390/educsci12110768.

Ana Rita Bernardo Garcia. World building and design of an immersive educational escape room: its impact on immersion. Master's thesis, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2022. URL https://hdl.handle.net/10216/142901.

Daniele Gravina, Ahmed Khalifa, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. Procedural content generation through quality diversity. In *2019 IEEE Conference on Games (CoG)*, pages 1–8, 2019. doi: 10.1109/CIG.2019.8848053.

Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9:1–22, February 2013. ISSN 1551-6857. doi: 10.1145/2422956.2422957.

D. Hooshyar, M. Yousefi, M. Wang, and H. Lim. A data-driven procedural-content-generation approach for educational games. *Journal of Computer Assisted Learning*, 34:731–739, December 2018a. ISSN 02664909. doi: 10.1111/jcal.12280.

Danial Hooshyar, Moslem Yousefi, and Heuiseok Lim. A procedural content generation-based framework for educational games: Toward a tailored data-driven game for developing early english reading skills. *Journal of Educational Computing Research*, 56:293–310, April 2018b. ISSN 0735-6331. doi: 10.1177/0735633117706909.

Marjaana Kangas, Pirkko Siklander, Justus Randolph, and Heli Ruokamo. Teachers' engagement and students' satisfaction with a playful learning environment. *Teaching and Teacher Education*, 63:274–284, April 2017. ISSN 0742051X. doi: 10.1016/j.tate.2016.12.018.

Mohammad Amin Kuhail, Areej ElSayary, Shahbano Farooq, and Ahlam Alghamdi. Exploring immersive learning experiences: A survey. *Informatics*, 9:75, September 2022. ISSN 2227-9709. doi: 10.3390/informatics9040075.

Sonsoles Lopez-Pernas, Aldo Gordillo, Enrique Barra, and Juan Quemada. Examining the use of an educational escape room for teaching programming in a higher education setting. *IEEE Access*, 7:31723–31737, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2902976.

Kayley M. Lyons, Nikki G. Lobczowski, Jeffrey A. Greene, Jason Whitley, and Jacqueline E. McLaughlin. Using a design-based research approach to develop and study a web-based tool to support collaborative learning. *Computers & Education*, 161:104064, February 2021. ISSN 03601315. doi: 10.1016/j.compedu.2020.104064.

Marco A. Moreno-Armendáriz, Hiram Calvo, José A. Torres-León, and Carlos A. Duchanoy. Iorand: A procedural videogame level generator based on a hybrid PCG algorithm. *Applied Sciences*, 12:3792, April 2022. ISSN 2076-3417. doi: 10.3390/app12083792.

Tiago Furtado Rossini Paula Pinto. Procedural modeling of an immersive educational escape room. Master's thesis, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2022. URL `https://hdl.handle.net/10216/145071`.

Jan Plass, Bruce Homer, and Charles Kinzer. Playful learning: An integrated design framework, December 2014. doi: 10.13140/2.1.4175.6969.

Jessica Reuter, Marta Ferreira Dias, Marlene Amorim, Claudia Figueiredo, Claudia Veloso, and Claudia Figueiredo. How to create educational escape rooms? Strategies for creation and design. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 694–698, New York, NY, USA, October 2020. ACM. ISBN 9781450388504. doi: 10.1145/3434780.3436664.

Louis Rice. Playful learning. *Journal for Education in the Built Environment*, 4:94–108, December 2009. ISSN 1747-4205. doi: 10.11120/jebe.2009.04020094.

Luiz Rodrigues and Jacques Brancher. Playing an educational game featuring procedural content generation: Which attributes impact players' curiosity? *RENOTE*, 17:254–263, July 2019. ISSN 1679-1916. doi: 10.22456/1679-1916.95952.

Ana Rita Fonseca Santos. Edscape-framework for learning with virtual escape rooms-3d environment. Master's thesis, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2022. URL `https://hdl.handle.net/10216/142734`.

Gillian Smith. Understanding procedural content generation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 917–926, New York, NY, USA, April 2014. ACM. ISBN 9781450324731. doi: 10.1145/2556288.2557341.

Gillian Smith. The future of procedural content generation in games. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 10:53–57, June 2021. ISSN 2334-0924. doi: 10.1609/aiide.v10i3.12748.

D. Sousa, A. Coelho, M.F. Torres, A.R. Garcia, and T. Rossini. Adaptability and procedural content generation for educational escape rooms. In *Proceedings of the European Conference on Games-based Learning*, volume 2022-Octob, pages 766–773, 2022. ISBN 9781914587511.

Joshua Taylor and Ian Parberry. Randomness + structure = clutter: A procedural object placement generator. In *Proceedings of the Conference Name*, pages 424–427. Publisher Name, 2011. doi: 10.1007/978-3-642-24500-8_57.

Luís Tinoca, João Piedade, Sofia Santos, Ana Pedro, and Sara Gomes. Design-based research in the educational field: A systematic literature review. *Education Sciences*, 12:410, June 2022. ISSN 2227-7102. doi: 10.3390/educsci12060410.

Tim Tutenel, Rafael Bidarra, Ruben Smelik, and Klaas Jan de Kraker. Rule-based layout solving and its application to procedural interior generation. In *Proceedings of the CASA Workshop on 3D Advanced Media in Gaming and Simulation (3AMIGAS)*, February 2009.

Yuzhong Zhang, Guixuan Zhang, and Xinyuan Huang. A survey of procedural content generation for games. In *2022 International Conference on Culture-Oriented Science and Technology (CoST)*, pages 186–190. IEEE, August 2022. ISBN 978-1-6654-6248-8. doi: 10.1109/CoST57098.2022.00046.

Celina Mo Zheng. Edscape-sistema web de condecorações e mini jogos a utilizar em escape room educativos. Master's thesis, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2022. URL `https://hdl.handle.net/10216/142590`.

Appendix A

Ontology

In this appendix, we explain in more detail the developed ontology. First, we present and describe each field of ontology. And lastly, some examples of how this information would be stored are shown.

A.1 Attributes

To be easier to present and explain every attribute, we use the classes in the code that store the information before being saved in a file. Nine classes are used: Settings, GameSettings, ConfigGameObj, ClueObj, ThoughtObj, IterativeObj, Minigame, PathOp and TextOp. The information from the Settings and GameSettings is stored in a constant file ("settings.json"), and the remaining data is stored in files created in runtime after the user creates an IEER. The following subsections are named according to the previous classes and only contain their corresponding attributes.

A.1.1 Settings

The Settings class is responsible for storing information about all the available IEERs games that are shown to the user when entering the EdScape homepage.

- **games (List<GameSettings>):** List of games that are provided to the user when playing the EdScape game. These games are from type GameSettings, which is aborded in the following subsection.

A.1.2 GameSettings

The GameSettings class stores superficial information about an IEER, such as the name, the file path containing specific information about the IEER and if the IEER is still in the test phase.

- **name (string):** Name of the IEER that is displayed to the user.
- **path (string):** Path of the file belonging to the IEER containing all its specific information.

- **test (bool):** Boolean variable that indicates if the IEER is still in the test phase. If true, both initial and final surveys are omitted. Otherwise, they are not.

A.1.3 ConfigGameObj

The ConfiGameObj class holds all the specific information about the IEER, meaning everything we need to know about the IEER we have created or will play is here. All the attributes mentioned in Chapter 4 are present below, together with the others. The ones that were not mentioned are either generated in code or disabled for the user since, at the moment, they are not yet suitable.

- **gameId (string):** Unique game identifier generated in runtime after creating a new IEER. It can be later used to access the game information on the website.
- **userEmail (string):** Email of the user logged in that played the IEER.
- **issuerEmail (string):** Email of the educator that created the IEER. This grants access to all information from those who played the IEER.
- **creationDate (string):** Date when the IEER was created in the format of "dd/mm/yyyy hh:mm:ss".
- **ieerPurpose (string):** What is the overall purpose of the IEER?
- **ieerLearningObjectives (string []):** Array with the learning objective of the IEER.
- **ieerType (string):** What type of IEER will be developed? 3D environment or VR environment?
- **requiredKnowlage (string):** What knowledge is required to succeed in the game? Is it explicit, assumed, retrievable, or a mix?
- **numOfParticipants (int):** How many participants need to play simultaneously?
- **ieerCoursePosition (string):** Where will the IEER be positioned in the course curriculum? As a stand-alone activity, at the introduction of a course, during a course in addition to a lecture, as an assessment, or as a serial story?
- **ieerStoryType (string):** Will the story be stand-alone like an entire movie or framed as an episode with a continuous narrative arc?
- **ieerForEvaluation (bool):** Will the IEER be a formative or summative assessment tool?
- **languageBarriers (bool):** Are there any language barriers that may prevent non-native speakers from playing the game?
- **differentLevelsOfImpairments (bool):** Are there any tasks that may prevent participants with differing mobility levels or sensory impairments from playing the game?

- **selfReflectionTime (float):** How much time will be available for self-reflection after the game?
- **initialTime (float):** What will the game's duration be in seconds? The default value is 3600 seconds (60 minutes).
- **textOptionIDs (List<string>):** List with the available game languages. The user is asked to select a language if this list is not empty.
- **infoIntroductionTitle (List<TextOp>):** List of TextOp, objects that contain different versions of the introductory information title, taking into account the number of available languages. This appears before the game starts and is always available to the player.
- **infoIntroductionText (List<TextOp>):** List of TextOp, objects that contain different versions of the introductory information text, taking into account the number of available languages. This appears before the game starts and is always available to the player.
- **introductionVideoPath (List<PathOp>):** List of PathOp, objects that contain different versions of the introductory video, taking into account the number of available languages. The video follows the introductory information.
- **startText (List<TextOp>):** List of TextOp, objects that contain different versions of the start text, taking into account the number of available languages. The message appears after the video and before the game starts.
- **timeEndedText (List<TextOp>):** List of TextOp, objects that contain different versions of the time ended text, taking into account the number of available languages. The message appears when the game time runs out.
- **congratulationsText (List<TextOp>):** List of TextOp, objects that contain different versions of the congratulations text, taking into account the number of available languages. The message appears when the game finishes successfully.
- **endAfterSurveyText (List<TextOp>):** List of TextOp, objects that contain different versions of the text that appears after answering the final survey, taking into account the number of available languages.
- **initialSurveyPath (List<PathOp>):** List of PathOp, objects that contain different versions of the initial survey, taking into account the number of available languages. The initial survey appears before the introductory information.
- **finalSurveyPath (List<PathOp>):** List of PathOp, objects that contain different versions of the final survey, taking into account the number of available languages. The last survey appears after the game finishes.

- **clueObjs (List<ClueObj>):** List of CluObj, objects with detailed information about a clue. The clues appear to the user when playing the game.
- **thoughtObjs (List<ThoughtObj>):** List of ThoughtObj, objects with detailed information about a thought. The thoughts appear to the user when playing the game.
- **interactiveObjs (List<IterativeObj>):** List of IterativeObj, objects with detailed information about an interactive object. The interactive objects are placed in the room before playing the game.

A.1.4 ClueObj

The clueObj class contains detailed information about the clues that are displayed to the user during the gameplay.

- **itemIdNotOwned (string):** Name of the item the user has not yet interacted with. If left empty, the clue is presented at the beginning of the game.
- **inLevel (string):** It is used only to trigger this clue when the user enters this level. Levels can be changed when interacting successfully with an object.
- **showClueAfterTime (float):** Time in seconds when the clue should be displayed to the user. The default value is 0, meaning the user needs to ask for the clue to be shown.
- **message (List<TextOp>):** List of TextOp, objects that contain different versions of the clue message, taking into account the number of available languages.
- **penalization (float):** Penalization in seconds if the user asks for a clue before it is shown automatically.
- **destroyAfterShowing (bool):** Boolean variable that indicates if the clues must be destroyed after being shown or repeated.
- **suppressNextCluesUntilGrabItem (bool):** Boolean variable that indicates if the following clues must be suppressed until the player grabs the object.

A.1.5 ThoughtObj

The thoughtObj class stores detailed information about the thoughts displayed to the player during gameplay.

- **itemGrabbed (string):** Item grabbed by the player that triggers this. If left empty, it is presented at the beginning of the game.
- **showThoughtAfterTime (float):** Time in seconds when the thought should be displayed to the user.

- **message (List<TextOp>):** List of TextOp, objects that contain different versions of the thought message, taking into account the number of available languages.

A.1.6 IterativeObj

The iterativeObj class holds detailed information about the interactive objects. Interactive objects are placed in the room before the game starts.

- **id (string):** Name of the object placed in the room.
- **entersLevel (string):** When the player interacts successfully with this object, the level changes to this.
- **initialPosition (float []):** Initial position of the object in a 3D room (x,y,z).
- **initialRotation (float []):** Initial rotation of the object in a 3D room (x,y,z).
- **finalPosition (float []):** Final position of the object in a 3D room (x,y,z).
- **finalRotation (float []):** Final rotation of the object in a 3D room (x,y,z).
- **unlockItemsIds (string []):** Array of objects that the user needs to interact successfully before interacting with this one.
- **messageLocked (List<TextOp>):** List of TextOp, objects that contain different versions of the locked message, taking into account the number of available languages. The message is displayed to the user if this object is locked.
- **messageSucceeded (List<TextOp>):** List of TextOp, objects that contain different versions of the success message, taking into account the number of available languages. The message is displayed to the user if he successfully interacts with the object.
- **messageExited (List<TextOp>):** List of TextOp, objects that contain different versions of the exit message, taking into account the number of available languages. The message is displayed to the user if he leaves the interaction with the object.
- **miniGame (MiniGame):** Minigame that is presented to the user when interacting with the object.
- **introTitle (List<TextOp>):** List of TextOp, objects that contain different versions of the introduction title, taking into account the number of available languages. The introduction title is displayed to the user at the beginning of the interaction with the object.
- **introText (List<TextOp>):** List of TextOp, objects that contain different versions of the introduction text, taking into account the number of available languages. The introduction text is displayed to the user at the beginning of the interaction with the object.

- **videoPath (List<PathOp>):** List of PathOp, objects that contain different versions of the introductory video, taking into account the number of available languages. The video is shown to the user after the introductory information.
- **finalIntroTitle (List<TextOp>):** List of TextOp, objects that contain different versions of the final title, taking into account the number of available languages. The final title is displayed to the user at the end of the interaction with the object.
- **finalIntroText (List<TextOp>):** List of TextOp, objects that contain different versions of the final text, taking into account the number of available languages. The final text is displayed to the user at the end of the interaction with the object.
- **finalVideoPath (List<PathOp>):** List of PathOp, objects that contain different versions of the final video, taking into account the number of available languages. The video is shown to the user after the last information.
- **winsGame (bool):** Boolean variable that, if true, the game finishes after interacting successfully with this object.

A.1.7 Minigame

The minigame class registers information about the minigame that the player may play.

- **id (string):** Type of the minigame that is played. At the moment, only the URL type is available.
- **pathOptions (List<PathOp>):** List of PathOp, objects that contain different versions of the minigame path, taking into account the number of available languages.

A.1.8 PathOp

The pathOp class contains information about the language and the corresponding path.

- **id (string):** Chosen language to identify the path.
- **path (string):** Path to reach the content to be displayed.

A.1.9 TextOp

The textOp class contains information about the language and the corresponding text.

- **id (string):** Chosen language to identify the text.
- **text (string):** Text to be displayed to the user.

A.2 Examples

In this section, we present two examples: (1) a "settings.json" file that holds the information about the Settings and GameSettings classes; (2) a generated JSON file of an IEER created by a colleague during the prototype test.

```
{
    "games":
    [
        {
            "name": "Ethics 101 Test",
            "path": "Escape2Ethics101Test/_globalGame.json",
            "test": false
        },
        {
            "name": "Rita",
            "path": "Rita/rita.json",
            "test": false
        },
        {
            "name": "Test",
            "path": "Test/configurationEN.json",
            "test": false
        },
        {
            "name": "Exemplo",
            "path": "Exemplo/exemplo.json",
            "test": false
        },
        {
            "name": "Exemplo Com Solucao",
            "path": "Exemplo Com Solucao/exemplo_com_solucao.json",
            "test": false
        }
    ]
}
```

Listing A.1: Example of a settings file

```
{
    "gameId":"History Adventure31_05_2023_16_13_17",
    "userEmail":"",
    "issuerEmail":"up201809566@up.pt",
    "creationDate":"31/05/2023 16:13:17",
    "ieerPurpose":"Os participantes, numa sala virtual, terao de responder corretamente
        a algumas perguntas de Historia, relacionadas com o tema que estao a estudar no
        momento, e, posteriormente, receberao indicacoes da localizacao dos objetos que
        necessitarao para sair da sala.",
    "ieerLearningObjectives":["Aprender a materia incluida no programa de Historia"],
    "ieerType":"",
    "requiredKnowlage":"Mix",
    "numOfParticipants":1,
    "ieerCoursePosition":"Stand-alone activity",
    "ieerStoryType":"Framed with a continuous narrative arc",
    "ieerForEvaluation":false,
    "languageBarriers":false,
    "differentLevelsOfImpairments":false,
    "selfReflectionTime":0.0,
    "initialTime":3600.0,
    "textOptionIDs":["Portuguese"],
    "infoIntroductionTitle":[
        {
            "id":"",
            "text":"History Adventure"
        },
        {
            "id":"Portuguese",
            "text":"Aventura de historia"
        }
    ],
    "infoIntroductionText":[
        {
            "id":"",
            "text":"Welcome to History Adventure"
        },
        {
            "id":"Portuguese",
            "text":"Bem vindo a aventura de historia"
        }
    ],
    "introductionVideoPath":[],
    "startText":[
        {
            "id":"",
            "text":"Lets go"
        },
        {
            "id":"Portuguese",
            "text":"Vamos"
        }
    ],
    "timeEndedText":[
        {
            "id":"",
            "text":"The time is finished"
        },
        {
            "id":"Portuguese",
            "text":"A hora esta terminada"
        }
    ],
    "congratulationsText":[
        {
            "id":"",
            "text":"Well done"
        },
        {
            "id":"Portuguese",
            "text":"Bom trabalho"
        }
    ],
    "endAfterSurveyText":[
        {
            "id":"",
            "text":"End"
        },
        {
            "id":"Portuguese",
            "text":"Fim"
        }
    ],
    "initialSurveyPath":[],
    "finalSurveyPath":[],
    "clueObjs":[
        {
            "itemIdNotOwned":"Book_1",
            "inLevel":"",
            "showClueAfterTime":0.0,
            "message":[],
            "penalization":0.0,
            "destroyAfterShowing":false,
            "suppressNextCluesUntilGrabItem":false
        }
    ],
    "thoughtObjs":[
        {
            "itemGrabbed":"Book_1",
            "showThoughtAfterTime":5.0,
            "message":[
                {
                    "id":"",
                    "text":"What I need to find?"
                },
                {
                    "id":"Portuguese",
                    "text":"O que eu preciso encontrar?"
                }
            ]
        }
    ],
    "interactiveObjs":[
        {
            "id":"Book_1",
            "entersLevel":"",
            "initialPosition":[],
            "initialRotation":[],
            "finalPosition":[],
            "finalRotation":[],
            "unlockItemsIds":[],
            "messageLocked":[],
            "messageSucceeded":[],
            "messageExited":[],
            "miniGame":{
                "id":"",
                "pathOptions":[]
            },
            "introTitle":[],
            "introText":[],
            "videoPath":[],
            "finalIntroTitle":[],
            "finalIntroText":[],
            "finalVideoPath":[],
            "winsGame": false
        }
    ]
}
```

Listing A.2: Example of JSON file generated from an IEER creation

Appendix B

Backoffice Design

This appendix shows all the constructed input field drafts in different sections to be easier to read. Note that all the drafts below may not correspond to the actual input field or may not be present in the current backoffice version.

B.1 File Remove

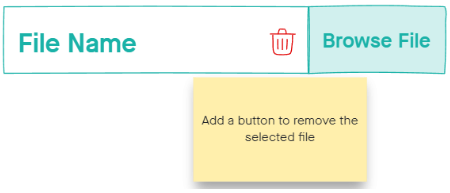


Figure B.1: Draft of a way to remove the file from the input field

B.2 Form Field Hint

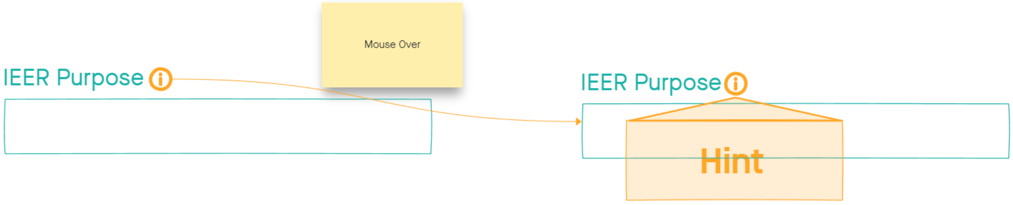
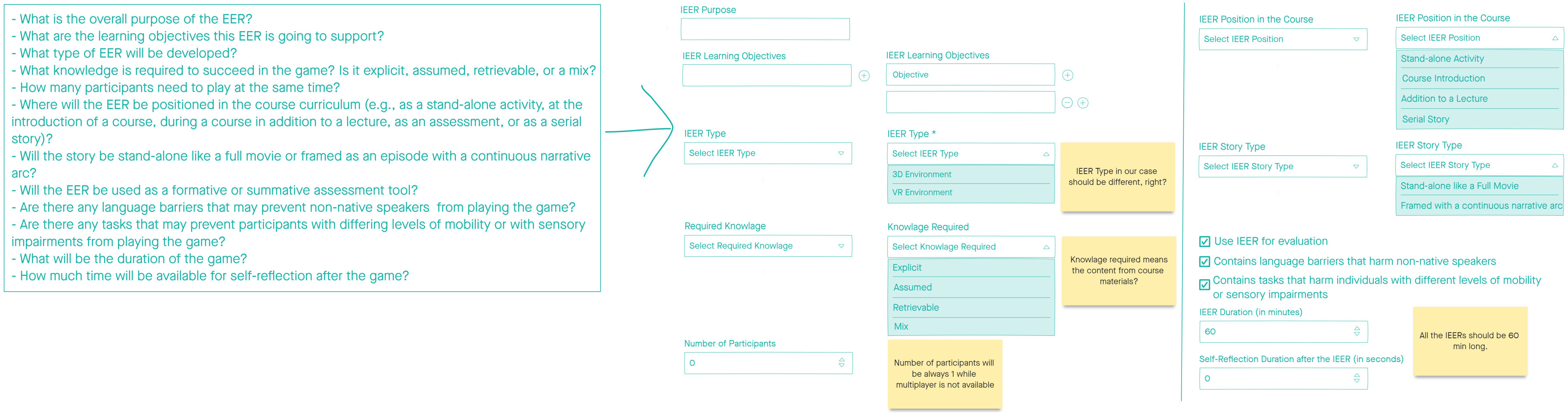


Figure B.2: Draft of a way to show information about a form field

B.3 SMART Questions



- What is the overall purpose of the EER?
- What are the learning objectives this EER is going to support?
- What type of EER will be developed?
- What knowledge is required to succeed in the game? Is it explicit, assumed, retrievable, or a mix?
- How many participants need to play at the same time?
- Where will the EER be positioned in the course curriculum (e.g., as a stand-alone activity, at the introduction of a course, during a course in addition to a lecture, as an assessment, or as a serial story)?
- Will the story be stand-alone like a full movie or framed as an episode with a continuous narrative arc?
- Will the EER be used as a formative or summative assessment tool?
- Are there any language barriers that may prevent non-native speakers from playing the game?
- Are there any tasks that may prevent participants with differing levels of mobility or with sensory impairments from playing the game?
- What will be the duration of the game?
- How much time will be available for self-reflection after the game?

IEER Type in our case should be different, right?

Knowlage required means the content from course materials?

Number of Participants
0

Number of participants will be always 1 while multiplayer is not available

☑ Use IEER for evaluation

☑ Contains language barriers that harm non-native speakers

☑ Contains tasks that harm individuals with different levels of mobility or sensory impairments

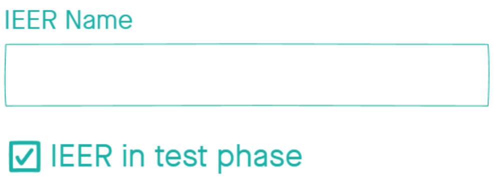
IEER Duration (in minutes)
60

Self-Reflection Duration after the IEER (in seconds)
0

All the IEERs should be 60 min long.

Figure B.3: Input field drafts from the SMART questions of Fotaris and Mastoras (2022)

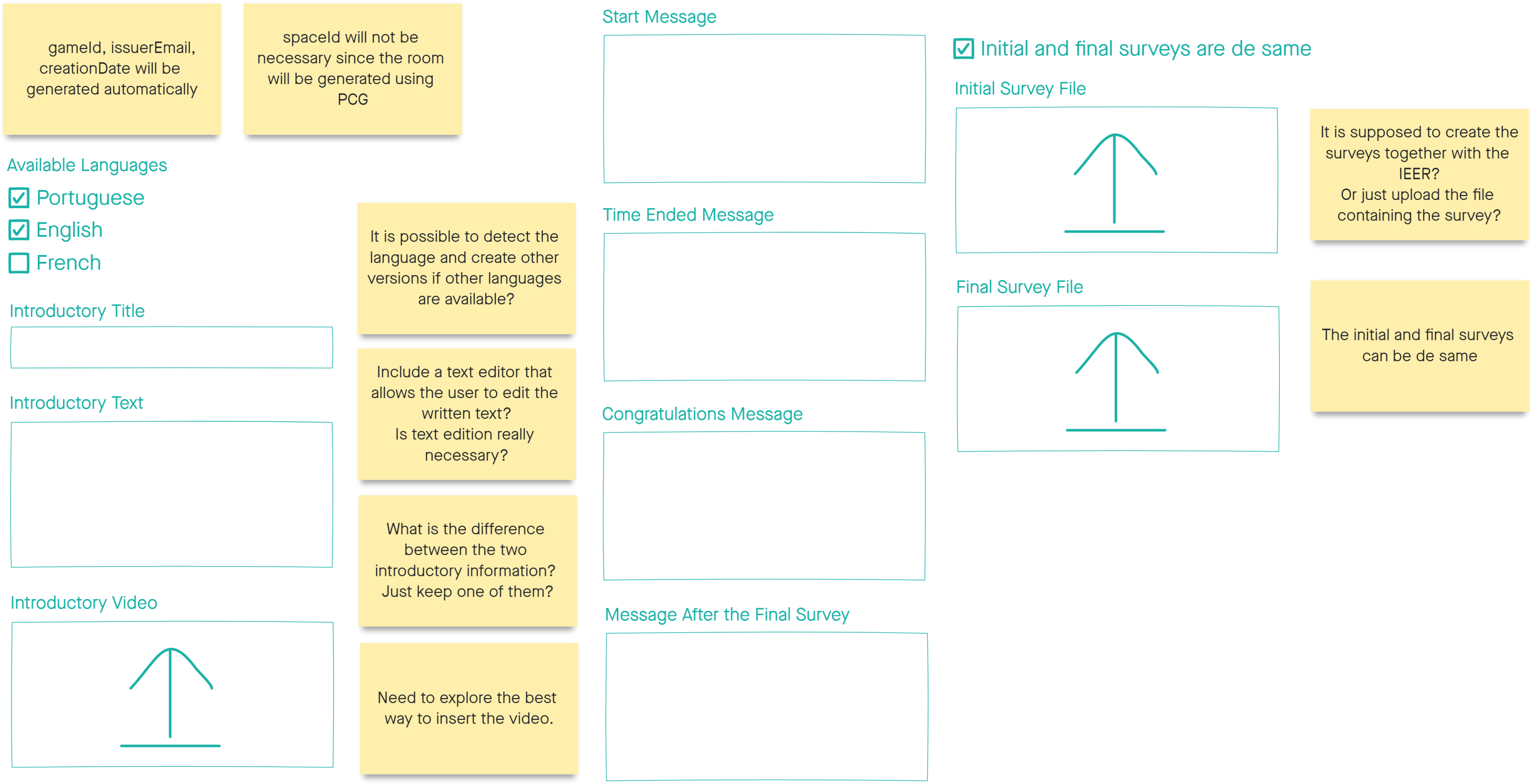
B.4 Settings



☑ IEER in test phase

Figure B.4: Input field drafts from the settings attributes of Santos (2022) ontology

B.5 Game



gameId, issuerEmail, creationDate will be generated automatically

spaceId will not be necessary since the room will be generated using PCG

It is possible to detect the language and create other versions if other languages are available?

Include a text editor that allows the user to edit the written text?
Is text edition really necessary?

What is the difference between the two introductory information?
Just keep one of them?

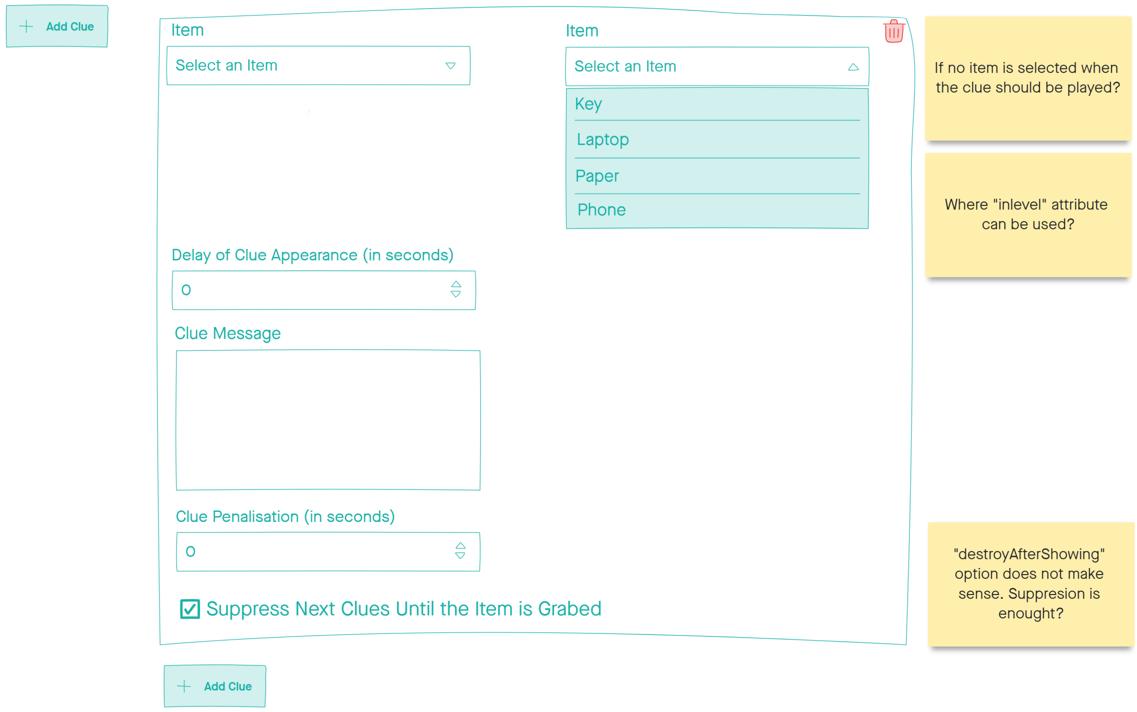
Need to explore the best way to insert the video.

It is supposed to create the surveys together with the IEER?
Or just upload the file containing the survey?

The initial and final surveys can be de same

Figure B.5: Input field drafts from the game attributes of Santos (2022) ontology

B.6 Clues


Add Clue
Item
Select an Item
Item
Select an Item
Key
Laptop
Paper
Phone
Delay of Clue Appearance (in seconds)
0
Clue Message
Clue Penalisation (in seconds)
0
☑ Suppress Next Clues Until the Item is Grabed
Add Clue

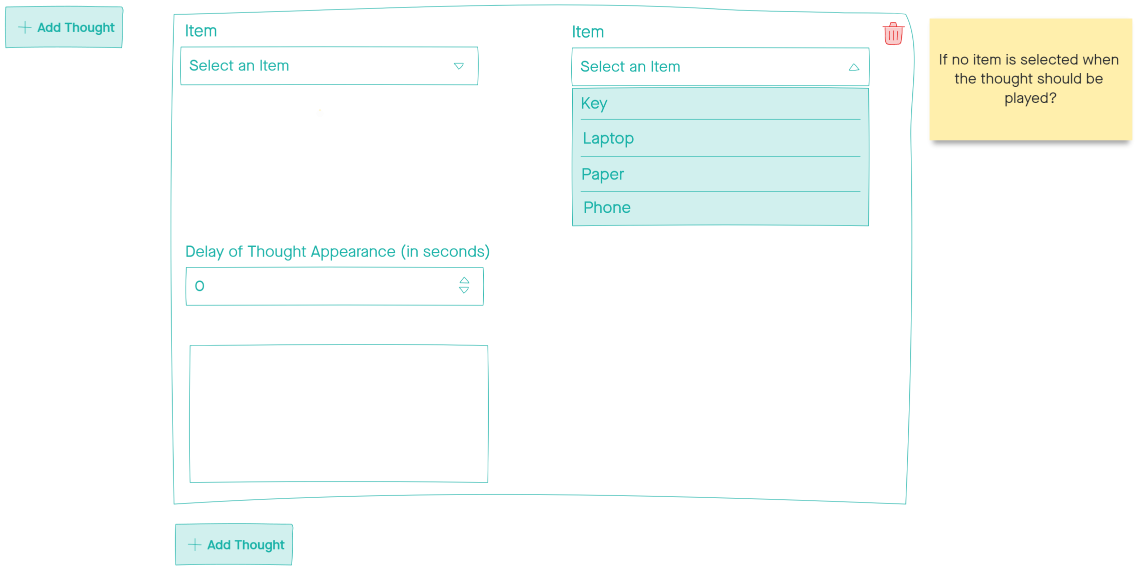
If no item is selected when the clue should be played?

Where "inlevel" attribute can be used?

"destroyAfterShowing" option does not make sense. Suppresion is enought?

Figure B.6: Input field drafts from the clue attributes of Santos (2022) ontology

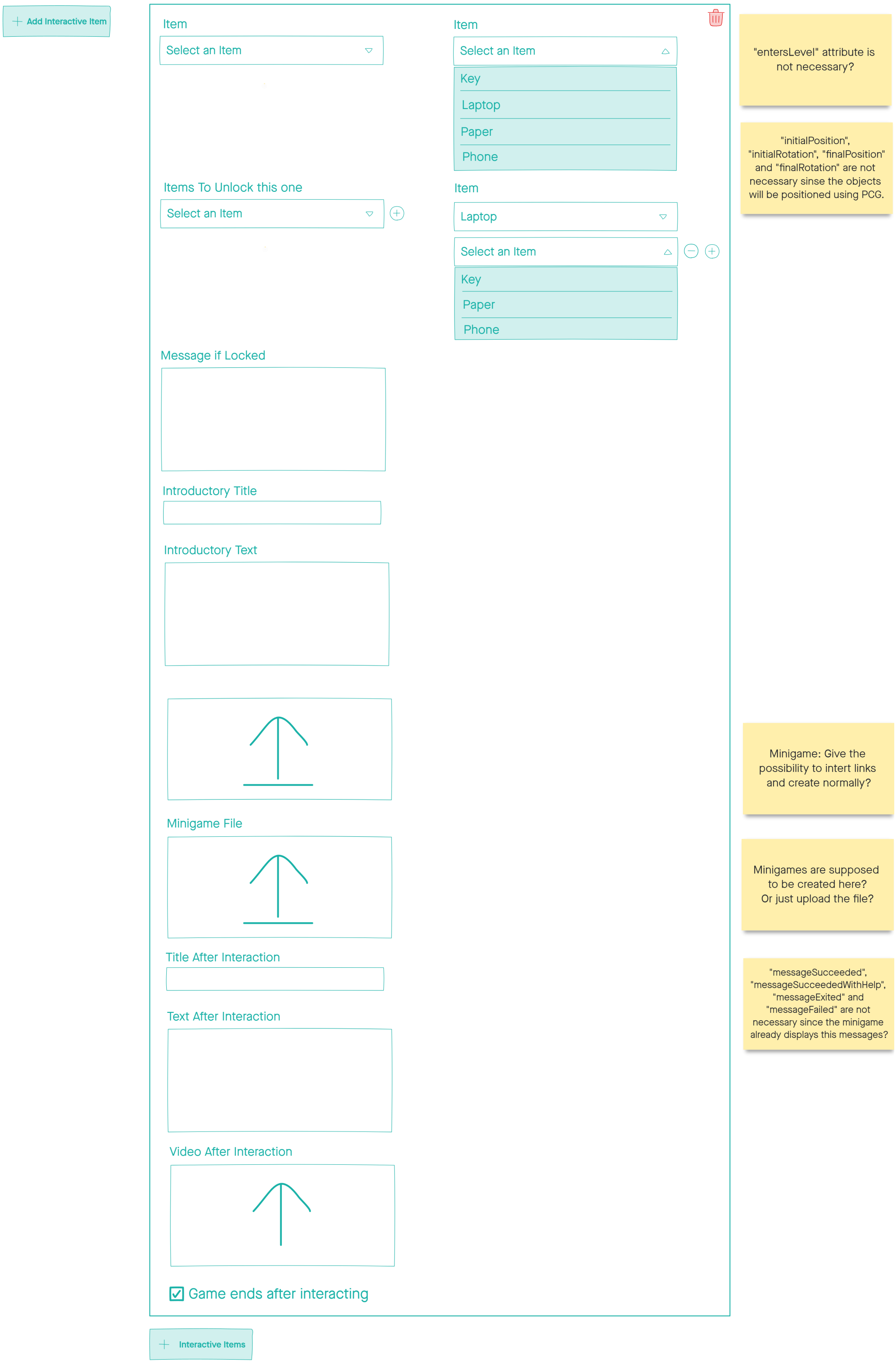
B.7 Thoughts



If no item is selected when the thought should be played?

Figure B.7: Input field drafts from the thought attributes of Santos (2022) ontology

B.8 Interactive Objects



"entersLevel" attribute is not necessary?

"initialPosition", "initialRotation", "finalPosition" and "finalRotation" are not necessary sinse the objects will be positioned using PCG.

Minigame: Give the possibility to intert links and create normally?

Minigames are supposed to be created here? Or just upload the file?

"messageSucceeded", "messageSucceededWithHelp", "messageExited" and "messageFailed" are not necessary since the minigame already displays this messages?

Figure B.8: Input field drafts from the interactive objects attributes of Santos (2022) ontology

B.9 Minigame

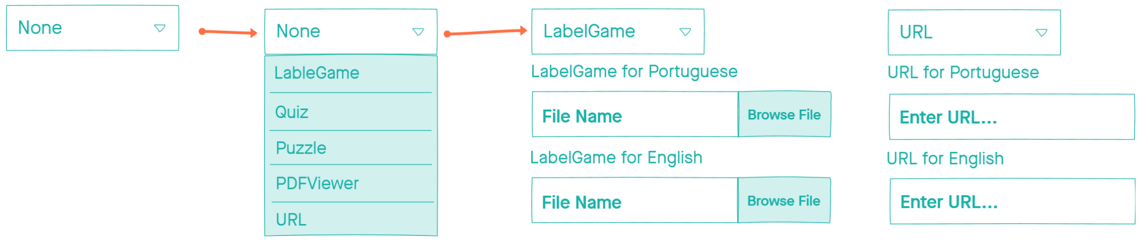
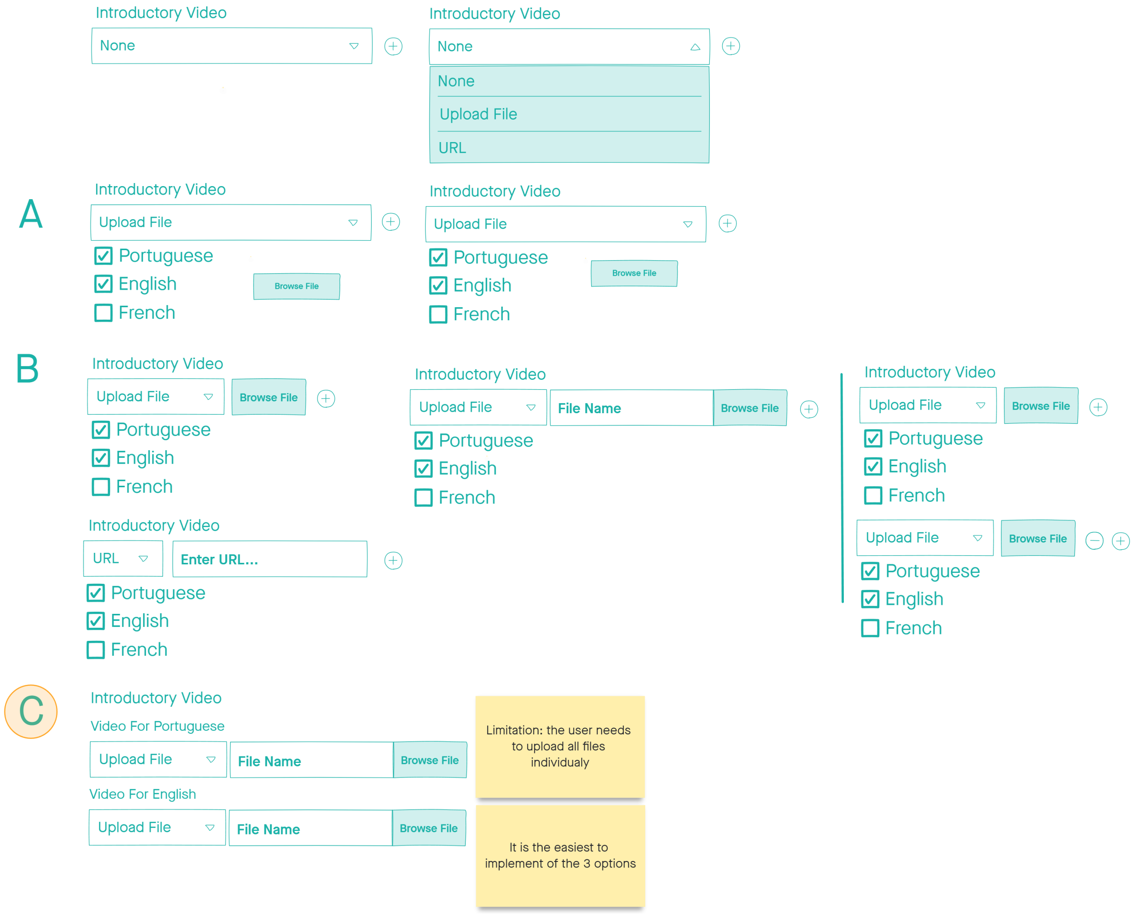


Figure B.9: Input field drafts from the minigame attributes of Santos (2022) ontology

B.10 File or URL



Limitation: the user needs to upload all files individualy

It is the easiest to implement of the 3 options

Figure B.10: Input field drafts from the attributes related to files of Santos (2022) ontology

Appendix C

Backoffice Prefabs

This appendix shows all the developed prefabs, as with the previous appendix, in different sections to be easier to read. We present the prefabs of the panels and the form fields from the more simple to the complex ones.

C.1 Information Icon

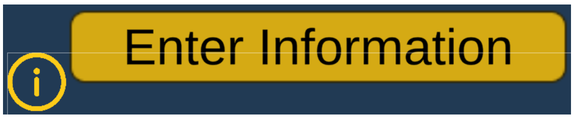


Figure C.1: Prefab of the information icon

C.2 Resizable Input Field

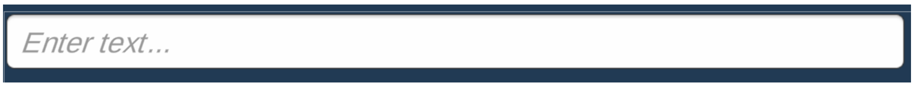


Figure C.2: Prefab of the resizable input field

C.3 Resizable Input Field + Buttons

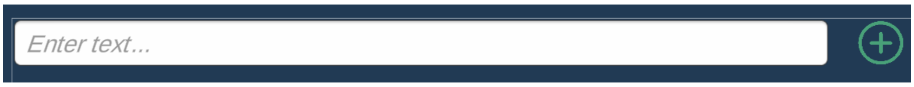


Figure C.3: Prefab of the resizable input field with buttons to add and remove new objects of the same type

C.4 Dropdown + Buttons

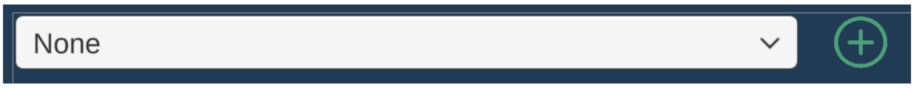


Figure C.4: Prefab of the dropdown with buttons to add and remove new objects of the same type

C.5 Dropdown

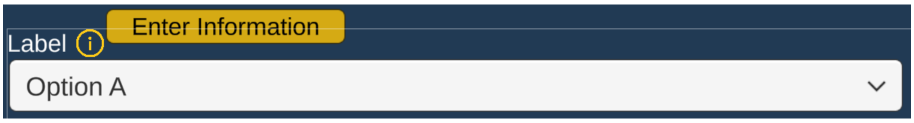


Figure C.5: Prefab of the dropdown

C.6 Dropdowns

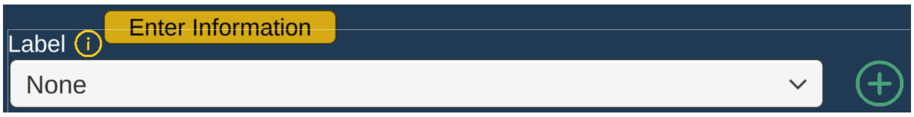


Figure C.6: Prefab of the dropdowns, where it is possible to add and remove dropdowns

C.7 Text Input

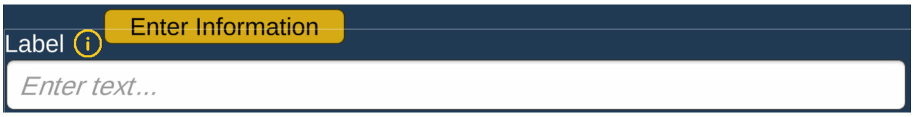


Figure C.7: Prefab of the text input

C.8 Text Inputs

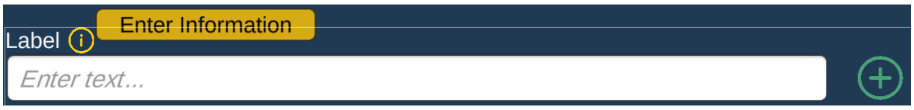


Figure C.8: Prefab of the text inputs, where it is possible to add and remove text inputs

C.9 Decimal Input

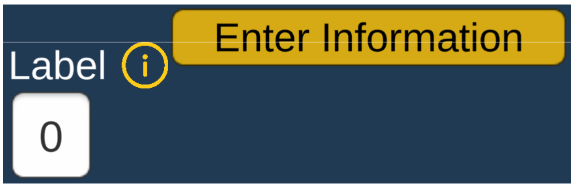


Figure C.9: Prefab of the decimal input

C.10 Integer Input

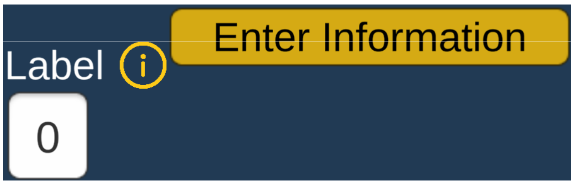


Figure C.10: Prefab of the integer input

C.11 Toggle

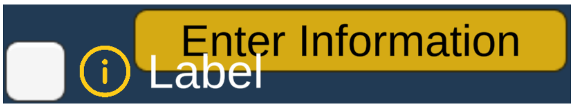


Figure C.11: Prefab of the toggle

C.12 Equal Input Switch

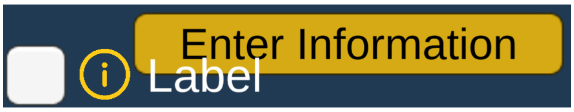


Figure C.12: Prefab of the equal input switch

C.13 Multi-input

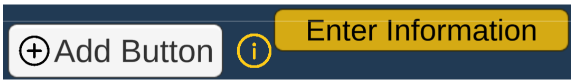


Figure C.13: Prefab of the multi-input

C.14 Toggle Group

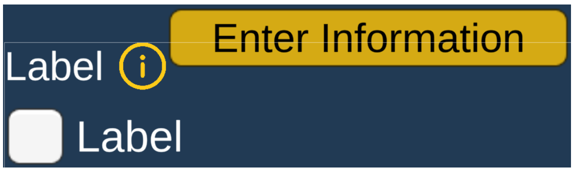


Figure C.14: Prefab of the toggle group

C.15 Minigame

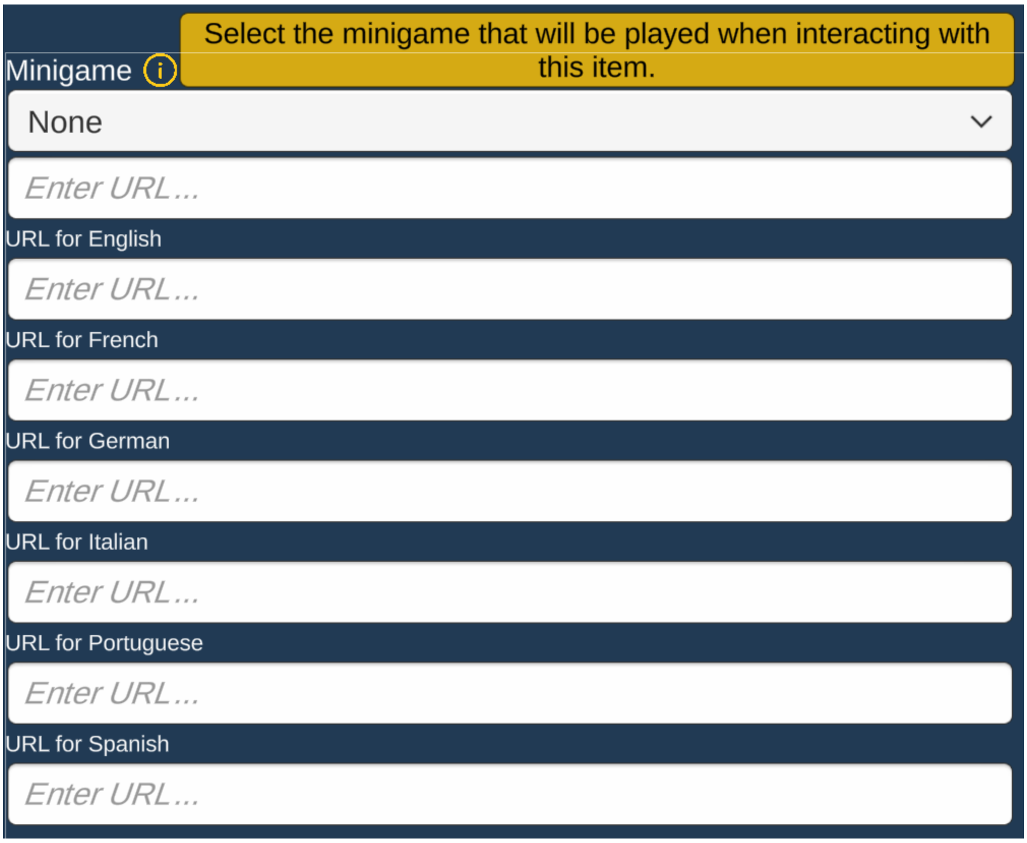
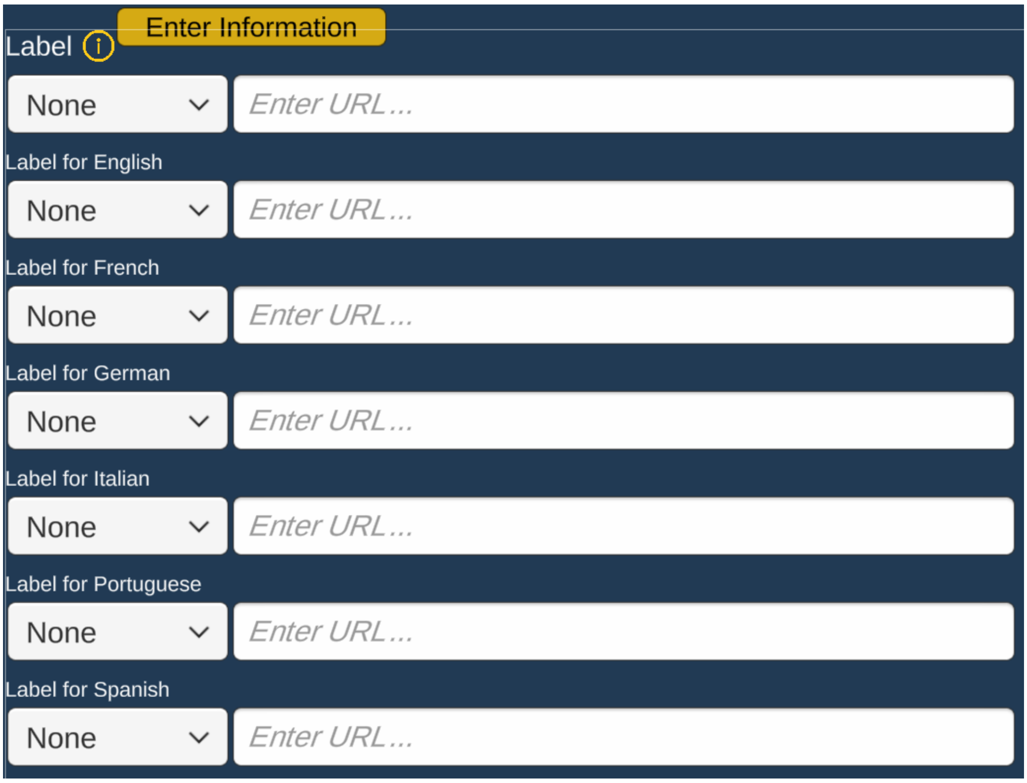


Figure C.15: Prefab of the minigame

C.16 File or URL Input



Label ⓘ Enter Information

None | Enter URL…

Label for English

None | Enter URL…

Label for French

None | Enter URL…

Label for German

None | Enter URL…

Label for Italian

None | Enter URL…

Label for Portuguese

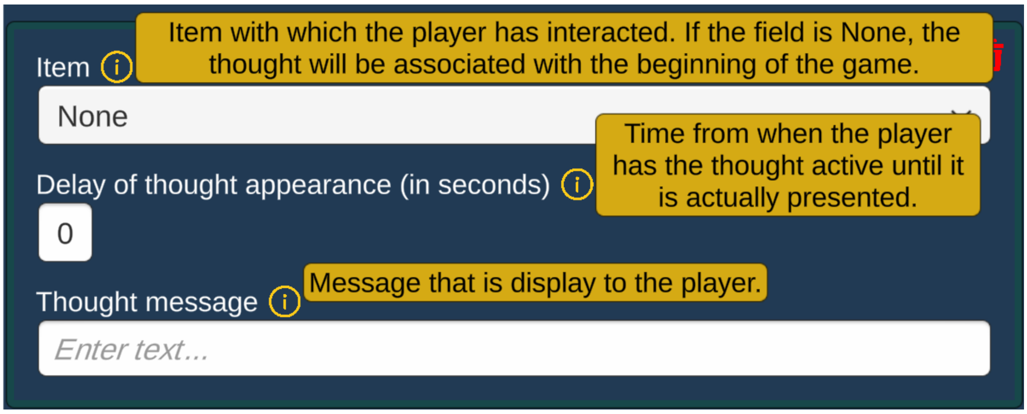
None | Enter URL…

Label for Spanish

None | Enter URL…

Figure C.16: Prefab of the file or URL input

C.17 Thought



Item ⓘ Item with which the player has interacted. If the field is None, the thought will be associated with the beginning of the game.

None

Delay of thought appearance (in seconds) ⓘ Time from when the player has the thought active until it is actually presented.

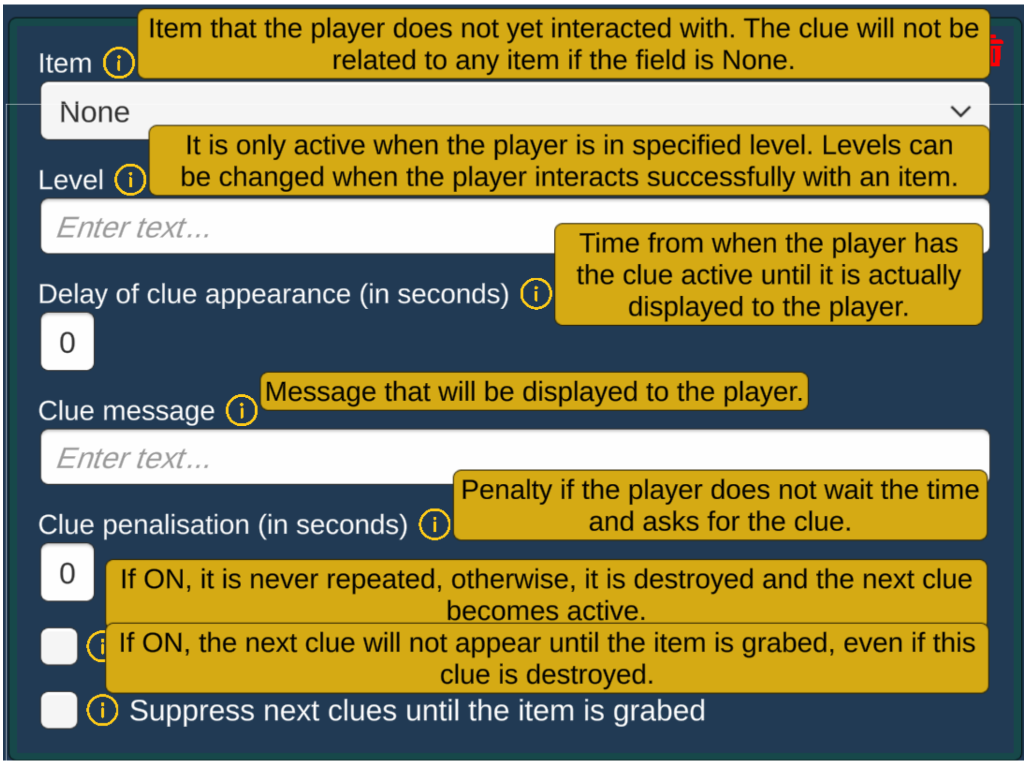
0

Thought message ⓘ Message that is display to the player.

Enter text…

Figure C.17: Prefab of the thought

C.18 Clue



Item
Item that the player does not yet interacted with. The clue will not be related to any item if the field is None.

None

Level
It is only active when the player is in specified level. Levels can be changed when the player interacts successfully with an item.

Enter text...

Delay of clue appearance (in seconds)
Time from when the player has the clue active until it is actually displayed to the player.

0

Clue message
Message that will be displayed to the player.

Enter text...

Clue penalisation (in seconds)
Penalty if the player does not wait the time and asks for the clue.

0

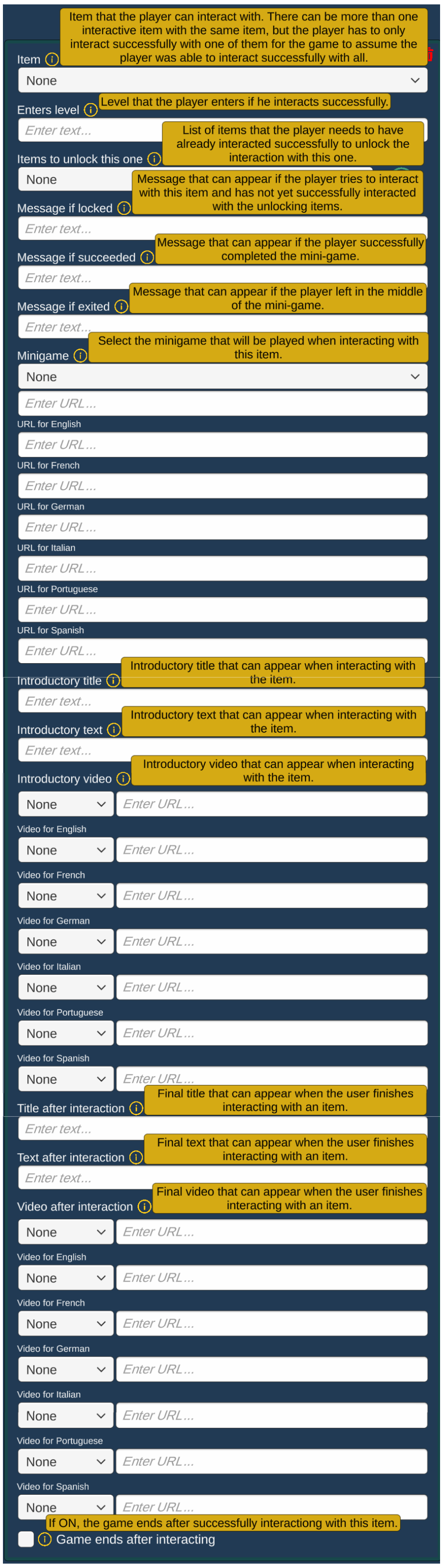
□ If ON, it is never repeated, otherwise, it is destroyed and the next clue becomes active.

□ If ON, the next clue will not appear until the item is grabed, even if this clue is destroyed.

□ Suppress next clues until the item is grabed

Figure C.18: Prefab of the clue

C.19 Interactive Objects



Item that the player can interact with. There can be more than one interactive item with the same item, but the player has to only interact successfully with one of them for the game to assume the player was able to interact successfully with all.

Item ⓘ

None

Level that the player enters if he interacts successfully.

Enters level ⓘ

Enter text...

List of items that the player needs to have already interacted successfully to unlock the interaction with this one.

Items to unlock this one ⓘ

None

Message that can appear if the player tries to interact with this item and has not yet successfully interacted with the unlocking items.

Message if locked ⓘ

Enter text...

Message that can appear if the player successfully completed the mini-game.

Message if succeeded ⓘ

Enter text...

Message that can appear if the player left in the middle of the mini-game.

Message if exited ⓘ

Enter text...

Select the minigame that will be played when interacting with this item.

Minigame ⓘ

None

Enter URL...

URL for English

Enter URL...

URL for French

Enter URL...

URL for German

Enter URL...

URL for Italian

Enter URL...

URL for Portuguese

Enter URL...

URL for Spanish

Enter URL...

Introductory title that can appear when interacting with the item.

Introductory title ⓘ

Enter text...

Introductory text that can appear when interacting with the item.

Introductory text ⓘ

Enter text...

Introductory video that can appear when interacting with the item.

Introductory video ⓘ

None Enter URL...

Video for English

None Enter URL...

Video for French

None Enter URL...

Video for German

None Enter URL...

Video for Italian

None Enter URL...

Video for Portuguese

None Enter URL...

Video for Spanish

None Enter URL...

Final title that can appear when the user finishes interacting with an item.

Title after interaction ⓘ

Enter text...

Final text that can appear when the user finishes interacting with an item.

Text after interaction ⓘ

Enter text...

Final video that can appear when the user finishes interacting with an item.

Video after interaction ⓘ

None Enter URL...

Video for English

None Enter URL...

Video for French

None Enter URL...

Video for German

None Enter URL...

Video for Italian

None Enter URL...

Video for Portuguese

None Enter URL...

Video for Spanish

None Enter URL...

If ON, the game ends after successfully interactiong with this item.

☐ ⓘ Game ends after interacting

Figure C.19: Prefab of the interactive objects form field

C.20 Popup

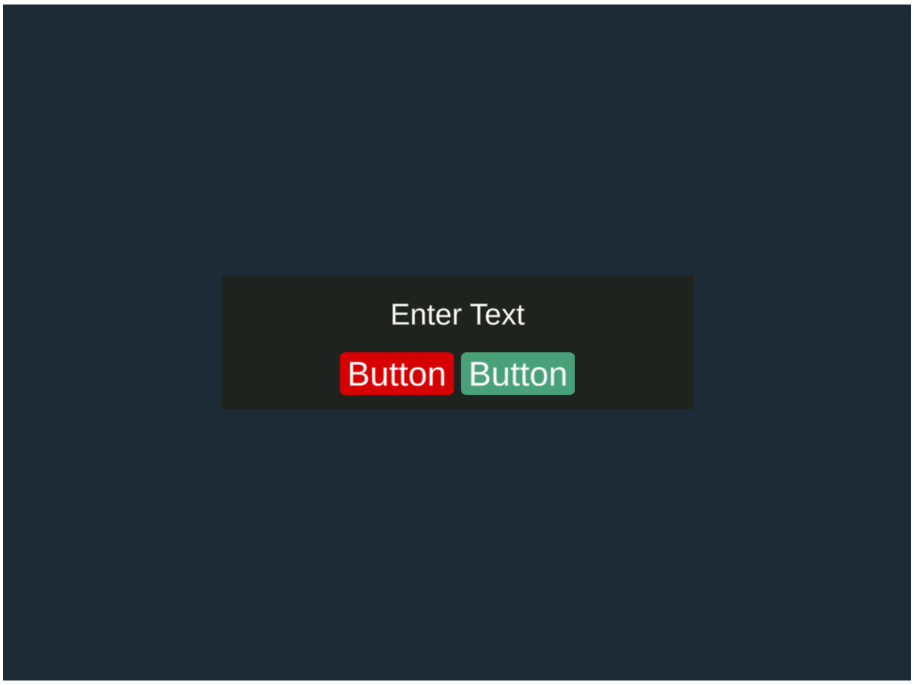


Figure C.20: Prefab of the popup

C.21 EdScape Homepage Panel

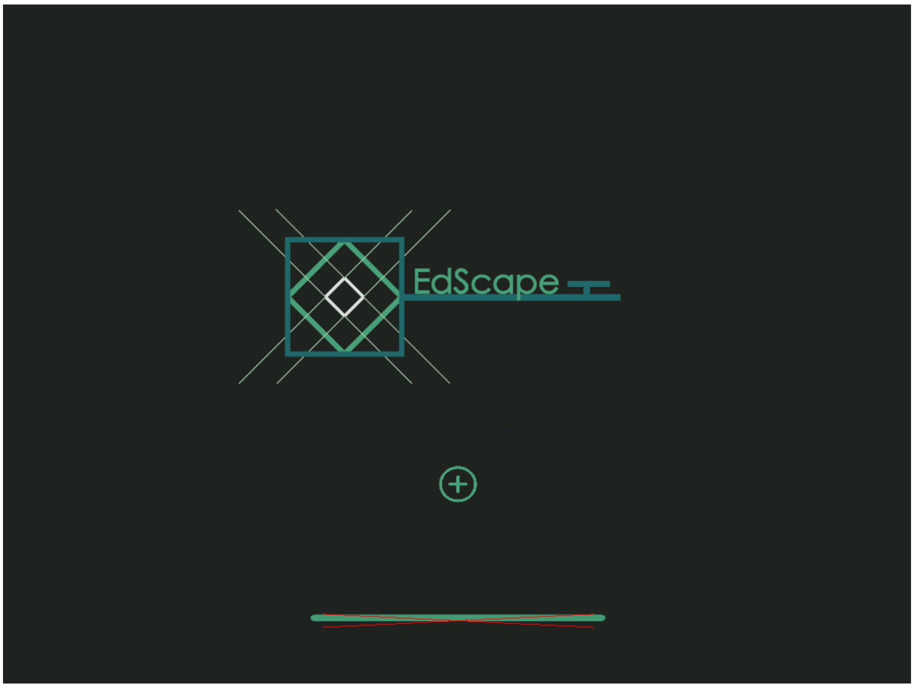


Figure C.21: Prefab of the EdScape homepage panel

C.22 Webview Panel

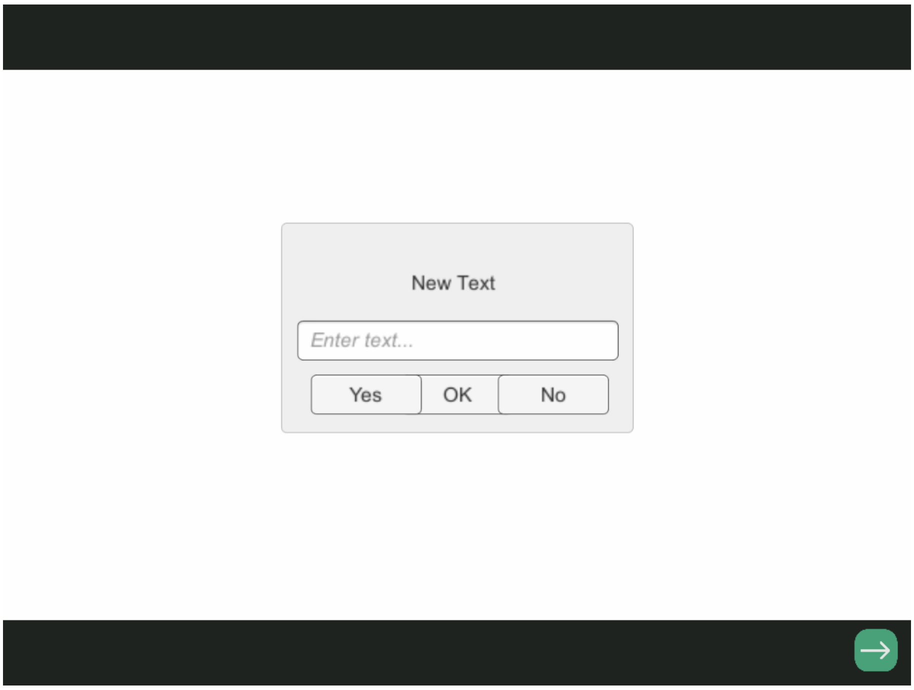
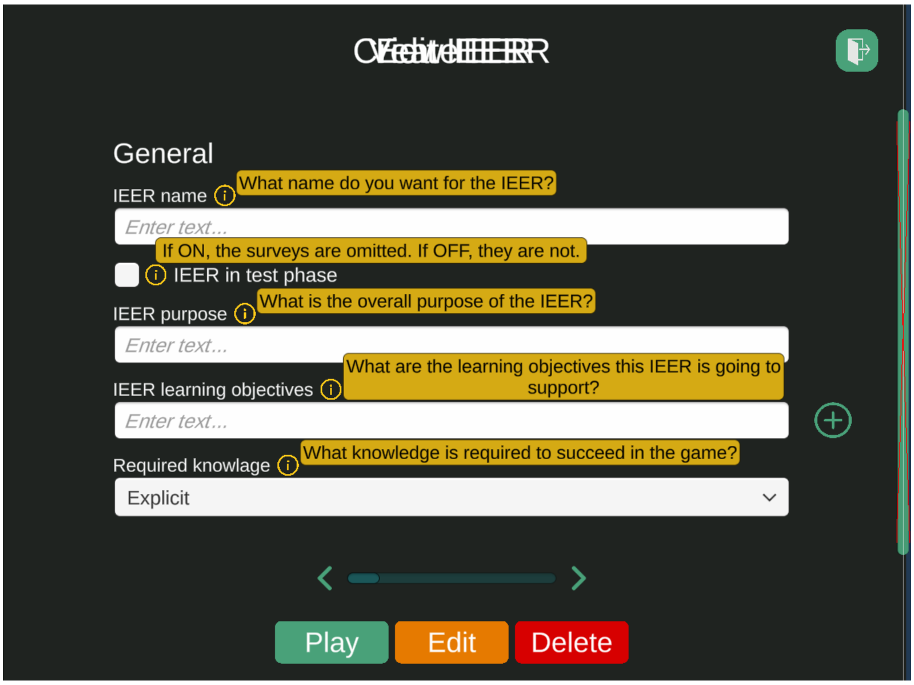


Figure C.22: Prefab of the webview panel

C.23 Backoffice Panel


General

IEER name ⓘ What name do you want for the IEER?

Enter text...

If ON, the surveys are omitted. If OFF, they are not.

□ ⓘ IEER in test phase

IEER purpose ⓘ What is the overall purpose of the IEER?

Enter text...

IEER learning objectives ⓘ What are the learning objectives this IEER is going to support?

Enter text...

Required knowlage ⓘ What knowledge is required to succeed in the game?

Explicit

Figure C.23: Prefab of the backoffice panel