User Engagement in Serious Games

Reno Yuri Camilleri

Supervisor: Dr. Vanessa Camilleri

June 2024

Submitted in partial fulfilment of the requirements
for the degree of Bachelor of Science in Information Technology (Honours)
(Artificial Intelligence).

L-Università ta' Malta
Faculty of Information &
Communication Technology

Abstract

In recent years, the use of games as educational tools, particularly serious games, has garnered significant attention for its ability to enhance student engagement. Now, with advancements in computational power and innovative Artificial Intelligence (AI) techniques, this project aims to create an educational maze game and website that bridges the gap between learning and entertainment.

The core idea behind this project is to investigate the effectiveness of integrating AI techniques to provide users with a more engaging experience. On that end, the AI techniques investigated include: adapting a newly created maze game's difficulty based on user performance as an attempt to bridge the gap between high-scoring and low-scoring students thus creating a more competitive environment that caters to students of all abilities, as well as creating a review system where a user after completing a particular quiz, could test their knowledge with newly created review questions and answers based on the original questions of that particular quiz.

In addition to this, this project empowers teachers with a user-friendly platform where they can effortlessly transform traditional quizzes into captivating, interactive maze games. Using this platform teachers can through the integration of AI, create new wrong answers with a click of a button to simplify their workflow, as well as view student performance in detail such as the speed at which a student managed to answer questions, as well as mistakes made. Meanwhile, students themselves can review their performance post-game and engage in additional exercises to reinforce their learning.

Initial results of the first pilot study indicate that the use of AI in game creation can increase both learner and teacher engagement through a gamified approach to learning. This holds many implications that can be further evaluated and tested in diverse educational settings.

Contents

List of Figures

List of Abbreviations

AI Artificial Intelligence.

API Application Programming Interface.

BKT Bayesian Knowledge Tracing.

CSS Cascading Style Sheets.

GA Genetic Algorithm.

GIF Graphics Interchange Format.

GPT Generative Pre-training Transformer.

HTML Hypertext Markup Language.

IRT Item Response Theory.

LLM Large Language Model.

NPC Non-Player Character.

PC Personal Computer.

PHP Hypertext Preprocessor.

RL Reinforcement Learning.

UI User Interface.

UML Unified Modeling Language.

1 Introduction

In recent years various academics such as K. Squire [1], D. Shaffer [2] and C. Steinkuehler [3] have published various studies on how video games can have a major impact on education. This project focuses on trying to figure out what Artificial Intelligence (AI) algorithms can be used in order to improve or maintain user engagement in games from an educational point of view.

1.1 Problem Definition

Although there are many educational games and gamified educational platforms on the market, this may not be enough to engage users in playing or learning. In fact according to Raptr Inc. a company which tracked more than 23 million gaming sessions for leisure purposes [4] only 10% of users finish a game that they started. This is also backed by a study conducted by E. Bailey and K. Miyata [5] which found that the mean "completion rate" for a sample of 725 games was 14%. With user engagement playing a pivotal role in educational games, this project aims to understand if it is possible to maintain the user's attention through the implementation of an AI algorithm.

AI already plays a crucial role in education today: from robots used to teach children routine tasks, spelling and pronunciation [6] to predictive analysis systems that predict a user's future performance based on their current and past behaviour to provide a better educational experience [6, 7]. While these systems are great in their own right, this project explores what would happen if we were to combine games and AI in order to produce a personalised approach to tackling a student's needs. This creates the problem of how would an AI be integrated to be able to personalise and adapt the learning environment for a student seamlessly.

1.2 Motivation

With the current advancements brought about by the fourth industrial revolution, the roles of teachers and students have drastically changed. Teachers now have a broader knowledge, including insights into how students learn, aided by AI-based learning portals [8] and students having gained more autonomy [8]. These advancements made smart learning environments more common where education is becoming more student-centred, and this allows for gamification in education, which is the process of applying game elements, such as rewards, competition and feedback to enhance learning experiences. A major setback with integrating games within education is the hesitation of teachers [9], this generally occurs due to teachers not being able to assess the content learned from a game [9] (either due to the game being too standardised for their classroom or their needs), as such this project will aim to give teachers a sense of control and agency of the educational content produced by the game.

This project uses a gamified learning environment to investigate the AI techniques that would support user engagement. Serious games are examples of games that are designed for specific purposes (Such as educational), that engage learners in goal-oriented tasks [8]. Serious games have been successfully employed in various contexts, including science education, health professional education, and engineering training [8]. Furthermore, AI has recently begun emerging within serious games to provide reusable AI components across various game engines and platforms which include components such as player modelling, natural language processing, and believable non-playing characters [10]. These aspects made a gamified learning platform ideal for this project, and the serious game created was built upon the framework of an existing serious game [11] that converts classroom quizzes into smart quizzes with gamified characteristics.

1.3 Aims and Objectives

This project aims to create a gamified learning platform with integrated AI-driven techniques that support learner personalisation and content adaptability. The game environment that focuses on problem-solving skills makes use of a Large Language Model (LLM) and Reinforcement Learning (RL) to study changes in user engagement as compared to the same game without the use of any AI. To achieve this aim, the following objectives were identified:

1. Investigate AI models and techniques that can be used to allow for personalisation of content displayed based on user attainments and past achievements.
2. Recreate and improve the environment of a previously designed game that lacked AI features.
3. Integrate the AI models and techniques to offer seamless gameplay in the serious game environment.
4. Build a separate AI layer using a LLM into the existing game architecture that can provide additional feedback on the learner's strengths and weaknesses and offer further personalised tests to aid the learner.
5. Compare a standard version of the Maze Game with an AI-driven version in terms of usability, engagement and performance on problem-solving tests.

1.4 Study Contribution

As of the time of writing, this paper has been accepted for publication by the Future Technologies Conference (FTC 2024) with the title of “Beyond the Maze: How AI Personalizes Learning and Drives Engagement in Educational Games” which will take place on the 14th and 15th of November 2024 in London. This paper will be published in the Springer series ”Lecture Notes in Networks and Systems” and submitted for consideration to Web of Science, SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH and SCImago.

1.5 Document Structure

The rest of this project will be structured as follows: chapter 2 will delve into the project’s background, differentiating between serious and educational games, discuss the serious game which will be used in the gamified learning platform, and the technologies considered for the AI algorithms. Chapter 3 will be a literature review exploring recent work in AI education and gamified environments, as well as a look into forms of user assessment. Chapter 4 will delve into the creation of the project, as well as discuss the design choices made. Chapter 5 will discuss the evaluation of results obtained, and lastly, Chapter 6 will serve as a conclusion of the project.

2 Background Research

2.1 Serious and Educational Games

This project makes use of a game that is both a "serious game" and an "educational game". While these terms are often used interchangeably, they in fact refer to two different types of games. Not all serious games are educational, and not all educational games are serious games.

At present, the academic community has not yet formed a unified definition for what is and what can be considered an educational game [12]. However, it is generally considered that an educational game can be defined as a computer software game that is both fun and educational and can skillfully integrate knowledge with games [12]. This splits educational games into two categories: games with the primary goal of education and games that offer educational aspects within the game. The former of which are all considered to be a subset of serious games. Whereas the latter can be implemented in all types of games, even those that have no real-world objective. For example, the Assassins' Creed games are primarily focused on action and gameplay, but the world of those games are built on actual historical data related to the time period the games take place in, and whenever a historic location or event is encountered in the game, the game gives you the ability to pause and read real-world facts on the location/event. This is done to such a good degree that there are even studies [13, 14] that attempt to use the game as a medium for teaching students.

A serious game on the other hand can be defined as a game designed for a specific purpose that engages learners in goal-oriented tasks and offers benefits such as interactivity and feedback [8, 15]. Furthermore, serious games can be used for various purposes, including education, health, recruitment, attitude change, and awareness raising [15]. Therefore, a serious game does not necessarily have to be an educational game, as education does not have to be the primary focus. Some studies that illustrate this include the use of serious games in a corporate environment [16] and serious games used in environmental awareness [17].

Various serious games and studies use a variety of different methods to measure engagement [18]. However, two very important aspects to analyse within serious games as defined by previous studies [18, 19] are the "game aspect" and the "serious aspect". The game aspect refers to how the serious game performed as a "game" meaning the objective is to analyse how much the users enjoy playing the game, whereas the serious aspect refers to analysing if the user managed to obtain the knowledge, or skills that the serious game should have given to the user. These aspects are both equally important as the goal of a serious game should be to give the user a certain benefit in a fun manner.

2.2 The Maze Game as a Foundation

The University of Malta's "Maze Game" [11] is a serious and educational web-based game that allows teachers to engage their class with an interactive quiz environment, it is similar to other quiz-like web games such as "Kahoot", "Quizizz" as well as "Quizalize" that offer gamified ways of tackling a classroom quiz. The reason the Maze Game was chosen as a platform instead of its counterparts was due to it already being a product of the University of Malta, meaning it was easy to get hold of and work with all of the resources of the game.

The Maze Game was originally developed in 2022 by G. Borg as an individually assigned project to create a maze-like educational game in Unity. The game worked by having a proper traditional maze-like structure, and the player while playing in first-person would traverse the maze. Each time the player would decide which path to turn towards, there would be text stating a question and two possible answers. The pathway with the correct answer would lead further towards the exit, whereas the pathway with the wrong answer would delay the user by having them go through a long empty pathway, till they eventually reach a dead end informing them that the pathway was incorrect.

The Maze Game was then further developed in 2023 by three students: D. Sciberras, R. Y. Camilleri and E. Gatt, who recreated the game from the ground up, turning the game into a web game to allow a greater audience of users to have access to the game, as opposed to the original game which only targeted windows computer users. This also gave the game the benefit of allowing teachers with no knowledge of the Unity engine to be able to use a simple web interface to insert their own questions into the Maze Game. The students also added seed-based procedural generation to generate pathways for each question, ensuring that a unique maze would be made for any new quiz generated via the teacher's questions, whereas students of the same quiz would face the same maze structure. Similarly to the first game, the objective of this game is to finish the game as soon as possible, picking a correct option would lead you closer to the exit, whereas picking a wrong option would lead you to a dead end. The final build of the "Maze Game" can be found on the Upskills Project website [11].

In regards to the second objective of this project, while the Maze Game already contains several aspects that are desired for this project, such as an easily accessible web page for teachers to create questions, as well as procedural generation of questions and answers. It was designed around the idea of giving users a question and two choices, the correct answer and the incorrect answer. All the aspects of the game are designed in a way as to follow that structure. Furthermore, the code-base of the Maze Game was not made in an adaptable format, as a result, it is hard to work with the original code-base, so while most of the ideas of the Maze Game and various aspects of it can be re-used for this project, the project will be mainly creating its own new code and system.

2.3 AI Technologies for User Adaptation

In regards to the first objective of this project, there is a large number of different AI algorithms that can be used to adapt to a user such as: Item Response Theory (IRT), Bayesian Knowledge Tracing (BKT) and RL. For this study, RL in particular will be analysed and used throughout the project.

2.3.1 Item Response Theory

Item Response Theory is a statistical model commonly used in educational and psychological assessments. In a quiz game environment, IRT can be applied to assess a user's ability level and the difficulty of the questions. IRT assumes that the probability of a correct response depends on both the user's ability as well as the item's difficulty [20]. In other words, to calculate the probability of student i, getting question j correct is:

$$P(\text{correct}_{i,j}) = \frac{1}{1 + e^{-(\theta_i - \beta_j)}} \tag{2.1}$$

where:

θ_i is the ability of player i,

β_j is the difficulty parameter of question j,

e is the base of the natural logarithm.

2.3.2 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing is a probability-based model, that tracks a user's knowledge state over time. In a quiz setting, BKT can be employed to model how a player's knowledge evolves based on their responses to different question types to check if the user has mastered a skill [21]. The model works by calculating the performance of the student up to a certain point with the model parameters. A study [21] using BKT found that students were more engaged and interacted more using a BKT approach.

2.3.3 Reinforcement Learning

RL is a machine learning algorithm where an agent learns to make decisions based on interactions with the environment, and obtaining rewards from those interactions. RL is one of the most popular machine learning algorithms and has been used in various domains such as in successful integration to board games such as backgammon [22, 23], checkers [22, 24] and Go [22, 25]. As well as domains such as robotic soccer [22, 26]. It has also been used in past quiz environments such as the study by S. Liu [27] recommending the adaption of RL systems in education.

In this project, RL will be used to adapt difficulty based on player performance. The main reason behind this choice is that whilst IRT is mainly used for educational assessments rather than for games, BKT is most used in adaptive learning systems such as for instance improving safety training outcomes in construction [28] and intelligent tutoring systems [29]. RL seems to be the better option because in recent years it has gained popularity in adaptive learning for the video game industry [23–27, 30, 31].

2.4 Summary

This chapter explores what makes a game serious and what makes a game educational, distinguishing between them and highlighting their potential in various fields. This chapter also introduces the Maze Game, initially developed as a Unity-based project and later transformed into a web-based platform by students, offering an interactive quiz environment for students and educators. Additionally, this chapter delves into AI technologies such as IRT, BKT, and RL, examining their applications in assessing user abilities and adapting educational experiences based on performance. These components form the groundwork for the project's objectives, aiming to create an adaptable and engaging educational gaming platform.

3 Literature Review

3.1 Emerging AI in Education

Artificial Intelligence has completely changed education as we know it and will become a significant part of the future. F. Almedia [8] describes education 4.0 as the next step in education. Education 4.0 is a new educational paradigm designed to address the needs and potentialities of the fourth industrial revolution. Education 4.0 encourages students to learn through experimentation, and Almedia [8] states that this can be done through the inclusion of games, both in a classroom environment and through a remote environment.

A. Alam [32] also seems to agree that AI is the future of education, Alam delves deep into the practical applications of AI within education, both in assisting students and teachers, but also in creating smart campus administration [6, 32], through the use of facial recognition for instance. In addition to all this with the rising popularity of LLMs, many different researchers are widely exploring smart tutoring systems [29, 32–35] which provide students with the benefit of learning in a one to one environment which is adaptive, and can provide immediate feedback without the need of a human teachers' intervention [33–35]. This allows students the ability to take control of their own learning and perform at their own pace.

Alam states that AI tutors are necessary as there are too many students for instance in English courses as compared to teachers, as such teachers can not give personalised assistance to each student, whereas an intelligent system can. To back up his claim he mentions "Duolingo" a multilingual learning platform that provides each user with personalised content to help each user learn a language through a game-like structure. Alam mentions that students who used Duolingo as a platform managed to perform better on standardised exams as compared to students who did not [32].

A. Alam also mentions using AI technologies such as image recognition and prediction algorithms to simplify student performance evaluation [6, 32], as education is highly focused on students taking tests, as such traditionally teachers would spend a significant amount of time creating and grading questions, whereas AI can not only grade work in a short amount of time, but it can perform its own evaluation by creating unique questions and even providing feedback after grading.

3.2 Performance-Based Game Adaptivity

As a user progresses through any game environment it is necessary for the game to adapt its difficulty to the user based on their different skills, and on how capable they are at learning and adapting over time [22]. When it comes to changing game difficulty over time there are two primary approaches which games tend to use, either a linear / non-adaptive approach where a game's difficulty either stays the same, or linearly becomes harder over time irrespective of how the user is performing, or through an adaptive approach where the player's performance is analysed and the game adapts based on the performance [22, 36].

While making a non-adaptive approach is easier than having to create AI algorithms that adapt to a user's performance, various studies [32, 36, 37] have shown that an adaptive approach is more effective at aiding a student in their education. In the first study on "employing adaptive learning and intelligent tutoring to virtual classrooms" [32], A. Alam states that students perform better and are more likely to remember what they learned when in an adaptive environment.

In the second study by S. Sampayo-Vargas performed a research experiment [36] where the same game was made with two versions, one with a linear approach, and one making use of a scaffolding [36, 38] adaptive-difficulty approach. The scaffolding adaptive approach worked by having a system where the user had to demonstrate mastering a particular level of content before a "scaffold" was removed and would move on to the next level. Similarly, if the student performed badly, a new "scaffold" would be put into place lowering the student's current level. At the end of the study [36] it was found that the adaptive game group managed to obtain significantly higher results than those in the incremental game.

In the third study by T. Jagušt, competitive, collaborative and adaptive gamification techniques were investigated in a mathematical setting [37]. In their experiment they incorporated a game where a user had to solve mathematical questions to defeat a "virus-like" enemy, together with this they also experimented by having one group of students learn in a typical classroom environment without making use of the game. The game had several different modes to experiment on what worked best, one of the modes employed a personalised adaptive algorithm which calculated how long it was taking each student to solve a problem, as well as if the student or virus was winning. The algorithm would then reduce the time by one second if the student was winning, or otherwise, it would give the student more time if the virus was winning. This meant that the students would always be kept on edge of their limits. The results obtained from the experiment [37] showed an improvement in all three gamification conditions as opposed to the regular classroom environment, with the adaptive condition coming with the highest average number of attempts made by students, as well as with the best overall performance.

It is worth noting, that educational games are not the only types of games that employ an adaptive game difficulty style with a variety of games using adaptive difficulty techniques to ensure a user is always engaged [39]. This extends to a variety of genres from action games that ensure that the AI doesn't instantly defeat a user in combat, to even horror games such as "Alien Isolation" using a monster AI that adapts based on how the user plays the game, to ensure the user uses different methods of sneaking past the monster and is always kept on edge [39]. A common method mentioned by G. Andrade [22] when it comes to game balancing is through the use of Genetic Algorithm (GA). However, GAs take a long while to compute and are more effective in Non-Player Character (NPC)-like enemies rather than in a quiz game environment. Hence, Andrade [22] mentions the use of RL algorithms that can be used with great efficiency in a large variety of games such as for instance backgammon, checkers and Go.

3.3 User Assessment and Feedback

The second phase of the project is giving users post-quiz exercises, these exercises are very important as the AI agent would inform the students about their weak areas, and if the users decide to opt into the second phase, they would be given supplementary questions ranging from simple to more complex in order to consolidate their knowledge.

P. Brusilvosky [40] in their study of individualised exercises mentions that one way of creating post-quiz exercises is through the use of "parameterised questions and exercises". Parameterised questions are question templates generated by the author, or for instance a teacher. These templates are then used with varying parameters to create a massive sum of questions using only a few templates, which can then create a very large number of unique questions for each student to practice. This system works well as it could for instance simplify the work done by the teacher, by using the initial quiz questions made by the teacher as the parameterised question templates. Since the main objective of post-quiz exercises is to provide students with a method of self-assessment and improvement, cheating is not an issue [40].

A more recent study [38] utilised a Deep Q-Network RL-based AI implementation using a rule-based decision-making strategy. In the study [38] making use of an adaptive feedback system through hints, attempts, and feedback messages were mentioned. This feedback system was shown to be quite effective, as such using the aforementioned 3 systems in a quiz-like environment could be implemented by giving the user a newly generated question with a hint of what the correct answer is, a retry system after getting a question wrong, as well as educational feedback telling the student why they got a question wrong.

The main problem with using parameterised questions in post-quiz exercises is the evaluation of a user's answer [40]. Depending on what type of subject is being assessed sometimes a question might have more than one correct answer depending on the context, for instance, if the question "Who discovered America?" were to be asked in a history quiz both the answers “Christopher Columbus” and “Leif Erikson” might be correct [41] depending on further context. Similarly, a randomly generated question might accidentally not generate a correct answer or have a correct answer that is not entirely correct. The result of this issue is that most post-quiz applications that go for this approach tend to be developed for physics or math-related subjects [40], where correct answers can be calculated using a formula with the question’s parameters.

3.4 Ethics in Educational Game Design

R. McCall [42] discusses in detail the importance of ethics in serious games with reference to the American Psychological Association (APA) guidelines. These discuss the principles of consent, risk and harm, coercion, privacy, transparency and trust. To combat these issues games need to maintain a level of user privacy and maintain only necessary data which should be secured. It is also very important for games to be transparent about the use of user data and what data will be collected as serious games generally obscure their true objectives from users [42].

Even when users initially agree to data collection, problems can occur if updates or disclosures about the data collected differ from initial user expectations, resulting in mistrust [42]. Therefore, consent should be a continuous process, with users regularly informed about and consenting to data usage. Providing users with access to their data and examples of its potential use can help maintain transparency and trust [42]. This project aims to be as transparent as possible regarding user data and what data is collected.

3.5 Summary

This chapter delves into how AI is reshaping education through Education 4.0, emphasising interactive and personalised learning. Advocates like F. Almedia and A. Alam highlight AI’s role in personalised learning, smart campus management, and streamlined assessment using technologies like facial recognition and adaptive tutoring systems. This chapter also delves into performance-based game adaptivity, demonstrated by researchers A. Alam, S. Sampayo-Vargas, and T. Jagušt, which enhance learning outcomes with tailored approaches. User assessment and feedback are also discussed, and explored by researchers P. Brusilvosky and W. S. Sayed, which offer users personalised experiences, though challenges persist in evaluating context-dependent responses.

4 Methodology

This chapter provides a complete overview of the decisions made, design and implementation of this project in depth, an explanation will be given as to why certain decisions were made. Objective 1 was investigated in the background research and literature review, whereas this chapter will focus on objectives 2 - 4, with objective 5 being covered in the next chapter. Due to this project having many visual design features, only a few images will be shown within this chapter, with the rest being available in the Appendices.

The creation of the serious game and the website were split into two sections due to their complexity. The game was developed using the Unity engine (Long Term Support - Version 2022.3.20f1). The website was developed using XAMPP (Version 3.3.0). XAMPP is a free and open-source web server package developed by Apache Friends and consists of both the Apache HTTP Server as well as MySQL and various other packages. For this project, only MySQL and Apache are used from the XAMPP package. Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript and Hypertext Preprocessor (PHP) are used for the website.

4.1 Usability Design

This project is intended to assist both teachers as well as students with having a simple yet effective system in place to create interactive gamified quizzes and play them. While this project only defines one class of a user within the program itself, that has access to both the systems in place for students as well as those in place for teachers, the needs of teachers as well as students are vastly different. As a result the following Unified Modeling Language (UML) - Use Case Diagram was devised to compare and contrast the systems that will be in place for both parties.

Within the UML - Use Case Diagram we will be considering the database and the Generative Pre-training Transformer (GPT) 3.5 Application Programming Interface (API) as external entities. This project has its own login and sign-up system in place, and most of the application relies on the user being logged in, to view and access the majority of the features. In addition to what can be seen in the diagram, there is also an error 404 page that is shown when a user tries accessing a page that does not exist.

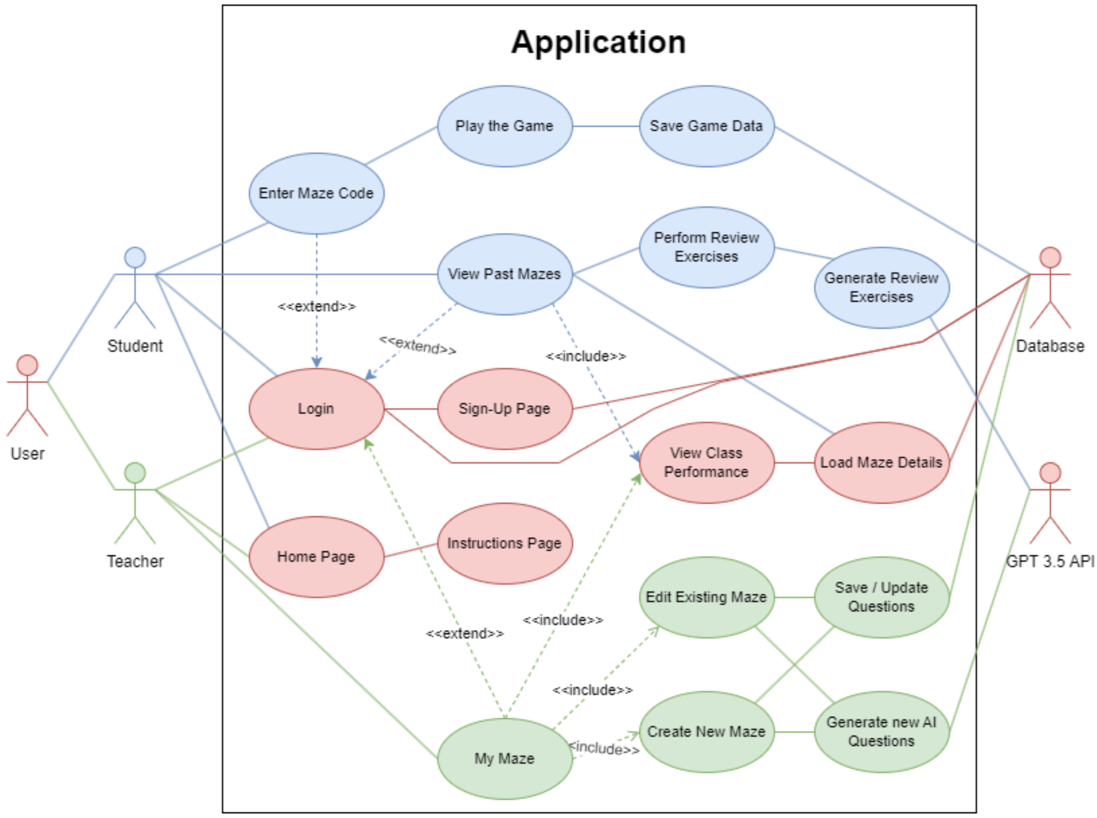


Figure 4.1 Use Case Diagram of the System Design

Teachers

After logging in a teacher will be primarily using the “My Maze” page which will be discussed in detail in the website creation section within this chapter. The My Maze page acts as a system where teachers can manage all their mazes. From this page a teacher will be able to view all their existing mazes, create new mazes, and edit existing mazes by either changing questions, adding new questions or even removing old questions. The teacher will also be able to delete any no longer necessary maze, and will also be able to acquire a particular maze’s access code to share with students and view the class performance of their students.

Looking into user design, to make this system more beneficial for teachers, the following three mechanics will be implemented to make the application more teacher-friendly: firstly, the maze results will feature the score of every student, as well as award them with badges to show the teacher information such as which student got the least number of mistakes, and which students were the fastest. Secondly, if the use of AI is enabled by the teacher, an additional fourth wrong answer for each question will be generated by the AI based on the structure of a teacher’s wrong answers. That way a teacher won’t have to go through the effort of creating even more answers for each question. Thirdly, the post-quiz exercises will be entirely managed by an AI system to allow students to have a diverse set of questions to review their knowledge learnt, whilst allowing the teacher to avoid all the hassle of creating a ton of unique review exercises.

Students

After logging in students will primarily use the "History" and "Maze Game" pages. The Maze Game page will first require a student to input a code of a particular maze to access the game, then they are able to play the game. A student can play the same Maze Game as many times as they would like. However, the attempt numbers are saved and shown on the class leaderboard. Within the History page, a student would be able to review each maze they have played, as well as all their attempts at beating the same maze. For each attempt, a student would be able to view their score, the top score, the class rankings, the list of all questions, and answers, and whether or not they got the question correct. There will also be graphs to show the overall score against time, as well as the time taken per question. If the teacher enabled the use of AI, students would also be able to use this page to practice and review the questions they went through within the maze.

Looking into user design, most of the AI systems will be included within the Maze Game itself. If AI is enabled then the game would adapt the difficulty based on user performance between high-performing and under-performing students. This difficulty adjustment would allow high-performing students to still face a challenge, whilst allowing the underperforming students a challenging yet not too difficult experience. Furthermore, AI will be used within the review exercises to ensure students always find new unique questions to practice what they learned. Review exercises will act as an educational tool, so the AI in addition to adapting review question difficulty for students, will also provide hints that a student may view if they wish, as well as explain why a particular answer is correct, or why it is wrong.

4.2 Data Management and Storage Ethics

A lot of care was placed in data management to safeguard any ethical concerns that could arise from this project. In terms of the collection of data, within the game and website, the only data that is saved is the user's display name, username and password. No identifying features regarding any user are present within the project, and care was placed to safeguard usernames and passwords. In addition to this, within the user evaluation that was carried out (more information later) the only identifying piece of information collected were the user's consent forms stating that they accepted to take part in this project. These consent forms will only be kept until the end of this study, and all data given apart from the results obtained (where the users have been anonymised) will be deleted at the end of the publication of this study.

4.3 Creating a new Serious Game

As mentioned in the Background Research, the Maze Game was developed with the intent of having a structure of a question, and two answers: the correct answer and the incorrect answer. Due to the code base being built specifically to support that structure, a lot of features need to be adjusted or changed. Furthermore, the original version of the game lacks a lot of needed features such as tagging users that pick the wrong option, unoptimised procedural generation, and immediate procedural generation (Which wouldn't allow the AI system to adapt continuously based on performance).

The original Maze Game also has a very simplistic and outdated User Interface (UI), and the website that is connected to the Maze Game only gives teachers the option to create mazes with right and wrong answers and then submit the quiz to be converted to a game, you could not check what your past quizzes were like, the performance of students or even edit questions. As such the website as well as the game will mainly be remade from scratch, to create a new and improved version of the original game for this project. From this moment onwards when referring to the project, it will be referred to as the "Improved Maze Game", whereas "Maze Game" will be used when discussing the original version.

This section will provide a basic overview of each aspect of the serious game, for a more detailed overview of the implementation of the Improved Maze Game, refer to Appendix A.

4.3.1 Reused Assets

While most aspects of the Maze Game will be remade, some standalone systems can be carried over to the Improved Maze Game without needing to be changed, such as the music, movement script and camera script.

Music

The Maze Game features five different music songs, of varying lengths and volumes. In the Maze Game only two of those five songs are implemented and used. Due to the unnecessary file space taken up by the music, in the Improved Maze Game, only the two used songs are kept, and these are managed by two systems: the "Music Manager" that holds the menu music playing on a loop and the "Game Manager" object in the main scene.

Two additional scripts were made for this project in relevance to the music which were the “VolumeController.cs” and “InGame_Music.cs” files that can be found in the Start and Game scenes respectfully. These scripts allow the user to set a music volume using a slider in the settings, and the volume is saved in between scenes.

Movement and Camera

The movement and camera scripts were re-used within the Improved Maze Game with no changes made. The movement script features a basic first-person movement script that allows movement either using “WASD” or the arrow keys with a jumping mechanic whenever the player presses the spacebar key. The camera script also features first-person camera movement using the mouse. Just like in the Maze Game, the movement scripts and the camera script were attached to a capsule game object named “Player”.

4.3.2 Start Menu

Figure 4.2 New User Menu Design

The Improved Maze Game is split across 3 scenes, the start scene, the game scene and the end scene. Within the start scene, there is the main menu. The Improved Maze Game contains all the same four options from the Maze Game but with a new more professional-looking design. When making the design care was placed to ensure that all sizes within the interface scaled based on the height of the user screen. Since this project is mainly aimed at Personal Computer (PC) and laptop users, throughout the design and testing process the UI was always tested with a 16:9 aspect ratio (the most common ratio for desktop monitors). Apart from the figure above, a before vs after set of images can be found in Appendix A.

The main menu is composed of 3 main frames which are: the main frame that can be seen in the figure above. The settings frame which shows all the controls of the game as well as includes the volume slider that was mentioned in the music section above, and the credits frame which contains the list of the developers that worked on the Maze Game.

4.3.3 Prefabs

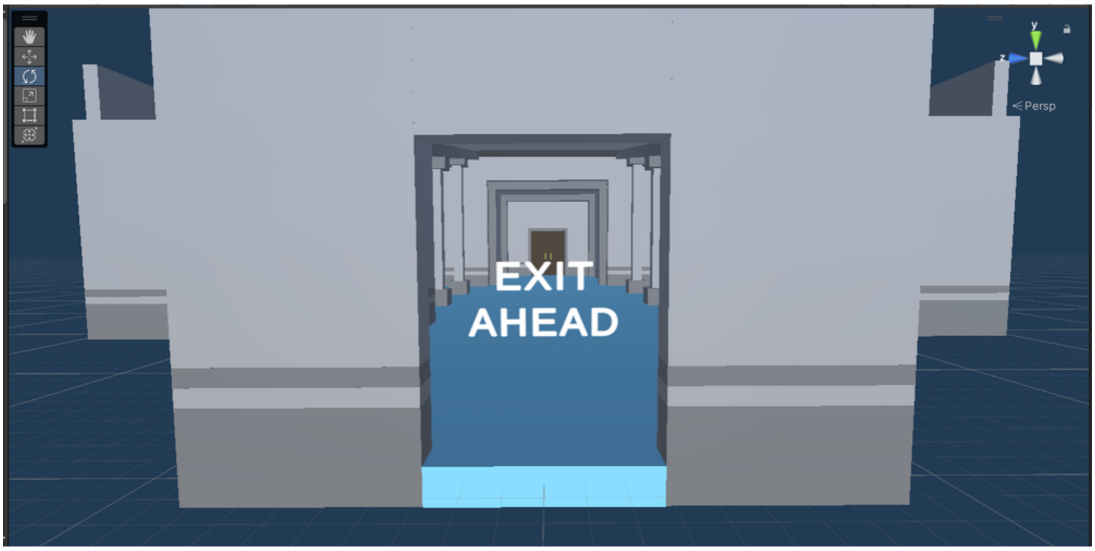


Figure 4.3 Design of the Final Room Prefab

Start and End Room

When moving to the game scene, the first room the user will encounter would be the Start room. The gameplay actually starts from this room onwards as such it is important to ensure that the user faces an aesthetically nice and consistent room design within the entirety of the maze. In this project, we are working with a maze, but there are a variety of different types of mazes. For this project, the Start room as well as all the other rooms within the maze were made to have a modern "Escape Room" style. This style was chosen to replicate the same style used within the Maze Game.

Question Rooms

The Question Rooms within the maze act as large rooms with a single entrance and multiple exits. The number of exits determines the number of possible answers a user has for a particular question given to them. For this project three different types of

Question Rooms were implemented, one with 3 exit choices, one with 4 exit choices and one with 5 exit choices. The AI will be making use of these 3 different choice rooms to adjust difficulty once implemented. In terms of design, a similar design was used to that of the Maze Game, where each room is large and open and has a carpet leading to all the possible answers to make it simple for a user to notice the number of pathways. An invisible wall that can be passed through is placed in each Question Room close to the entrance, the wall with the use of the "Questionroom_Tagger.cs" is in charge of detecting when a player collides with it, and in such an event is used by other scripts to perform various functions.

Figure 4.4 Example of a Question Room with 4 Choices

In terms of script design, all question rooms make use of a "Choice_Prefab.cs" script. This script features references to the "start" and "choices[]" game object transformations, in addition to holding a reference to the user tagger game object and a question string. The start represents the entrance to the question room and is used by the procedural generator to align the start positions of each room with the exit positions of other rooms. To avoid visual bugs, it is important that all start and exit game objects are of equal size. When it comes to the choices parts, it is important to distance them by a reasonable amount to allow space for pathways to be spawned.

When the user tagger is triggered, the tagger game object is immediately deleted to avoid it from being triggered a second time, and then the player's movement speed gets set to zero to freeze them in place. As seen in the previous figure, the Questions themselves are placed inside the UI for better readability, this UI is managed by a "Question_UI.cs" script which simply changes the text on the UI when activated, it also displays every second the amount of time the user has before they get unfrozen. Freezing the user acts as a way to give users reading time, so both fast readers and slow readers have time to read the question before deciding on an answer. Once the freeze time is over, the user's speed is restored. This function also is responsible for assisting the Procedural Generation and the AI systems, more details can be found in Appendix A.

Pathways

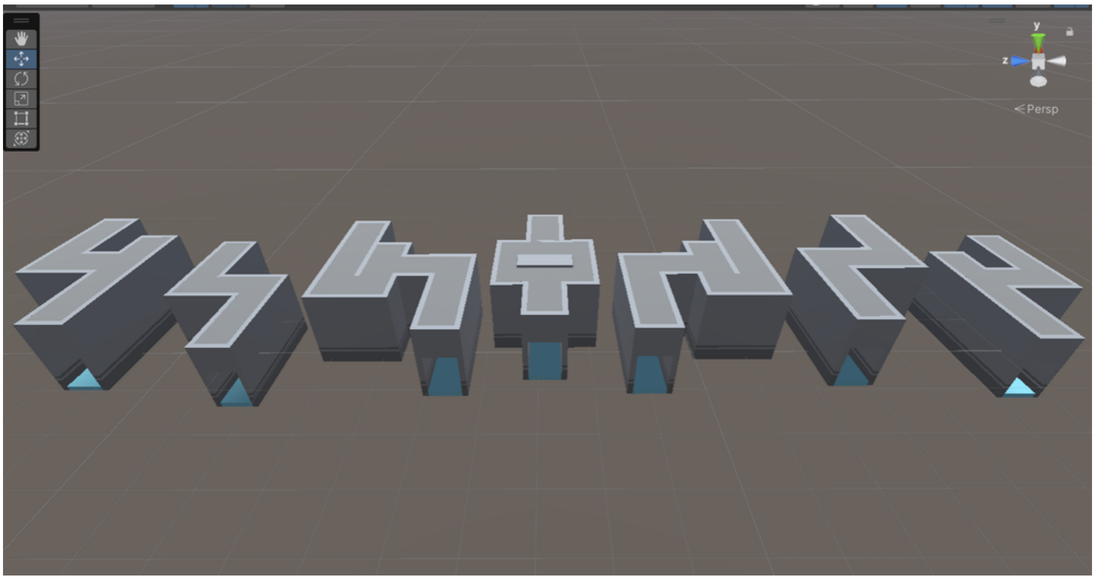


Figure 4.5 All of the different pathways placed next to each other

Pathways act as corridors to move from one question room to another. They also serve the purpose of making the game feel like a proper maze. When it comes to the design of Pathways, this project splits them into three categories, pathways facing the left, pathways facing the middle, and pathways facing the right. All three different pathway types are used to generate the maze without collisions. Just like Question Rooms, pathways feature a start and end point, as well as an invisible user tagger. In addition to this, the Pathway prefabs also have an invisible part that can be passed through at the very entrance that features a text UI that showcases the answer placed on the pathway. Since the pathway starts will be aligned to the exits of the Question Rooms, the text will appear right in front of every exit.

In terms of design it is important that from the entrance of a pathway, the user should not be able to see the exit as that would be a giveaway if a pathway is correct or wrong. In addition to this, the user tagger should be placed in a position where the user would only touch it as soon as they are about to see the exit of the pathway. Thus tagging a user just at the moment when they realise if a pathway is correct or not. Lastly, the pathway prefab should have a bounding box collider that a user can pass through, that covers the entire prefab. This bounding box will be used in the procedural generation to avoid different pathways colliding with one another.

The Pathways use a "Pathway_Prefab.cs" script that features the start, end and user tagger parts. In addition, it also features a reference to the text UI of the corridor, as well as two boolean values "userTaggerEnabled" and "correctAnswer". The prior is set to true by default and is only disabled in the very first corridor of the Improved Maze Game, whilst the "correctAnswer" boolean is set to true when the corridor has the correct answer, and false when it has the incorrect answer. When a user is tagged, the tagging part is deleted, and then depending on if the pathway had the correct answer or not, the pathway prefab will either request the procedural generation to generate the next question room and set of pathways, or it would alternatively send a request to the AI to save the wrong answer, as well as send a request to the score manager to deduct points.

End Walls

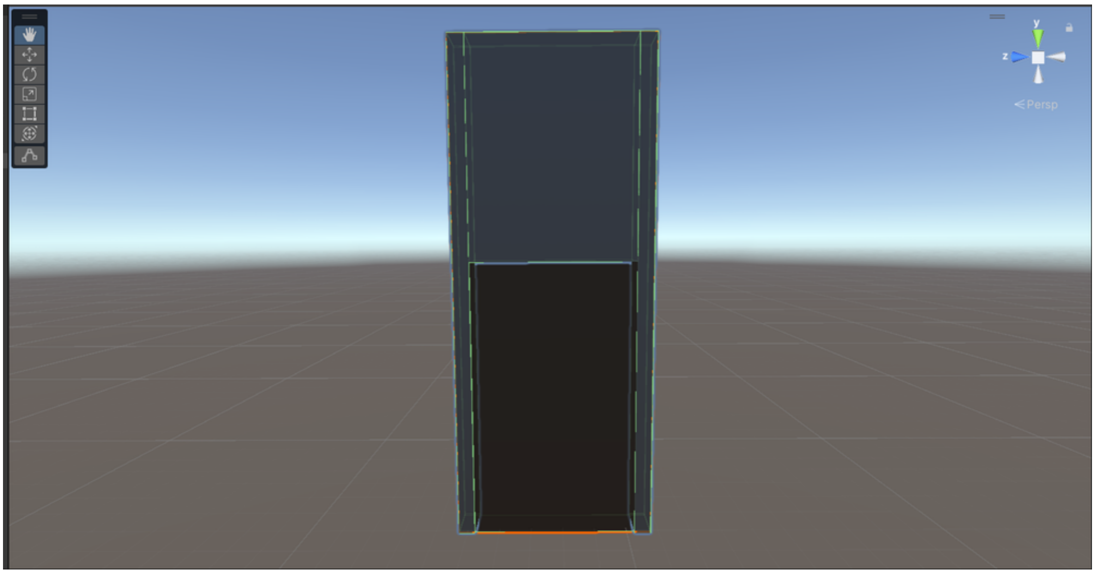


Figure 4.6 Design of the End Wall

The End walls were designed to look like closed doors, the main objective of End walls is to be placed at entrances of question rooms once a user has triggered the question room and become frozen. This way the user would not be able to return to a previous question room. The main purpose behind this is saving memory space since previous question rooms will become inaccessible, they would be able to be removed to save space. Additionally, End walls are placed at the end of every pathway prefab that does not lead to a question room.

4.3.4 Procedural Generation

The Procedural Generation within this game is controlled by a script titled "Maze_Generator.cs", within the game this script was attached to an empty game object called "GameManager". This script holds multiple public object references which need to be set by the developers such as a reference to the Start Room game object within the scene, a reference to the Final Room prefab, a reference to the End Wall, a reference to the Time Controller script, as well as list references to the number of Question Rooms, Left Pathways and Right Pathways available as well as references to their actual prefabs. In addition, there is also a reference to the Centre Pathway.

For the first maze section generation, a "Begin()" function was implemented, it is important not to use Unity's built-in "Start()" function, as the procedural generation should not run immediately, the goal is to make the procedural generation a stand-alone script that is managed by an exterior script, in this project's case it will be managed with a RL algorithm later on. As such the maze generation should only begin once the algorithm loads all the necessary data it needs to load, and is ready to launch the maze.

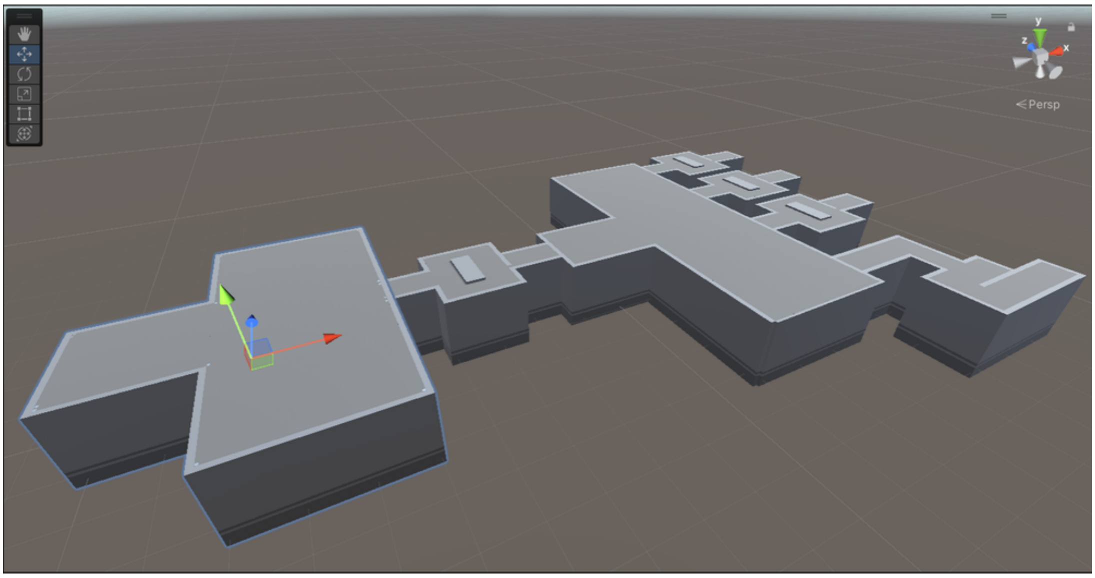


Figure 4.7 Generation of the First Question Room

Loading New Rooms

New Question Rooms and pathways are loaded together using the previous correct pathway's end point as a point of reference for positioning the next rooms. Once a collision occurs between the user and the user tagger in a pathway, the Score Manager is alerted to run a function called "RequestNewMaze()" or "ReceiveWrongAnswer()" depending on if the prefab has the correct answer or not, the Score Manager would then be responsible for dealing with the user score and running a function within the AI algorithm to request the Maze Generator to Generate the next maze.

The question room generation works as follows: data is sent to the "GenerateNextMaze()" function, this function then spawns a question room that fits the number of questions passed (As such it is up to the AI algorithm to manage the number of wrong answers passed on). Once the question room is aligned, the question variable within the question room prefab is set to the passed question. A random integer between 0 and the number of questions is then generated, which will represent the pathway that will contain the correct answer. Afterwards, the "AddCorridors()" function is run which is in charge of loading new pathways.

Pathway loading works as follows: first, the midpoint of the number of pathways is found. Then for every exit point within the question room starting from the leftmost room till the middle, there is a 2/3 chance of spawning a random type of left pathway, or else there is a 1/3 chance of spawning a centre pathway. Afterwards, the pathway spawned is checked to see if it collides with the previous pathway spawned. If there is a collision then the generation script will try up to 100 times to spawn a different left or centre pathway that does not collide with the previous pathway. If after 100 attempts such a pathway is not found, a centre pathway (Which should NEVER be wide enough to collide with other pathways) is spawned. Afterwards, this process is repeated starting from the very right down till the middle, and this time either using the centre pathway or the right pathway, with a bias towards the right. The reason for this design is to further prevent collisions, by starting with left pathways from the left to the middle, then moving to right pathways from the right to the middle, the chances of running into a collision are significantly reduced.

After each pathway is placed successfully if the pathway contains the correct answer (Which can be found by comparing the index to the random index) then the pathway is validated using a "ValidateCorridor()" function, otherwise, it is invalidated using an "InvalidateCorridor()" function. In the prior case, the pathway prefab is marked to contain the correct answer, and the answer is displayed on the prefab entrance, in the latter case, the prefab also contains the wrong answer displayed, but it has an End wall placed at the very end.

Removing Old Rooms

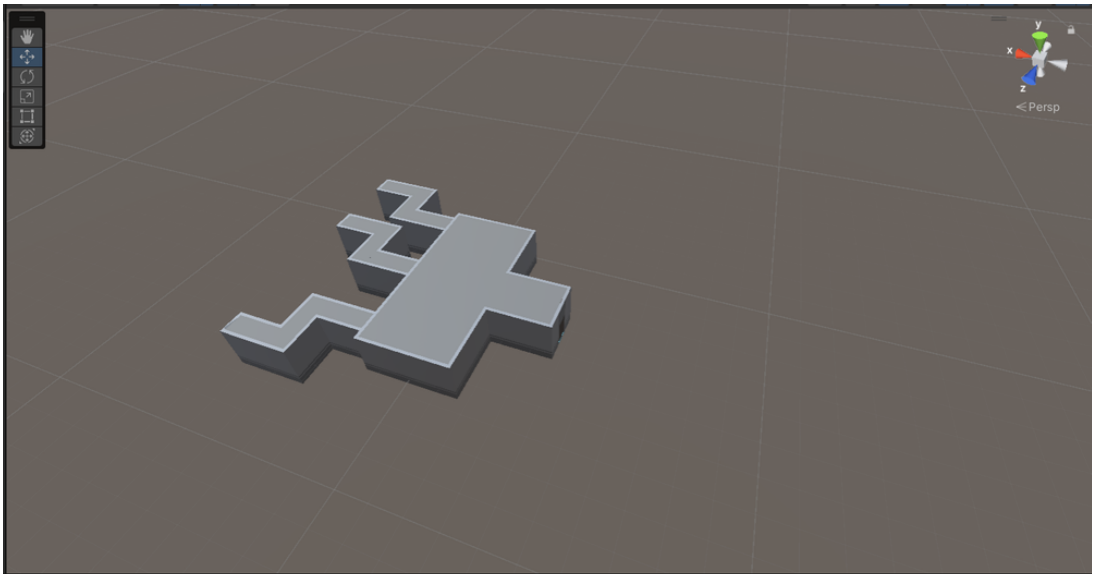


Figure 4.8 Old Prefabs are Removed to Save Space

An issue with the Maze Game was that in some cases when the mazes got quite large, the game would start lagging. To avoid facing this issue, once a user completes a question, the question room entrance gets locked off with an End wall. Following this, old question rooms and their respective corridors are de-spawned to save space.

To achieve this, the Maze Generator has a list of a list of objects that contains all the items that need to be de-spawned. Each time a question room, corridor, or end wall is spawned, they are added to the list of list of objects, and each time a new question room is reached, the parent list is incremented. That way once older rooms are closed off, the previous lists of objects (excluding the current list of objects that would include the current question room and its corridors) would have the objects being references deleted.

4.3.5 Score Manager

The Score Manager is a script within the Game Manager game object that keeps track of the user's score and displays the score at the top-left of the screen. Apart from decrementing this score every second, this script also halves the current number of points the user has when a wrong answer is chosen. While other similar smart quiz games like Kahoot for instance opt to give users zero points when a wrong answer is picked, as stated previously this project wishes to explore methods of increased engagement through the normalisation of student results, as such by halving the number of points a user current has instead of setting it to zero, the user still has a chance to climb up the leader board.

The Score Manager is also responsible for requesting new mazes, sending the current score and sending the total score to the AI algorithm. The reason behind this decision is that each time a correct answer is chosen, the AI algorithm would generally need score information in order to understand the performance of a player within that particular question and update its information to adjust the maze accordingly.

4.3.6 Timer

The Timer script is responsible for showing the current time in the format of "MINUTES: SECONDS" at the top of the screen so a user would know how long they spent playing the Improved Maze Game. The Timer holds the following variables: "totalTime" which holds the total time that the user has spent within the game, a "timeStamp" a floating-point variable that holds the time stamp of when the last question was completed and is updated each time a question is answered correctly, thus getting information on how long it took to complete a question and a "timerActive" variable which indicates if the timer should update or not.

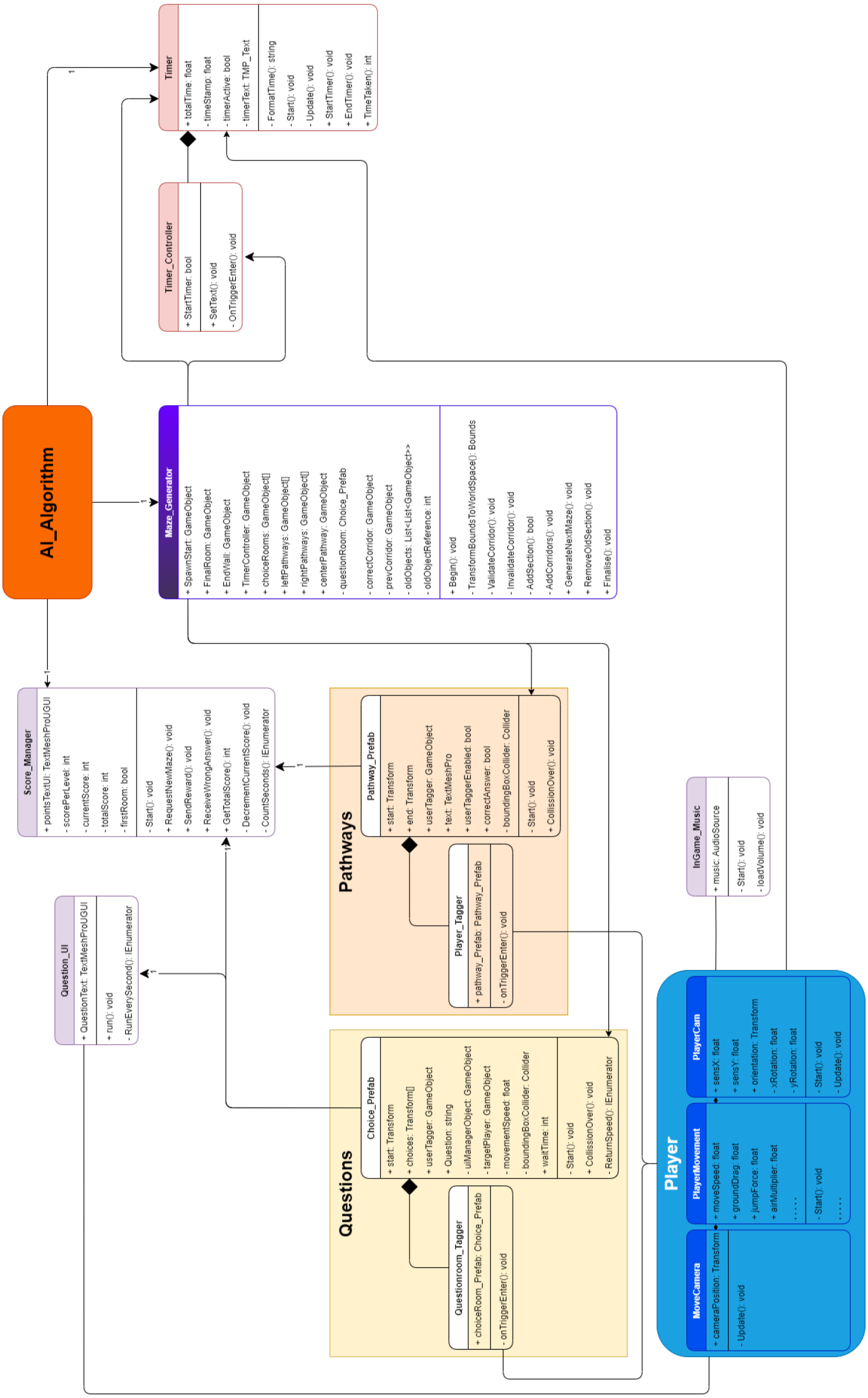
4.3.7 Ending the Game

Once the final question is reached, the AI algorithm is responsible for running the "Finalise()" function within the Maze Generator. This causes the maze generator to place a Final Room instead of a question room with corridors at the end of the last correct answer within a maze. Once the user gets tagged by the user tagger in the final room, the AI is responsible for saving all the information needed, and sending it to the web server, afterwards the total score obtained by the user is saved within a Unity Player Pref and the user is moved onto the next scene, which is the End Screen.

End Screen

The End Screen features a simple UI that congratulates the user for completing the Improved Maze Game and shows the score obtained by the user. After a period of time (Within this project 12 seconds) an auto-exit function is run that runs a JavaScript function "javaScriptGameFinished()", the exit functionality is then managed by the website, where in this case the user would be redirected to a performance page to view their right and wrong answers.

4.3.8 UML Class Diagram of the System



4.4 Creating a new Website

The original Maze Game web page featured an index page that scaled based on screen width and featured a lot of fixed options instead of dynamic options. As a result, while the website looked decent on PC monitors of certain sizes, it would be hard to view on other common classroom devices such as tablets or phones for instance. Furthermore, the original website did not process user data obtained from the maze and gave teachers a very simple way of creating mazes. The method given allowed teachers to write a question, right answer and wrong answer then move forward until the maze is ready, and then a code would be given. If the code was lost, or the teacher wanted to change a question or review the list of questions they made, it would not be possible within the old website.

As shown in figure 4.1, this project aims to completely redesign and add a ton of functionality to the website, allowing teachers an easier and more user-friendly experience to create mazes, adding a new level of Artificial Intelligence in generating additional wrong answers to the maze, and providing users with review exercises, as well as giving users and teachers a way to view class performance in any given maze. Furthermore, while the website was made primarily for desktop and laptop users, this project makes use of dynamic sizes to ensure that each page is still easy to view and access on tablets and phones. A complete comparison between the old version of the website vs the new version can be found in Appendix B, as well as detailed descriptions of each Web Page.

4.4.1 Index Page, Navigation Bar and Footer

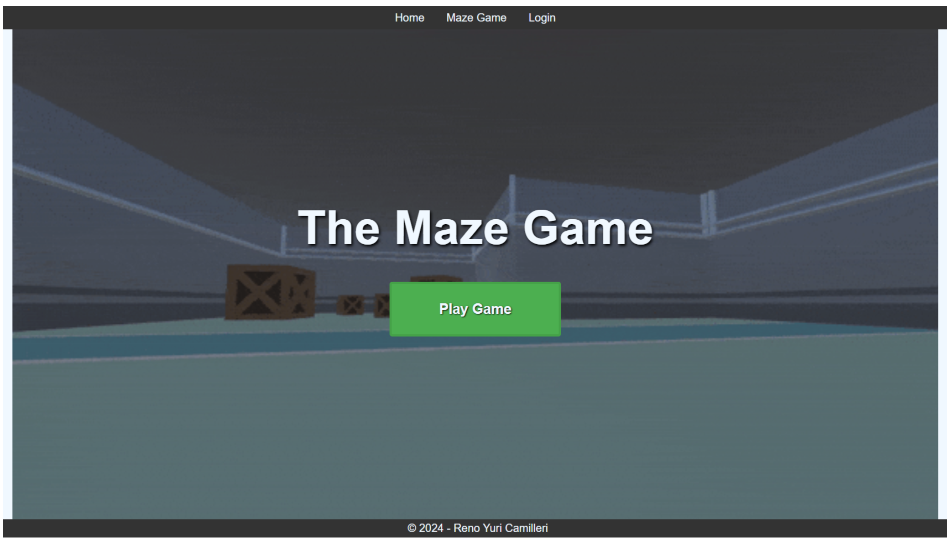


Figure 4.9 Index Page

All the pages within the Improved Maze Game website feature a similar design of splitting the page into 3 main sections, the navigation bar at the top, the main content in the centre, and the footer at the bottom of the page. In terms of the navigation bar, when a user is logged off it features 3 options: "Home", "Maze Game" and "Login". While the Maze Game is shown as an option it is mainly there just to show that it will be an available option after login, since you need to be logged in to play the Maze Game (To save data based on your performance), as such until you log in, the Maze Game option will simply direct you towards the login page.

After logging in the navigation bar will remove the "Login" option, and instead add the following additional options: "My Mazes", "History" and "Logout". The first option is intended for teachers and allows them to access all the mazes they have created, more info later. The second option allows students to view the history of their completed mazes and all their attempts for a particular maze, as well as the page where a user can view review exercises. Logout is used to log out of your account. In addition to this: the "Home" option is changed to direct a user to the instructions page instead of the Index Page, and the "Maze Game" option now directs the user to the game verification page, where a student would need to insert a maze code to play the maze.

The Index Page in itself has a very simple design, the background is an animated Graphics Interchange Format (GIF), showcasing how the Improved Maze Game looks like, as well as features one button "Play Game". This design was chosen to be similar to the design of the original Maze Game. As previously mentioned, the user needs to be logged in to play the Maze Game, and since this page is only accessible if a user is not logged in, the user will simply be directed to the login page.

4.4.2 Instructions Page

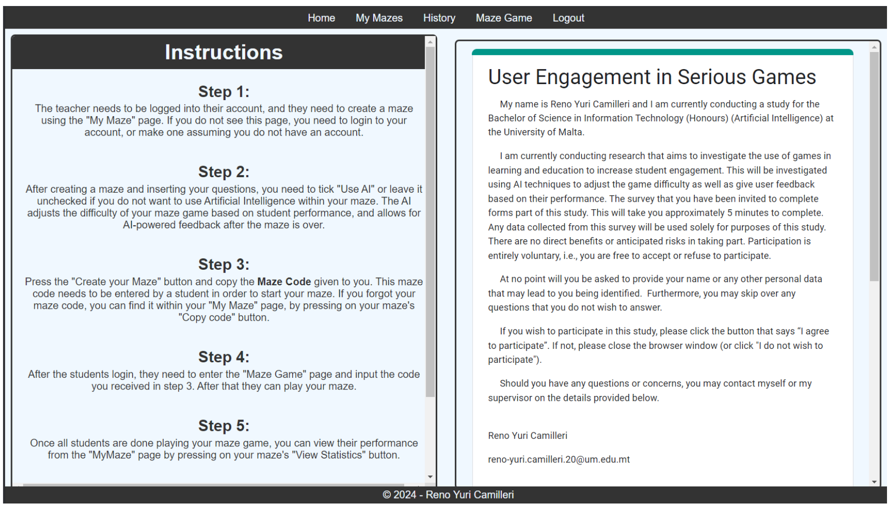
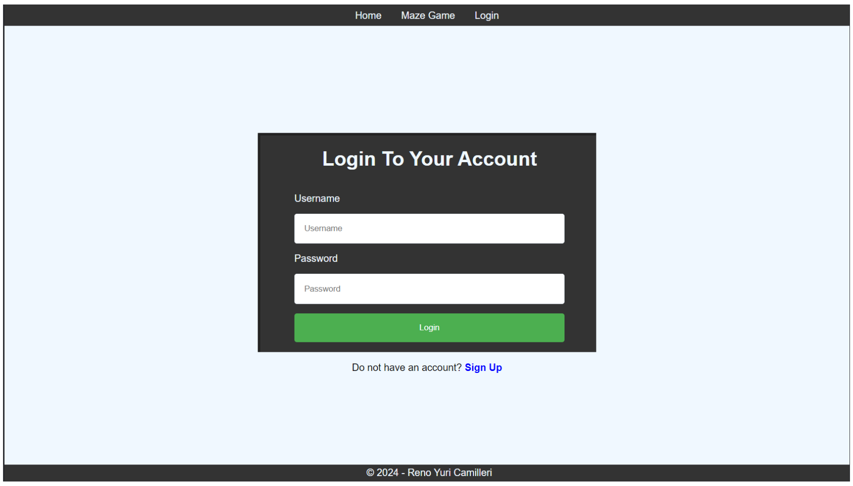


Figure 4.10 Instructions Page

The instructions page as can be seen in the figure above, is made up of two scroll-able frames splitting the main section into two, left and right. On the left-hand side, there are some basic instructions for users and teachers to follow on how to use this website. On the right-hand side, there is an iframe embedding a Google form holding the "User Engagement in Serious Games" form which will be used in the evaluation section of our project.

4.4.3 Login and Sign up Page



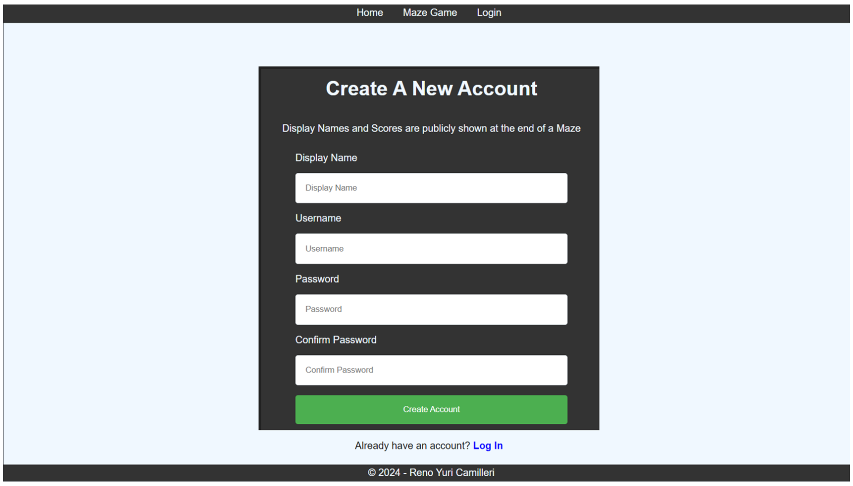


Figure 4.11 Login Page (Above) and Sign Up Page (Below)

Both the Login Page and the Sign-Up page feature a form aligned to the centre of the screen. A username and password are required for a user to access their account, whereas the Display Name is used to identify a user within their classroom, for example, let's say there are three users that share the same name, they could put their name as their display name, but then have separate usernames. Within the sign-up page, there is also a

"Confirm Password" field for a user to write down their password a second time if they misspell their password on the first attempt.

As seen in the figure above the login page has a link to the sign-up page below the form, whereas the sign-up page has a link to the login page below the form. It is common practice to either make the sign-up page its own option in the navigation bar or else place it as an option within the login page (The latter being what this project did), as most users associate the two pages to be connected. Both forms have a button to submit at the end.

When designing the security of accounts, apart from using PHP's password management encryption, both JavaScript and PHP side checks were implemented to avoid users manipulating the client-side script. When making a new account the process defined in Appendix B is followed to verify that users have a decent level of security within their account:

Once the server-side PHP script is run, the password is then hashed, and the database is checked for if the username sent already exists, if the username exists, then the process is aborted, otherwise, the new account is made using the details provided.

4.4.4 Maze Creator 'My Maze' Page

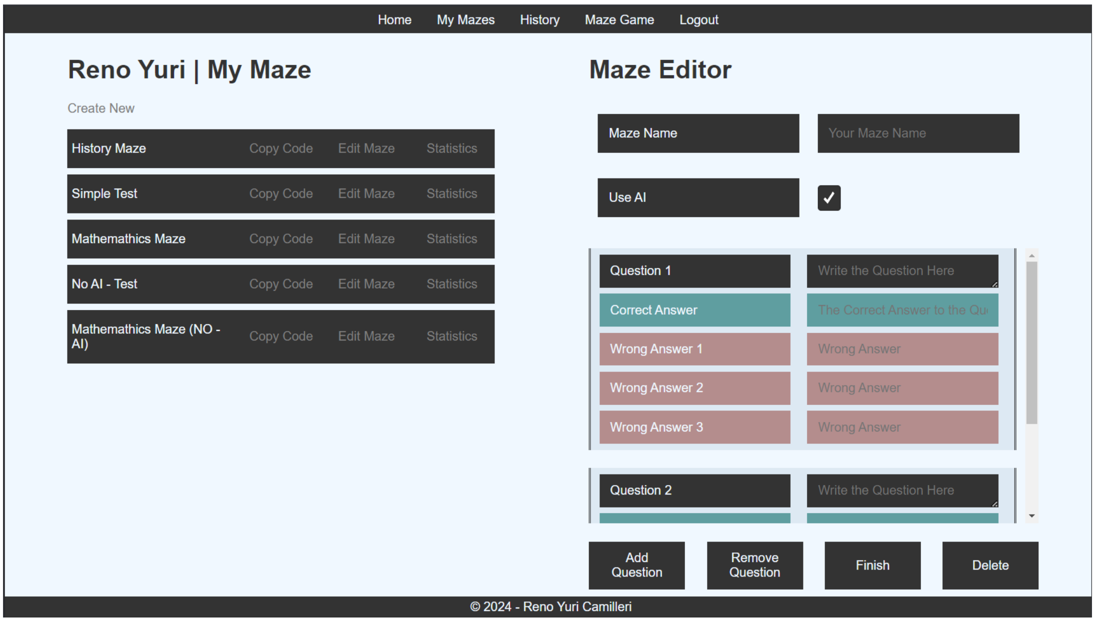


Figure 4.12 My Maze Page

The My Maze page is split into two main aspects, which are the Maze list and the Maze Editor. When the page is opened only the Maze List can be viewed expanded and in the centre, so that teachers can view their mazes, and copy the maze code or view statistics for a particular maze without needing to see the editor for no reason. Should the editor be required either by pressing "Create New" or "Edit Maze" on one of the mazes, the Maze Editor would open up an empty maze format, or open the respective maze.

When opening an existing maze, in addition to filling in all the values, the new AI-generated wrong answer is shown as an additional field. If "Use AI" is disabled then the new AI wrong answer is simply shown as "AI Disabled".

Once a new Maze is made or updated, a PHP script is run that either updates the list of questions or creates new ones. Depending on whether AI is enabled or not, the PHP script would also make a call to the OpenAI API to make use of GPT 3.5-Turbo to generate a new wrong answer, and that answer is saved in the database in addition to the other answers.

4.4.5 Maze Performance Page

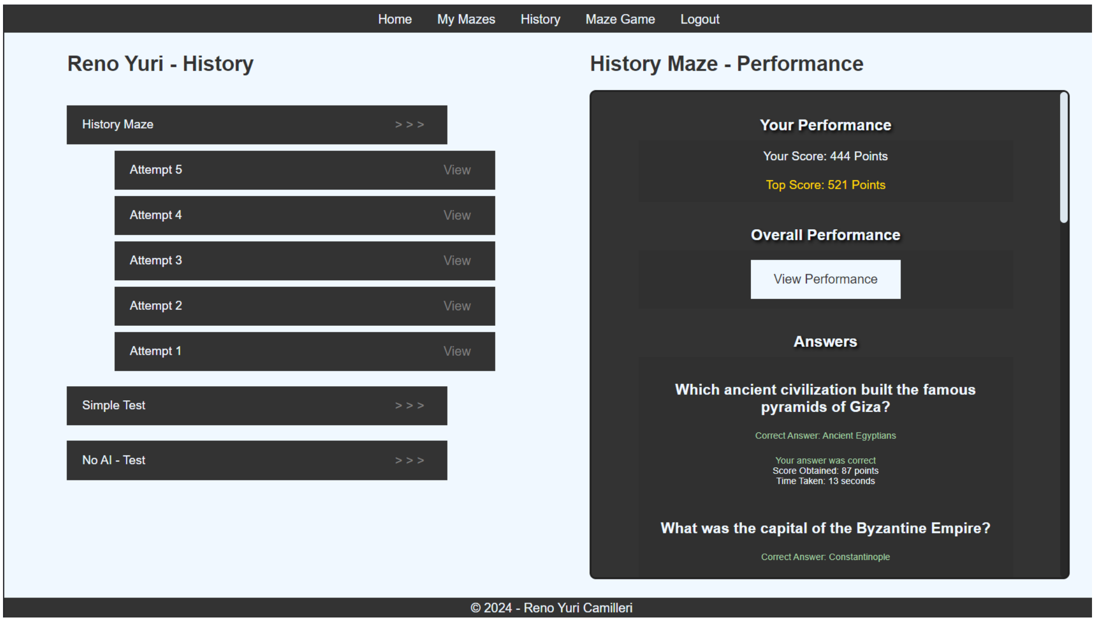


Figure 4.13 Maze History and Performance Page

The Maze Performance Page features the history of mazes that the user has done, in order to avoid cluttering the screen, the user's attempts are hidden and can only be viewed once a particular maze is selected, after the attempts are displayed as in the figure above, a user can view their performance in a maze's particular attempt. From the performance page, the user can view all the correct answers, the time it took them to answer the question, the points they made, and the number of mistakes they made assuming they made a mistake. At the top of the page under "Overall Performance" the user can go to the Maze Statistics Page to view the class performance for a particular maze. Additionally, if AI is enabled, at the bottom of the page the user can perform AI-powered review exercises (More info in the section "Building an AI Feedback Layer").

4.4.6 Maze Statistics Page

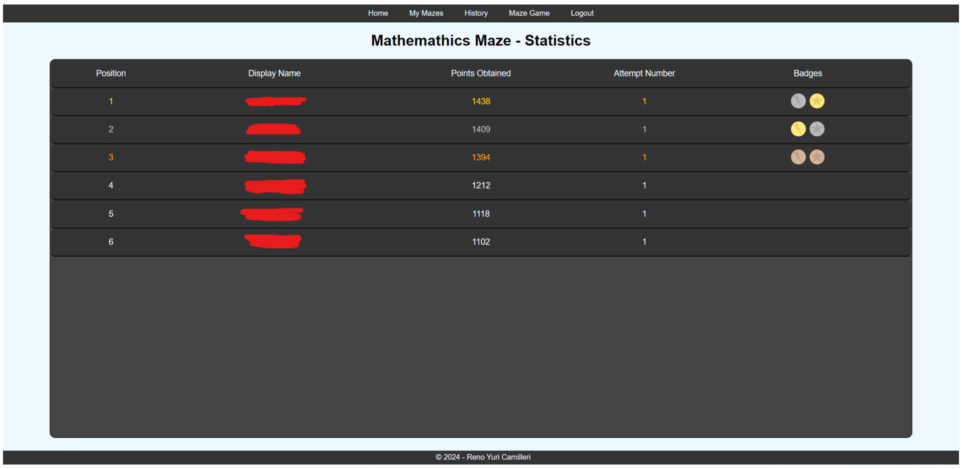


Figure 4.14 Maze Statistics Page (Display Names Censored)

The Maze Statistics page features all the users that finished a particular maze, all placed in order of ranking from first to last place. The top 3 users have their colours set to Gold, Silver and Bronze respectively. In addition to this, the scores obtained by each user are shown as well as what attempts they currently have undergone. Badges achieved by users are also displayed here, the current badges available in the game are for the top 3 fastest players, and the top 3 players with the least mistakes. If two users are tied for a particular badge, they both obtain the same badge.

To stop users from accessing random Mazes they should not have access to, each maze employs a unique code, a teacher should send this code to their students, and the students would insert the code inside this page. The code is then checked if it is valid or not, if invalid an error would return, if valid then the Maze Game with that particular code is loaded.

4.5 Integrating Reinforcement Learning

With the website and game now complete, in addition to the "GPT-3.5" algorithm mentioned in the "My Maze" section of the website, which offers teachers an easier experience to create more options in the serious game environment, another algorithm that is user-focused would be the RL algorithm.

The RL algorithm is implemented as the AI algorithm mentioned within Chapter 4.2 and manages the Maze Generation, starting/ending the game, loading questions obtained from the website database, saving wrong answers in a list, converting lists to a JSON string, and exporting player performance data back to the database.

The RL algorithm chosen for this project was a Q-Learning algorithm with an Epsilon Greedy strategy, this algorithm was chosen as opposed to IRT and BKT for various reasons such as: the recommendations made to use RL algorithms within Background Research, as well as due to it being very simple to implement, yet being highly efficient performance wise. In addition to this RL is a much more commonly used algorithm in video games, and has increased in popularity tremendously over recent years.

The reason an AI approach was chosen as opposed to hard coding scores to decide what room to spawn was to future-proof the Improved Maze Game. If the values were to be hard-coded then it would be harder to edit the game if a developer wished to change the layout of various corridors and question rooms, then the developers would need to run the game multiple times and estimate how long it takes to beat each room, and replaced the hard-coded value manually, in addition to this, if new question room types were to be introduced, then apart from adding new if statements, each value in prior question rooms would also need to be adjusted. On the other hand, by using AI, a developer can temporarily increase the learning rate of the RL algorithm and the algorithm would take care of all these issues on its own while the developers are simply play-testing the game. In addition to this, as previously mentioned, the Q-learning algorithm is very simple, so in the long run, it would still be more code efficient than a multitude of if statements.

Within this project, the Q-Learning algorithm only uses 5 different states ("VeryLow", "Low", "Average", "High" and "VeryHigh") which represent the performance of a student, with 3 different actions (3, 4, 5) which represent the number of choices within a question room. The result of every state and action is the predicted score a user would gain from that instance. Using this predicted score, the AI estimates the median score a user would get by completing the maze and pushes each user to obtain that score.

In terms of the QTable used by the algorithm, it is saved within the database as its own table. In addition to the questions of a maze being sent to the game, the QTable is also sent as a JSON string to the Maze Game and is loaded as a QTable dictionary within the game. As the user plays the game the scores obtained are sent to the algorithm, which updates the QTable via a small learning rate, and at the end of the game the new QTable is sent back to the database, using a very small learning rate set by the developer, the database's QTable is slightly pushed towards the new QTable. This is done as an additional method of updating the performance of the QTable.

Please Refer to Appendix C for a detailed description of how the RL Algorithm works.

4.6 Building an AI Feedback Layer

The AI feedback layer works using GPT 3.5 turbo, while its performance can be enhanced using the GPT 4 model, due to GPT 4 costing over 20 times more than the 3.5 turbo model (As of the writing of this thesis), it is not worth it from a financial standpoint to go for the GPT 4 model when the performance of the two is satisfactory with both versions of the model. The Feedback layer works as follows, all the questions of a particular maze are loaded, a question is chosen at random, and then the GPT API is called with the following prompt:

" *"Given the following question: "**(QUESTION)**", use that question as a parameterised question, and generate a new question of about "**(DIFFICULTY)**", with a correct answer, 3 wrong answers, a hint and a short explanation as to why an answer is correct. Try and make a new question of the same genre as the parameterised question. Return your response as a JSON output with the following structure: "{"review": {"question": "YOUR_QUESTION", "correct": "YOUR_ANSWER", "incorrect_1": "YOUR_WRONGANS", "incorrect_2": "YOUR_WRONGANS", "incorrect_3": "YOUR_WRONGANS", "hint": "INSERT SHORT HINT", "why_correct": "INSERT SHORT EXPLANATION AS TO WHY CORRECT"} }"* ***"***

The question parameter in bold is replaced by the question that is chosen, whilst the difficulty is set to one of the following strings based on the current performance of the user: "an easier difficulty", "a slightly easier difficulty", "the same level in terms of difficulty" or "a slightly harder difficulty". While these may not necessarily be the best prompt for engineering the feedback layer, from a lot of testing, these prompts generally produce the best outcome. The resulting JSON response of the GPT API has always been in the right format, whereas answers are correct most of the time (there have been a few cases where the response returned was not correct, please refer to Appendix D for more details). A JSON response by the API was also chosen to avoid making too many calls to the API, as more calls result in a higher expense for tokens, and take more time to process.

In terms of calling the API, the following changes were made: the "temperature" variable was set to 0.7, "top_p" was set to 0.7 and the "seed" is constantly randomly generated. The reason behind this choice was to increase the randomness of responses obtained, as the API would constantly generate the same parameter questions for the same question. If the temperature or top_p values are set too high, there is an increase in the chance of obtaining wrong answers, whilst having a small value results in repetitiveness. As such 0.7 was found after some testing to be a good value to balance out both aspects.

The feedback layer also makes use of scaffolding where if the user gets more questions correct, the difficulty increases, whereas on the other hand if the user gets more questions incorrect the difficulty decreases. The number of questions given to the user depends on the number of questions within the maze, but is never higher than 20, if a user desires they may attempt review exercises multiple times. After each answer is chosen, the user receives an explanation as to why a particular answer is correct, and throughout the maze, the user may choose to click on a "Hint" button, to view a hint.

For a detailed overview of how the Feedback layer works refer to Appendix C, whereas some responses returned by the AI may be viewed in Appendix D.

4.7 Summary

This chapter covers the original design of the Maze Game, the features that the Maze Game lacked and the transformation of the Maze Game into the Improved Maze Game, highlighting the design of the new system, then going in-depth on the re-design of the new serious game, the creation of the new website along with its new unique systems such as a more complex maze creator and performance review page, and implementing the game within the new website. The chapter then highlights the creation of the various AI algorithms used within the project to provide students with a difficulty-adaptive game, as well as post-game review exercises with scaffolding difficulty.

5 Evaluation

5.1 Literature Review of Game Evaluation Practices

T. Karsenti and S. Parent in their work [13] showed a variety of questions that they asked students to evaluate the effectiveness of their game both in terms of the fun aspect as well as the educational aspect which are the two primary aspects that need to be measured in serious games [18, 19]. A video game survey by Park University [43] was found whose questions were used to analyse the use of video games in the classroom environment. C. Ress has also provided a default survey [44] with basic questions that can be used to evaluate a game. Furthermore two different surveys by UPSKILLS [45, 46] (an Erasmus+ strategic partnership for higher education) were found that specifically focused on studying information related to video games.

Due to the academic value and effectiveness of the sources mentioned above, for this project questions derived from those studies were adopted to provide the survey questions, care was also given to ensure that all questions followed our Ethics plan to ensure all students were notified on what their data would be used for, and care was given to ensure anonymity.

5.2 Student Feedback Evaluation

5.2.1 The Participants

Moving onto the final objective, 13 students were recruited voluntarily to test out the Improved Maze Game, consent forms were given to the participants then 6 students were asked to play the AI version, whilst 7 students were asked to play the standard version of the Maze Game. Both groups were given similar instructions and played a maze with the same mathematical questions, which were derived from Ordinary Level Mathematics mental paper exam questions from the years 2023 - 2020. All students were of roughly the same age and are currently attending University level courses.

It is important to note that the scope of this thesis is not entirely in its evaluation, 13 participants were chosen using convenient sampling for a pilot evaluation. However, looking into future work which will be discussed later on, a large-scale evaluation, including both teachers and students would be ideal.

5.2.2 Instructions Given

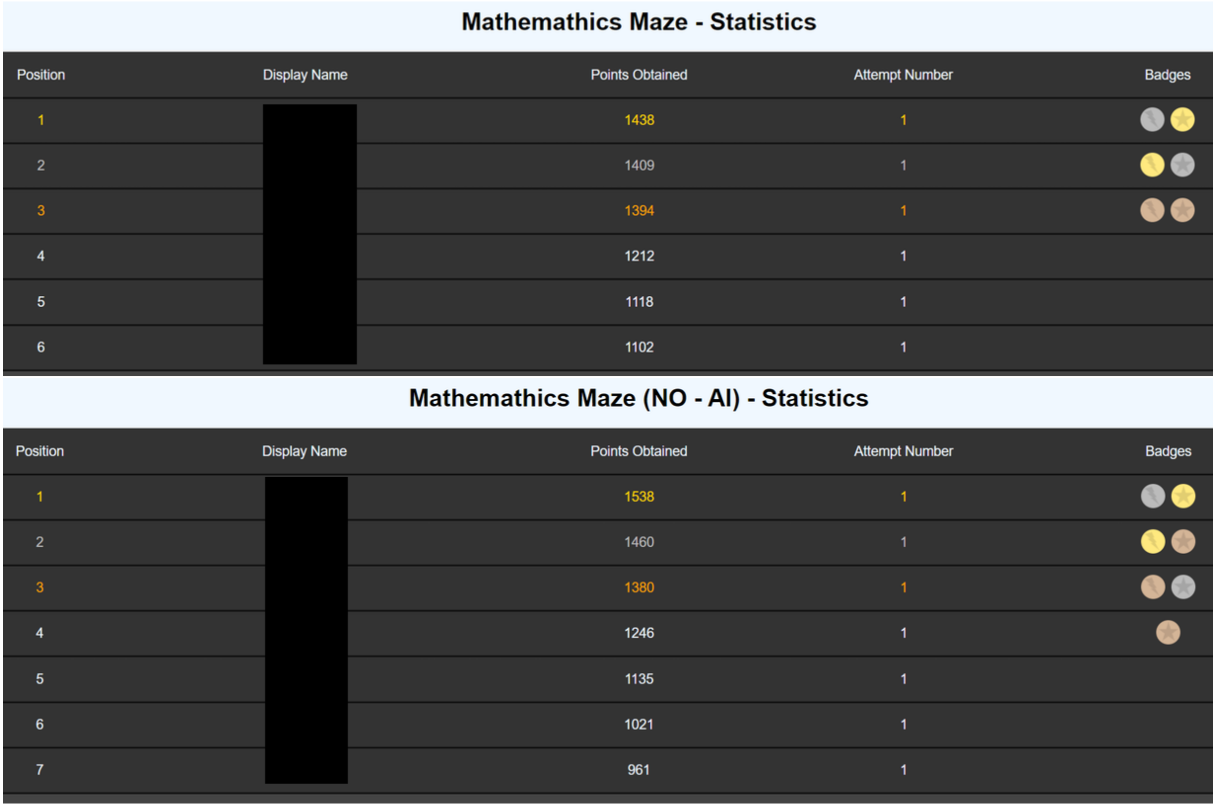
The list of questions and instructions given to the students can be found in Appendix D, as well as the instructions given to each student. All participants were asked to create an account, play the same Maze Game (one version using an AI algorithm, the other without), review their performance and create their own Maze Game to see the game from the point of view of a teacher. In addition to playing the game, the students testing out the AI version of the game were asked to perform a round of review exercises. Both groups of students were then asked to fill in a Survey to give feedback on what they experienced.

5.3 Reinforcement Learning Algorithm Evaluation

RL Algorithms are evaluated by their ability to either find a good policy or a good reward function [47, 48], this project aims for the latter. To find the optimal reward there are two things to consider: what brings the agent closer / further to its objective? For example, Deepmind's AlphaStar utilised RL to master a game called Starcraft [49, 50], the project gave rewards for good decisions within the game, and removed points over time for being idle or making bad decisions [49, 50]. Similarly, this project aims to push users towards an average performance, thus resulting in students underperforming getting easier questions, whilst students with higher performance getting harder questions. Similarly to the example since other quiz-like games such as Kahoot and Quizizz, have timed environments, points are reduced over time for taking too long to make a decision. The RL model then takes the points obtained from the user's performance to evaluate how good a particular decision is.

Since RL requires a large number of episodes to learn its performance, it is a common practice to have the RL model keep learning and improving itself over time [47, 48]. In the case of this project, after a Maze Game is finished, the results of the user's performance are sent back to the server, and the RL model is updated by a tiny learning rate to constantly improve itself and push itself towards the best performance.

5.4 Discussion of Results



Mathemathics Maze - Statistics

Position	Display Name	Points Obtained	Attempt Number	Badges
1	[redacted]	1438	1	
2	[redacted]	1409	1	
3	[redacted]	1394	1	
4	[redacted]	1212	1	
5	[redacted]	1118	1	
6	[redacted]	1102	1	

Mathemathics Maze (NO - AI) - Statistics

Position	Display Name	Points Obtained	Attempt Number	Badges
1	[redacted]	1538	1	
2	[redacted]	1460	1	
3	[redacted]	1380	1	
4	[redacted]	1246	1	
5	[redacted]	1135	1	
6	[redacted]	1021	1	
7	[redacted]	961	1	

Figure 5.1 Screenshot of the performance of Students using AI (Top) vs No AI (Bottom)

From the results obtained from the game, the AI users exhibited scores ranging from 1438, the highest, to 1102, the lowest. This signifies a commendable point difference of 336 between the top and bottom performers. On the other hand, the non-AI users showcased scores ranging from 1538 as the highest to 961 as the lowest, resulting in a more substantial point gap of 577. This observation highlights the efficiency of the RL algorithm, which appears to be functioning as intended. Moreover, it suggests that the game is adeptly adjusting its difficulty based on user performance, thereby supporting a greater challenge for both high and low-performing students.

As seen via the badges shown in both tables, students at the top are generally the fastest and most accurate students. However as seen throughout the methodology, both speed and precision are key factors in obtaining the highest number of points (Since as mentioned in the score manager section, points are lost over time, or by picking a wrong option). Through the use of badges, students can compare why another student may have obtained a higher point amount.

(a) AI Survey

(b) Non-AI Survey

Figure 5.2 User survey - Enjoyment and Replay value

The survey results showed that all participants completed the Maze Game, and the majority of users have played quiz-like games in the past. Results show that 50% of both AI and non-AI users dedicate 16 or more hours per week to playing video games, reflecting a substantial interest in gaming activities among participants. No one found the controls for the game difficult.

AI-game users expressed a greater enjoyment of the game compared to their non-AI counterparts and expressed a higher likelihood of revisiting the game in the future. Moreover, more AI-game users remarked that the game possesses greater learning potential as compared to non-AI users. This might be due to the inclusion of review exercises as a means of reinforcing previously acquired knowledge and allowing users to be able to revise content at their own pace.

All participants agreed that games may act as a great medium for learning, with the majority considering the AI-powered game to be particularly good in this regard. Additionally, most users expressed a preference for a gamified or game-based learning environment over traditional classroom environments as a medium for learning. Lastly, the AI-game participants who had access to the review exercises, provided positive feedback regarding the review exercises, highlighting their efficiency in achieving learning outcomes. A list of all the results obtained can be found in Appendix E.

6 Conclusion

This project managed to create a gamified learning platform through which the evaluation targeted achieving the aim introduced at the start of the project by comparing performance through the use of a game, and user engagement through the use of a survey to find the difference in skills and engagement between students in the same game environment and test the effectiveness of LLMs and RL in improving engagement.

6.1 Revisiting the Aims and Objectives

This project managed to achieve all five objectives previously provided:

1. Several AI models and techniques were reviewed and presented in Chapters 2 and 3 of this Final Year Project.
2. The Original Maze Game was imported in a Unity package, and whilst it served as a foundation of the new Maze Game environment, a number of modifications were carried out in terms of its user interface, front-end content creation for teachers and procedural content generation of the maze look and feel.
3. A new question creator using the GPT API in addition to a new Q-Learning AI model was implemented to improve user engagement.
4. A separate AI feedback layer was created and integrated using a LLM and was integrated into the existing game architecture to offer further personalised revision questions to aid the learner.
5. A comparative analysis was done for this pilot game where participants evaluated the AI and non-AI versions of the game to find the difference between usability, engagement and performance.

6.2 Challenges and Limitations

While RL Algorithms can become very accurate, a large number of runs of a program are required before the algorithm becomes fairly accurate, as such one challenge of this project is the need to retrain the algorithm each time a major change is made to the game.

A major issue with this project is the use of the GPT 3.5 API to create review questions. The issue with it is that while in most cases the new questions generated are accurate, there have been several cases where the AI would give a wrong answer. For instance, within the test questions, the GPT API returned wrong answers when it came to calculating time-based mathematical questions. Throughout testing, it appears that swapping to the GPT 4.0 API provides better answers, but it comes at a significantly higher cost for tokens generated.

In addition to this, the use of the GPT API in itself is also a limitation for the project, as it is a paid service meaning that if a school were to adopt this project, it would impose a lot of financial cost to have the API constantly being called by students each time they desire to perform review exercises.

6.3 Future Work

Following feedback obtained from the users who played the game, it is evident that enhancing accessibility features seems to be a factor that can be expanded on in the future to benefit the game's appeal and accessibility. One suggestion, in particular, was to incorporate a text-to-speech option to narrate both the question and potential answers within the game, to cater both to users with visual impairments and those that simply prefer audio-based interaction. Moreover, while the game was originally designed for devices that use keyboards, integrating touch-pad controls for tablet and smartphone users would allow more users to be able to access the game, especially considering that tablets are also commonly used devices within schools.

Personalisation can also be added to the game in the future to improve user engagement and immersion. However, implementing such features would result in ethical issues surrounding managing personal user data, which presents a challenge that extends beyond the current scope of the project's objectives. With that in mind, it would still be a good research point for future developments in user engagement.

In addition to this, while the sample size of 13 students was sufficient for the purposes of this project, it is worth noting that the 13 students may not indicate the opinions of the broader student population. Expanding this study to feature a larger and more diverse group of students would provide a better understanding of the effectiveness of this study. Additionally, while the current study focused on a mathematics-themed maze, exploring alternative subjects such as science subjects or languages could potentially result in different results.

In conclusion, the exploration of adaptability in content generation for serious games / gamified learning environments is an area of study with limited research, yet promising outcomes. As seen within the context of this project, the integration of gamification with the use of adaptive AI has the potential to improve the educational sector. However, to harness the full potential of gamification, it is important for research efforts to delve deeper into understanding what drives user engagement.

References

[1] K. Squire, *Video Games and Learning: Teaching and Participatory Culture in the Digital Age. Technology, Education–Connections (the TEC Series).* ERIC, 2011.

[2] D. W. Shaffer, K. R. Squire, R. Halverson, and J. P. Gee, "Video games and the future of learning," *Phi delta kappan*, vol. 87, no. 2, pp. 105–111, 2005.

[3] C. Steinkuehler and K. Squire, "Gaming in educational contexts," in *Handbook of educational psychology*, Routledge, 2024, pp. 674–695.

[4] B. Snow, *Why most people don't finish video games*, Aug. 2011. [Online]. Available: `http://edition.cnn.com/2011/TECH/gaming.gadgets/08/17/finishing.videogames.snow/index.html#:~:text=Only%2010%25%20of%20avid%20gamers`.

[5] E. Bailey and K. Miyata, "Improving video game project scope decisions with data: An analysis of achievements and game completion rates," *Entertainment Computing*, vol. 31, p. 100 299, 2019.

[6] L. Chen, P. Chen, and Z. Lin, "Artificial intelligence in education: A review," *Ieee Access*, vol. 8, pp. 75 264–75 278, 2020.

[7] H. Y. Zheng and Y. Zhou, "Predictive analytics and its uses in higher education," *Big Data on Campus: Data Analytics and Decision Making in Higher Education*, p. 50, 2020.

[8] F. Almeida and J. Simoes, "The role of serious games, gamification and industry 4.0 tools in the education 4.0 paradigm," *Contemporary Educational Technology*, vol. 10, no. 2, pp. 120–136, Apr. 2019. DOI: `https://doi.org/10.30935/cet.554469`.

[9] J. Huizenga, G. T. Ten Dam, J. Voogt, and W. Admiraal, "Teacher perceptions of the value of game-based learning in secondary education," *Computers & Education*, vol. 110, pp. 105–115, 2017.

[10] W. Westera *et al.*, "Artificial intelligence moving serious gaming: Presenting reusable game ai components," *Education and Information Technologies*, vol. 25, no. 1, pp. 351–380, Jul. 2019. DOI: `https://doi.org/10.1007/s10639-019-09968-2`. [Online]. Available: `https://link.springer.com/article/10.1007/s10639-019-09968-2`.

[11] R. Y. Camilleri, D. Sciberras, E. Gatt, and G. Borg, *Maze game - home page*, May 2023. [Online]. Available: `http://maze.upskillsproject.eu/`.

[12] J. Zeng, S. Parks, and J. Shang, "To learn scientifically, effectively, and enjoyably: A review of educational games," *Human Behavior and Emerging Technologies*, vol. 2, no. 2, pp. 186–195, 2020.

[13] T. Karsenti and S. Parent, "Teaching history with the video game assassin's creed: Effective teaching practices and reported learning," *Review of Science, Mathematics and ICT Education*, vol. 14, no. 1, pp. 27–45, 2020.

[14] C. A. MacLeod, "Undergraduate teaching and assassin's creed: Discussing archaeology with digital games," *Advances in Archaeological Practice*, vol. 9, no. 2, pp. 101–109, 2021.

[15] P. Caserman *et al.*, "Quality criteria for serious games: Serious part, game part, and balance," *JMIR Serious Games*, vol. 8, no. 3, e19037, Jul. 2020. DOI: `https://doi.org/10.2196/19037`.

[16] K. Larson, "Serious games and gamification in the corporate training environment: A literature review," *TechTrends*, vol. 64, no. 2, pp. 319–328, 2020.

[17] K. Madani, T. W. Pierce, and A. Mirchi, "Serious games on environmental management," *Sustainable cities and society*, vol. 29, pp. 1–11, 2017.

[18] G. Hookham and K. Nesbitt, "A systematic review of the definition and measurement of engagement in serious games," in *Proceedings of the Australasian Computer Science Week Multiconference*, ser. ACSW '19, Sydney, NSW, Australia: Association for Computing Machinery, 2019, ISBN: 9781450366038. DOI: `10.1145/3290688.3290747`. [Online]. Available: `https://doi.org/10.1145/3290688.3290747`.

[19] F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger, and R. Berta, *Assessment in and of serious games: An overview*, Feb. 2013. [Online]. Available: `https://www.hindawi.com/journals/ahci/2013/136864/`.

[20] S. Ikeda, T. Takagi, M. Takagi, and Y. Teshigawara, "A study on a method of estimating the difficulty of quizzes focused on quiz types," in *Proceeding from ICCE2011, The 19th International Conference on Computers in Education, Chiang Mai, Thailand*, 2011, pp. 312–316.

[21] Y. B. David, A. Segal, and Y. Gal, "Sequencing educational content in classrooms using bayesian knowledge tracing," in *Proceedings of the sixth international conference on Learning Analytics & Knowledge*, 2016, pp. 354–363.

[22] G. Andrade, G. Ramalho, H. Santana, and V. Corruble, "Challenge-sensitive action selection: An application to game balancing," in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IEEE, 2005, pp. 194–200.

[23] G. Tesauro, "Td-gammon, a self-teaching backgammon program, achieves master-level play," *Neural computation*, vol. 6, no. 2, pp. 215–219, 1994.

[24] A. L. Samuel, "Some studies in machine learning using the game of checkers. ii—recent progress," *IBM Journal of research and development*, vol. 11, no. 6, pp. 601–617, 1967.

[25] M. Abramson and H. Wechsler, "Competitive reinforcement learning for combinatorial problems," in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, IEEE, vol. 4, 2001, pp. 2333–2338.

[26] M. Riedmiller *et al.*, "Karlsruhe brainstormers-a reinforcement learning approach to robotic soccer," in *RoboCup 2000: Robot Soccer World Cup IV 4*, Springer, 2001, pp. 367–372.

[27] S. Liu, Y. Chen, H. Huang, L. Xiao, and X. Hei, "Towards smart educational recommendations with reinforcement learning in classroom," in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2018, pp. 1079–1084. DOI: `10.1109/TALE.2018.8615217`.

[28] S. Xu, M. Sun, W. Fang, K. Chen, H. Luo, and P. X. Zou, "A bayesian-based knowledge tracing model for improving safety training outcomes in construction: An adaptive learning framework," *Developments in the Built Environment*, vol. 13, p. 100 111, 2023.

[29] T. Schodde, K. Bergmann, and S. Kopp, "Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making," in *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, 2017, pp. 128–136.

[30] X. Fan, "The application of reinforcement learning in video games," in *2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Atlantis Press, 2023, pp. 202–211.

[31] I. Szita, "Reinforcement learning in games," in *Reinforcement Learning: State-of-the-art*, Springer, 2012, pp. 539–577.

[32] A. Alam, "Employing adaptive learning and intelligent tutoring robots for virtual classrooms and smart campuses: Reforming education in the age of artificial intelligence," in *Advanced Computing and Intelligent Technologies: Proceedings of ICACIT 2022*, Springer, 2022, pp. 395–406.

[33] F. AlShaikh and N. Hewahi, "Ai and machine learning techniques in the development of intelligent tutoring system: A review," in *2021 International Conference on innovation and Intelligence for informatics, computing, and technologies (3ICT)*, IEEE, 2021, pp. 403–410.

[34] R. Khazanchi and P. Khazanchi, "Artificial intelligence in education: A closer look into intelligent tutoring systems," in *Handbook of research on critical issues in special education for school rehabilitation practices*, IGI Global, 2021, pp. 256–277.

[35] R. Winkler and J. Roos, "Bringing ai into the classroom: Designing smart personal assistants as learning tutors," 2019.

[36] S. Sampayo-Vargas, C. J. Cope, Z. He, and G. J. Byrne, "The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game," *Computers & Education*, vol. 69, pp. 452–462, 2013.

[37] T. Jagušt, I. Botički, and H.-J. So, "Examining competitive, collaborative and adaptive gamification in young learners' math learning," *Computers & education*, vol. 125, pp. 444–457, 2018.

[38] W. S. Sayed *et al.*, "Ai-based adaptive personalized content presentation and exercises navigation for an effective and engaging e-learning platform," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3303–3333, 2023.

[39] A. M. Quadir and M. A. Khder, "Exploring the potential of a.i.-driven opponents in video games," in *2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, 2022, pp. 464–469. DOI: 10.1109/ICETSIS55481.2022.9888909.

[40] P. Brusilovsky and S. Sosnovsky, "Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack," *Journal on Educational Resources in Computing (JERIC)*, vol. 5, no. 3, 6–es, 2005.

[41] B. Little, "Why do we celebrate columbus day and not leif erikson day?" *National Geographic*, 2015.

[42] R. McCall and L. Baillie, "Ethics, privacy and trust in serious games," *Handbook of Digital Games and Entertainment Technologies*, pp. 611–640, 2017.

[43] S. Keim, *Video game survey by park university*. [Online]. Available: https://www.surveymonkey.com/r/parkvideogamesurvey.

[44] C. Ress, *Default survey questions (five-star ratings) | playtestcloud help center*. [Online]. Available: https://help.playtestcloud.com/en/articles/1187190-default-survey-questions-five-star-ratings.

[45] UPSKILLS, *Upskills game survey*. [Online]. Available: https://docs.google.com/forms/d/e/1FAIpQLSehPD_DTCQuyhm8kajcLpcrWRbrSHaTiRGL1x789Wyz9ixKZQ/viewform.

[46] UPSKILLS, *Game play preferences survey | upskills project*, May 2021. [Online]. Available: https://upskillsproject.eu/currently-running-game-play-preferences-survey/.

[47] D. L. Poole and A. K. Mackworth, “13.6 evaluating rl algorithms,” in *Artificial Intelligence 3E Foundations of Computational Agents*. 2023. [Online]. Available: https://artint.info/3e/html/ArtInt3e.Ch13.S6.html#:~:text=We%20can%20judge%20a%20reinforcement.

[48] b. venkatapathy baratchandar, *Evaluation metrics for reinforcement algorithms*, Nov. 2023. [Online]. Available: https://medium.com/@barathchandarcse/evaluation-metrics-for-reinforcement-algorithms-ff2bf5869fe4.

[49] O. Vinyals *et al.*, “Grandmaster level in starcraft ii using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[50] T. A. team, *Alphastar: Mastering the real-time strategy game starcraft ii*, Jan. 2019. [Online]. Available: https://deepmind.google/discover/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii/.

Appendix A Improved Maze Game

A.1 Maze Game Design Images

Figure A.1 Main Menu - Old vs New Design of the Main Frame

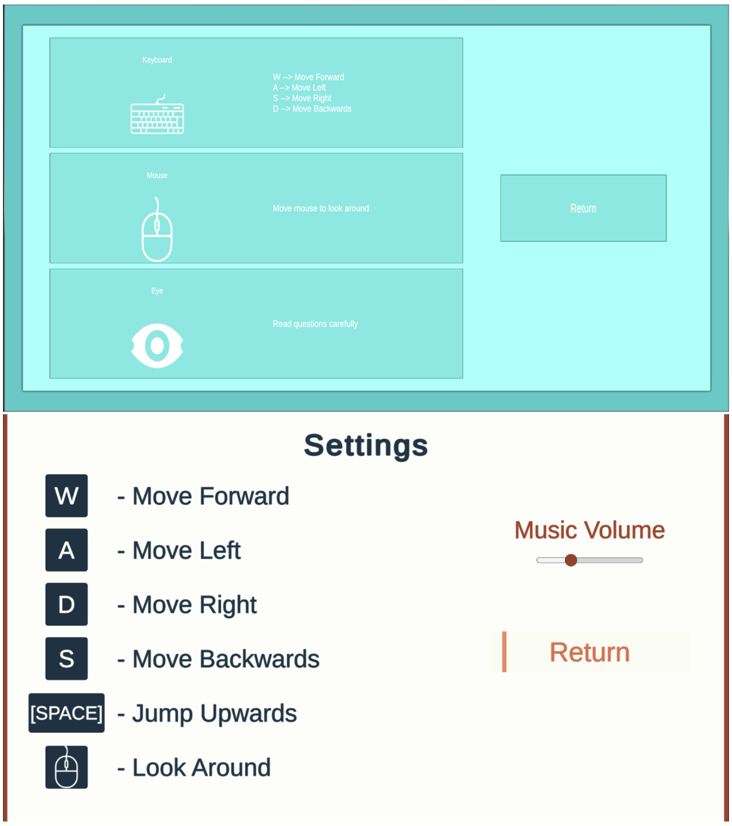


Figure A.2 Main Menu - Old vs New Design of the Settings Frame

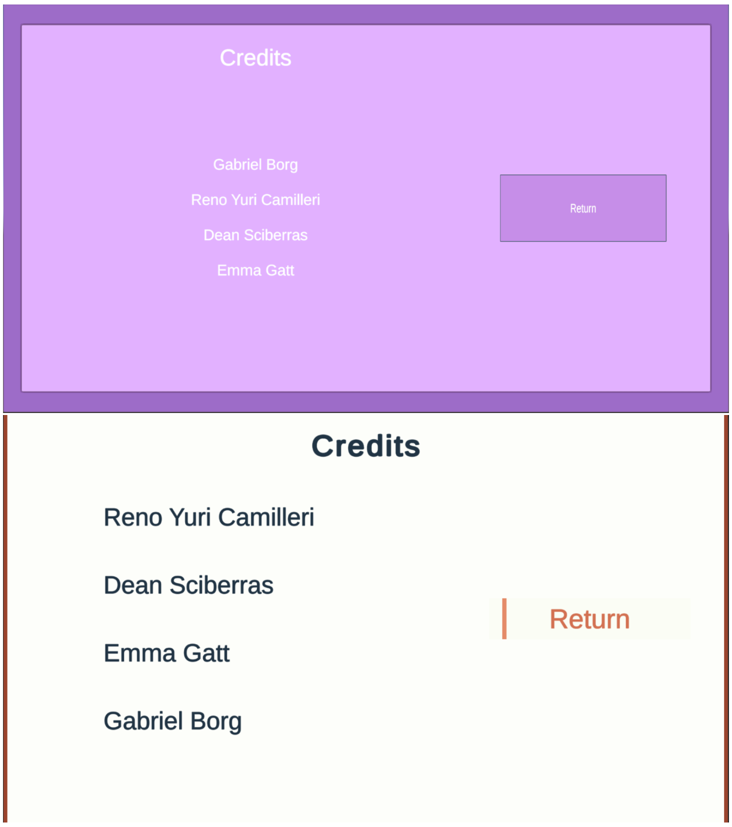


Figure A.3 Main Menu - Old vs New Design of the Credits Frame

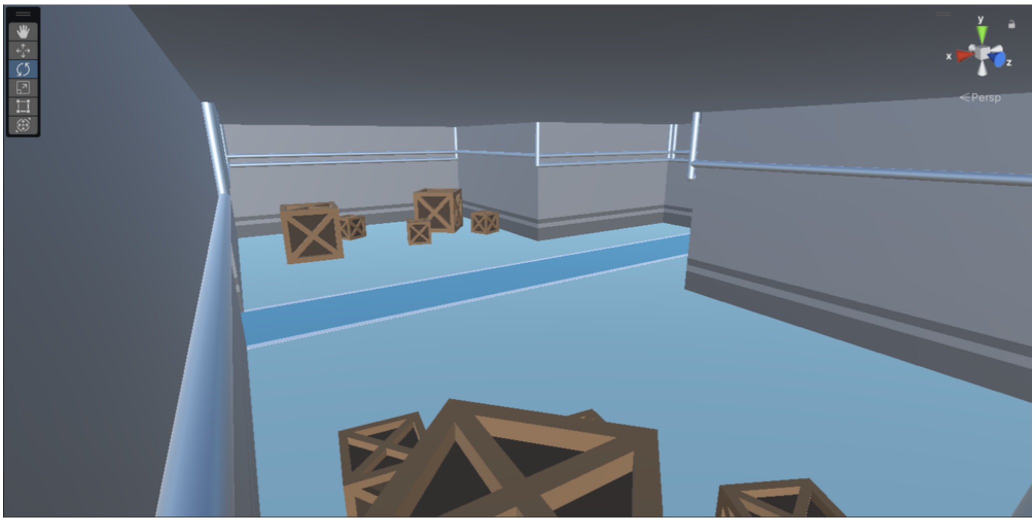


Figure A.4 Start Room Prefab

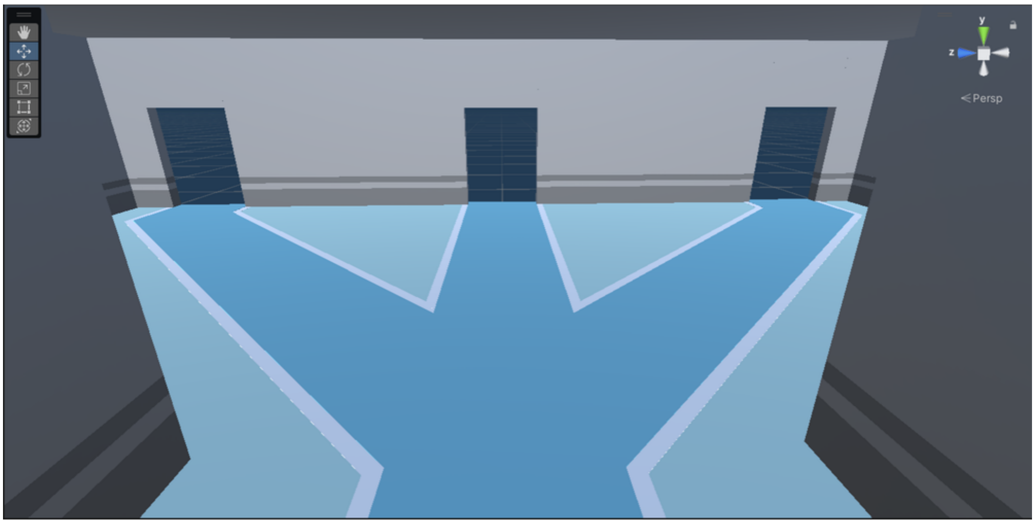


Figure A.5 Choice Rooms with 3 options

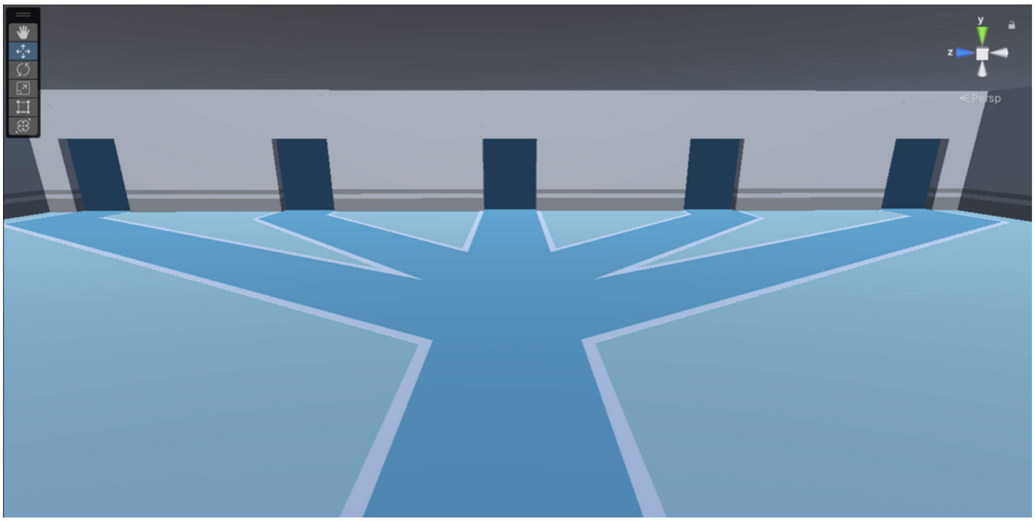


Figure A.6 Choice Rooms with 5 options

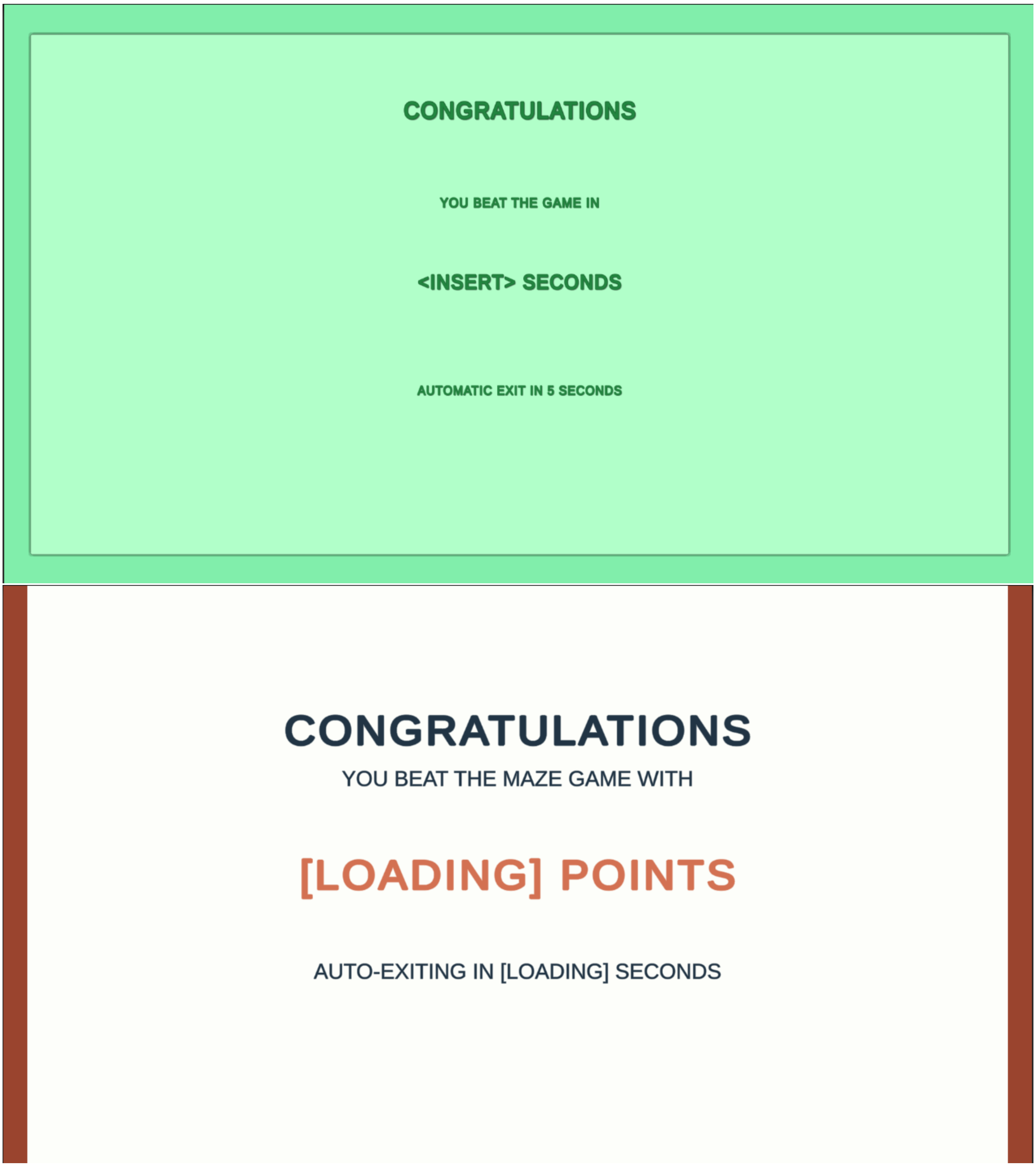


Figure A.7 End Screen - Old vs New Design

A.2 Maze Game Details

Music

The Maze Game features five different music songs, of varying lengths and volumes. In the Maze Game, only two of those five songs are implemented and used. Due to the unnecessary file space taken up by the music, in the Improved Maze Game, only the two used songs are kept, and the others were deleted from the Unity Project. Within the

Improved Maze Game the main menu has an object called ”Music Manager” that holds the menu music playing on a loop. Similarly, the main game scene has the music within the ”Game Manager” object.

Two additional scripts were made for this project in relevance to the music which were the ”VolumeController.cs” and ”InGame_Music.cs” files that can be found in the Start and Game scenes respectfully. A major problem within the Maze Game was that there was no volume slider to control how loud the music was, as such a new volume slider was added to the settings page of the main menu and the job of the volume controller manages updating the menu music volume based on the slider, as well as saving and loading the volume settings using Unity's Player Preferences system, this allows the audio settings to save in between scenes. As previously mentioned, the music songs have varying volumes, as such the in-game music script loads the volume settings from Player Preferences that the user set up in the main menu and multiplies those volume settings by two. (In this case, it was multiplied by two since the in-game music was roughly half as loud as the main menu music.)

Movement and Camera

The movement and camera scripts were re-used within the Improved Maze Game with no changes made. The movement script features a basic first-person movement script that allows movement either using ”WASD” or the arrow keys with a jumping mechanic whenever the player presses the spacebar key. The camera script also features first-person camera movement using the mouse. Just like in the maze game, the movement scripts and the camera script were attached to a capsule game object named ”Player”.

A.2.1 Start Menu

Menu Design

The Improved Maze Game is split across 3 scenes, the start scene, the game scene and the end scene. Within the start scene, there is the main menu. The Improved Maze game contains all the same four options from the Maze Game but with a new more professional-looking design. When making the design care was placed to ensure that all sizes within the interface scaled based on the height of the user screen. Since this project is mainly aimed at PC and laptop users, throughout the design and testing process the UI was always tested with a 16:9 aspect ratio (The most common ratio for desktop monitors).

The main menu is composed of 3 main frames which are: the main frame from which the user can play the game, quit the game or access the other 2 frames. The settings frame which shows all the controls of the game as well as includes the volume slider that was mentioned in the music section earlier on, and the credits frame which contains the list of the developers that worked on the Maze Game. Since the Improved Maze Game is built upon the foundation of the Maze Game, all of the prior developers who worked on the Maze Game are still listed in the credits section.

Menu Code

The Main Menu makes use of two different scripts which are ”MainMenu.cs” and ”ButtonEffect.cs”. The first script holds a reference to three separate frames, the main frame, the settings UI and the credits UI. The script is in charge of activating/deactivating these frames based on what button is pressed to take the user to the frame they wish to see. The second script is a custom effect script for buttons that makes the vertical line to the left of the button text disappear when a button is hovered upon. To show an additional sense of responsiveness, a ”Click” sound effect was added to the Improved Maze Game which plays a simple click sound effect whenever a button is clicked.

When ”Play” is pressed the main menu script uses Unity Engine’s Scene Manager to move on to the next scene which would be the game scene. Whereas when ”Quit” is pressed Unity Engine’s application external call is used to run the JavaScript function ”javaScriptQuitGame()” from within the web page where the game is hosted. Since this game will be played on a website, it would be best to have the website deal with the process of quitting the maze game rather than the Unity Player. It is worth noting that while this system still works as of the writing of this project, the application external call may become a deprecated unity function in the future.

A.2.2 Prefabs

Start and End Room

When moving to the game scene, the first room the user will encounter would be the Start room. The gameplay starts from this room onwards as such it is important to ensure that the user faces an aesthetically nice and consistent room design within the entirety of the maze. Unlike the other rooms within the maze, where the user is trying to solve questions as quickly as possible, in the start room the timer has not yet started, meaning the user has a chance to look around and properly digest the visuals of the Start room, as such it is important to give users a good impression of what the game is about. In this project, we are working with a maze, but there are a variety of different types of mazes. For this project, the Start room as well as all the other rooms within the maze were made to have a modern ”Escape Room” style. This style was chosen to replicate the same style used within the Maze Game.

Once the Start Room is designed, it is important to place it within the scene. The procedural generation script which will be discussed in the next section will use the Start room as the primary point from where all room and pathway generations will take place. Once the Start Room is placed, place the player prefab inside the Start Room. The End Room does not need to be placed within the scene as it will be auto-spawned and placed by the procedural generator. Within the End Room there should be an invisible wall placed that once triggered would proceed to end the game (More details in the section: "Ending the Game").

Both the Start Room and the End Room make use of a "Choice_Prefab.cs" script which will be discussed in detail in the next subsection. However, in the case of the Start Room, the "start" game object can be set to anything as it will not be used, and the "choices" game object variable can be set to "1" to only hold the exit of the Start Room. For the End Room, the "choices" game object variable can also be set to "1" and can mark any part since the script would not continue to generate rooms after that point.

Question Rooms

The Question Rooms within the maze act as large rooms with a single entrance and multiple exits. The number of exits determines the number of possible answers a user has for a particular question given to them. For this project three different types of Question Rooms were implemented, one with 3 exit choices, one with 4 exit choices and one with 5 exit choices. The AI will be making use of these 3 different choice rooms to adjust difficulty once implemented. In terms of design, a similar design was used to that of the Maze Game, where each room is large and open and has a carpet leading to all the possible answers to make it simple for a user to notice the number of pathways. An invisible wall that can be passed through is placed in each Question Room close to the entrance, the wall with the use of the "Questionroom_Tagger.cs" is in charge of detecting when a player collides with it, and in such an event is used by other scripts to perform various functions.

In terms of script design, all question rooms make use of a "Choice_Prefab.cs" script. This script features references to the "start" and "choices[]" game object transformations, in addition to holding a reference to the user tagger game object and a question string (The question string is managed by the Game Manager, whilst the rest need to be set by the developer). The start represents the entrance to the question room and is used by the procedural generator to align the start positions of each room with the exit positions of other rooms. To avoid visual bugs, it is important that all start and exit game objects (In this case the choices[] list acts as exit points) are of equal size. When it comes to the choices parts, it is important to distance them by a reasonable amount to allow space for pathways to be spawned.

When the user tagger is triggered, the tagger game object is immediately deleted to avoid it from being triggered a second time, and then the player's movement speed gets set to zero to freeze them in place. The Questions themselves are placed inside the UI for better readability, this UI is managed by a "Question_UI.cs" script which simply changes the text on the UI when activated, it also displays every second the amount of time the user has before they get unfrozen. Freezing the user acts as a way to give users reading time, so both fast readers and slow readers have time to read the question before deciding on an answer. Once the freeze time is over, the user's speed is restored. This function also is responsible for assisting the Procedural Generation and the AI systems, more details will be discussed in their relevant sections.

Pathways

Pathways act as corridors to move from one question room to another. They also serve the purpose of making the game feel like a proper maze. When it comes to the design of Pathways, this project splits them into three categories, pathways facing the left, pathways facing the middle, and pathways facing the right. All three different pathway types are used to generate the maze without collisions. Just like Question Rooms, pathways feature a start and end point, as well as an invisible user tagger. The tagger is very similar to that of the Question Rooms, in this case, the script is called "Player_Tagger.cs". In addition to this, the Pathway prefabs also have an invisible part that can be passed through at the very entrance that features a text UI that showcases the answer placed on the pathway. Since the pathway starts will be aligned to the exits of the Question Rooms, the text will appear right in front of every exit.

In terms of design it is important that from the entrance of a pathway, the user should not be able to see the exit as that would be a giveaway if a pathway is correct or wrong. In addition to this, the user tagger should be placed in a position where the user would only touch it as soon as they are about to see the exit of the pathway. Thus tagging a user just at the moment when they realise if a pathway is correct or not. Lastly, the pathway prefab should have a bounding box collider that a user can pass through, that covers the entire prefab. This bounding box will be used in the procedural generation to avoid different pathways colliding with one another.

The Pathways use a "Pathway_Prefab.cs" script that features the start, end and user tagger parts. In addition, it also features a reference to the text UI of the corridor, as well as two boolean values "userTaggerEnabled" and "correctAnswer". The prior is set to true by default and is only disabled in the very first corridor of the Improved Maze Game, whilst the "correctAnswer" boolean is set to true when the corridor has the correct answer, and false when it has the incorrect answer (This is managed by the procedural generation). The script also automatically references its own bounding box.

When a user is tagged, the tagging part is deleted, and then depending on if the pathway had the correct answer or not, the pathway prefab will either request the procedural generation to generate the next question room and set of pathways, or it would alternatively send a request to the AI to save the wrong answer, as well as send a request to the score manager to deduct points.

End Walls

The End walls were designed to look like closed doors, the main objective of End walls is to be placed at entrances of question rooms once a user has triggered the question room and become frozen. This way the user would not be able to return to a previous question room. The main purpose behind this is to save memory space. Since previous question rooms will become inaccessible, they would be able to be removed to save space. Additionally, End walls are placed at the end of every pathway prefab that does not lead to a question room.

A.2.3 Procedural Generation

The Procedural Generation within this game is controlled by a script titled ”Maze_Generator.cs”, within the game this script was attached to an empty game object called ”GameManager”. This script holds multiple public object references which need to be set by the developers such as a reference to the Start Room game object within the scene, a reference to the Final Room prefab, a reference to the End Wall, a reference to the Time Controller script (More info later on), as well as list references to the number of Question Rooms, Left Pathways and Right Pathways available as well as references to their actual prefabs. In addition, there is also a reference to the Centre Pathway.

For the first maze section generation, a ”Begin()” function was implemented, it is important not to use Unity’s built-in ”Start()” function, as the procedural generation should not run immediately, the goal is to make the procedural generation a stand-alone script that is managed by an exterior script, in this project’s case it will be managed with a RL algorithm later on. As such the maze generation should only begin once the algorithm loads all the necessary data it needs to load, and is ready to launch the maze (Data such as loading the questions and answers for instance). The Begin() function takes in as parameters an integer seed, as well as the question, correct answer, and incorrect answers.

The Begin() function starts by setting the Random seed to match the seed sent to the function. This was a design choice within the Maze Game that ensured that all students playing the same maze game would have the same seed and in turn would get the same rooms generated, creating a fair playing field for all students. While the AI algorithm will make each maze different for each student, this project aims to still give teachers the option to use the Improved Maze Game without enabling AI algorithms. Once the seed is set up, a Centre pathway prefab is cloned, and this clone is aligned in front of the exit of the starting room, with the user tagger disabled. To align prefabs within Unity we have to transform the entire model to be equal to the position of the exit part of the previous model, whilst also subtracting the local position of the starting part within the current model. Using the pathway as a new reference point, loading continues operating in a similar way till the end of the game.

Loading New Rooms

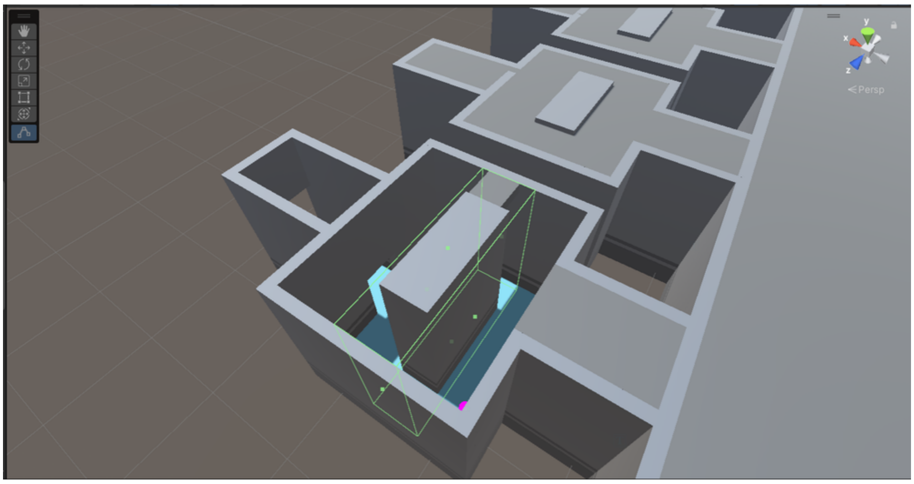


Figure A.8 Player Tagger to Generate the Next Question Room

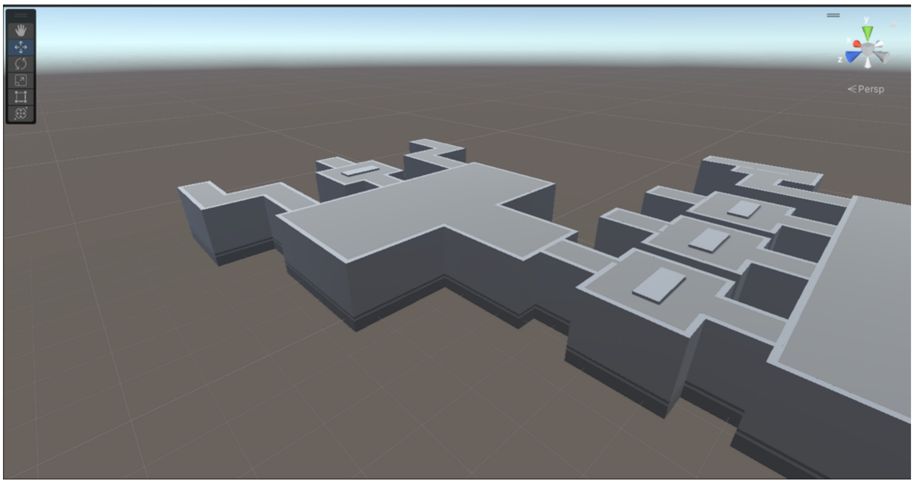


Figure A.9 Next Question Room is Generated with its Pathways

New Question Rooms and pathways are loaded together using the previous correct pathway's end point as a point of reference for positioning the next rooms. Once a collision occurs between the user and the user tagger in a pathway, the Score Manager is alerted to run a function called ”RequestNewMaze()” or ”ReceiveWrongAnswer()” depending on if the prefab has the correct answer or not, the Score Manager would then be responsible for dealing with the user score (More info later) and running a function within the AI algorithm to request the Maze Generator to Generate the next maze.

The question room generation works as follows: the question, correct answer and list of wrong answers are sent to the ”GenerateNextMaze()” function, this function then spawns a question room that fits the number of questions passed (As such it is up to the AI algorithm to manage the number of wrong answers passed on). Once the question room is aligned, the question variable within the question room prefab is set to the passed question. A random integer between 0 and the number of questions is then generated, which will represent the pathway that will contain the correct answer. Afterwards, the ”AddCorridors()” function is run taking the random index, correct answer and wrong answers as parameters. This function is in charge of loading new pathways.

Pathway loading works as follows: first, the midpoint of the number of pathways is found. Then for every exit point within the question room starting from the leftmost room till the middle, there is a 2/3 chance of spawning a random type of left pathway, or else there is a 1/3 chance of spawning a centre pathway. Afterwards, the pathway spawned is checked to see if it collides with the previous pathway spawned. If there is a collision then the generation script will try up to 100 times to spawn a different left or centre pathway that does not collide with the previous pathway. If after 100 attempts such a pathway is not found, a centre pathway (Which should NEVER be wide enough to collide with other pathways) is spawned. Afterwards, this process is repeated starting from the very right down till the middle, and this time either using the centre pathway or the right pathway, with a bias towards the right. The reason for this design is to further prevent collisions, by starting with left pathways from the left to the middle, then moving to right pathways from the right to the middle, the chances of running into a collision are significantly reduced.

After each pathway is placed successfully if the pathway contains the correct answer (Which can be found by comparing the index to the random index) then the pathway is validated using a ”ValidateCorridor()” function, otherwise, it is invalidated using an ”InvalidateCorridor()” function. In the prior case, the pathway prefab is marked to contain the correct answer, and the answer is displayed on the prefab entrance, in the latter case, the prefab also contains the wrong answer displayed, but it has an End wall placed at the very end.

Removing Old Rooms

An issue with the Maze Game was that in some cases when the mazes got quite large, the game would start lagging. To avoid facing this issue, once a user completes a question, the question room entrance gets locked off with an End wall. Following this, old question rooms and their respective corridors are de-spawned to save space.

To achieve this, the Maze Generator has a list of a list of objects that contains all the items that need to be de-spawned. Each time a question room, corridor, or end wall is spawned, they are added to the list of list of objects, and each time a new question room is reached, the parent list is incremented. That way once older rooms are closed off, the previous lists of objects (Excluding the current list of objects that would include the current question room and its corridors) would have the objects being references deleted.

A.2.4 Score Manager

The Score Manager is a script within the Game Manager game object that keeps track of the user's score and displays the score at the top-left of the screen. It has 4 main variables: "scorePerLevel" which affects the starting score a user has at the beginning of each Question room, the "currentScore" which holds the current score a user has obtained within a particular Question room, this score is decremented every second till it hits zero, "totalScore" which holds the combination of all the scores a user has obtained at the end of every Question room, and a boolean "firstRoom" which marks if the user is still in the first room, and is set to true by default. While this is set to true, the current score does not get decremented.

Apart from decrementing this score every second, this script also halves the current number of points the user has when a wrong answer is chosen. While other similar smart quiz games like Kahoot for instance opt to give users zero points when a wrong answer is picked, as stated previously this project wishes to explore methods of increased engagement through the normalisation of student results, as such by halving the number of points a user current has instead of setting it to zero, the user still has a chance to climb up the leader board.

The Score Manager is also responsible for requesting new mazes, sending the current score and sending the total score to the AI algorithm. The reason behind this decision is that each time a correct answer is chosen, the AI algorithm would generally need score information in order to understand the performance of a player within that particular question and update its information to adjust the maze accordingly.

A.2.5 Timer

The Timer script is responsible for showing the current time in the format of ”MINUTES: SECONDS” at the top of the screen so a user would know how long they spent playing the Improved Maze Game. The Timer holds the following variables: ”totalTime” which holds the total time that the user has spent within the game (This time is then formatted using a ”FormatTime()” function), a ”timeStamp” floating-point variable that holds the time stamp of when the last question was completed and is updated each time a question is answered correctly, thus getting information on how long it took to complete a question and a ”timerActive” variable which indicates if the timer should update or not. The timer active boolean is controlled via two functions: ”StartTimer()” and ”EndTimer()”.

A.2.6 Ending the Game

Once the final question is reached, the AI algorithm is responsible for running the ”Finalise()” function within the Maze Generator. This causes the maze generator the place a Final Room instead of a question room with corridors at the end of the last correct answer within a maze. Once the user gets tagged by the user tagger in the final room, the AI is responsible for saving all the information needed, and sending it to the web server, afterwards the total score obtained by the user is saved within a Unity Player Pref and the user is moved onto the next scene, which is the End Screen.

End Screen

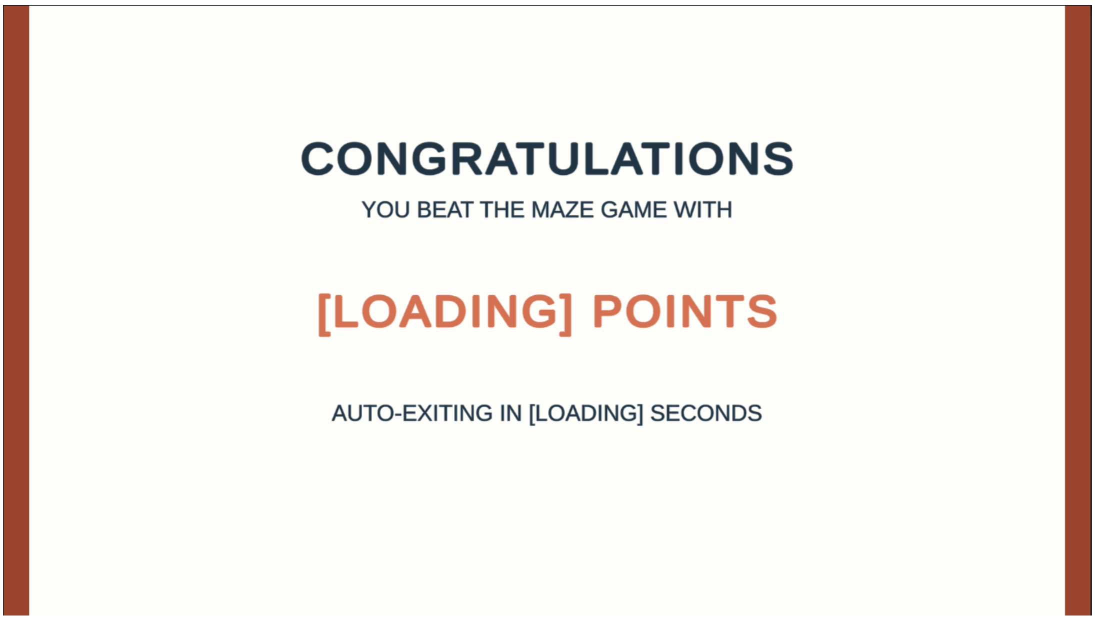


Figure A.10 The End Screen Design of the Improved Maze Game

The End Screen features a simple UI that congratulates the user for completing the Improved Maze game and shows the score obtained by the user. After some time (Within this project 12 seconds) an auto-exit function is run that runs a JavaScript function ”javaScriptGameFinished()”, the exit functionality is then managed by the website, where in this case the user would be redirected to a performance page to view their right and wrong answers.

Appendix B Website

B.1 Website Design Images

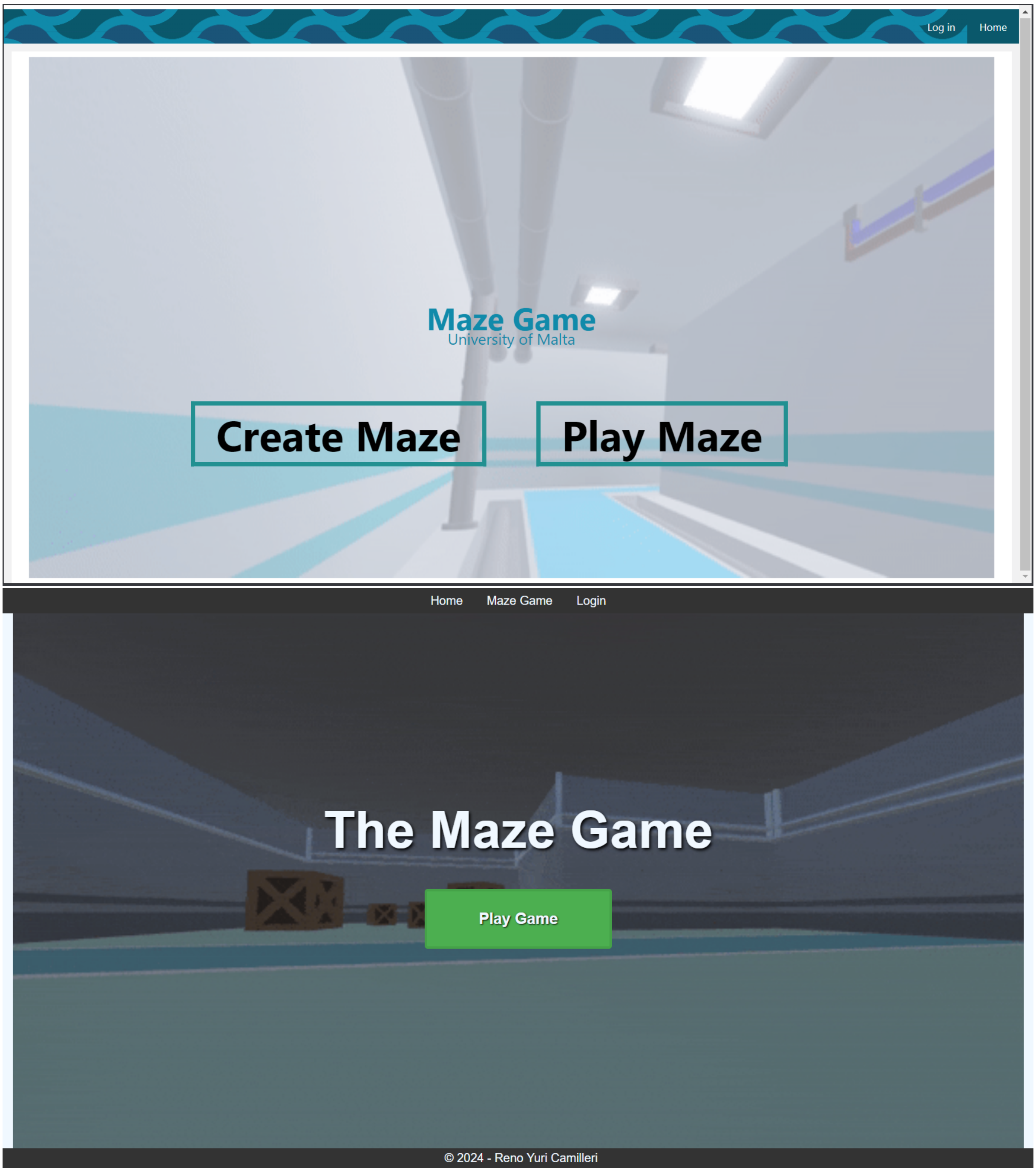


Figure B.1 Index Page - Old (Top) vs New (Bottom) Design for Desktop

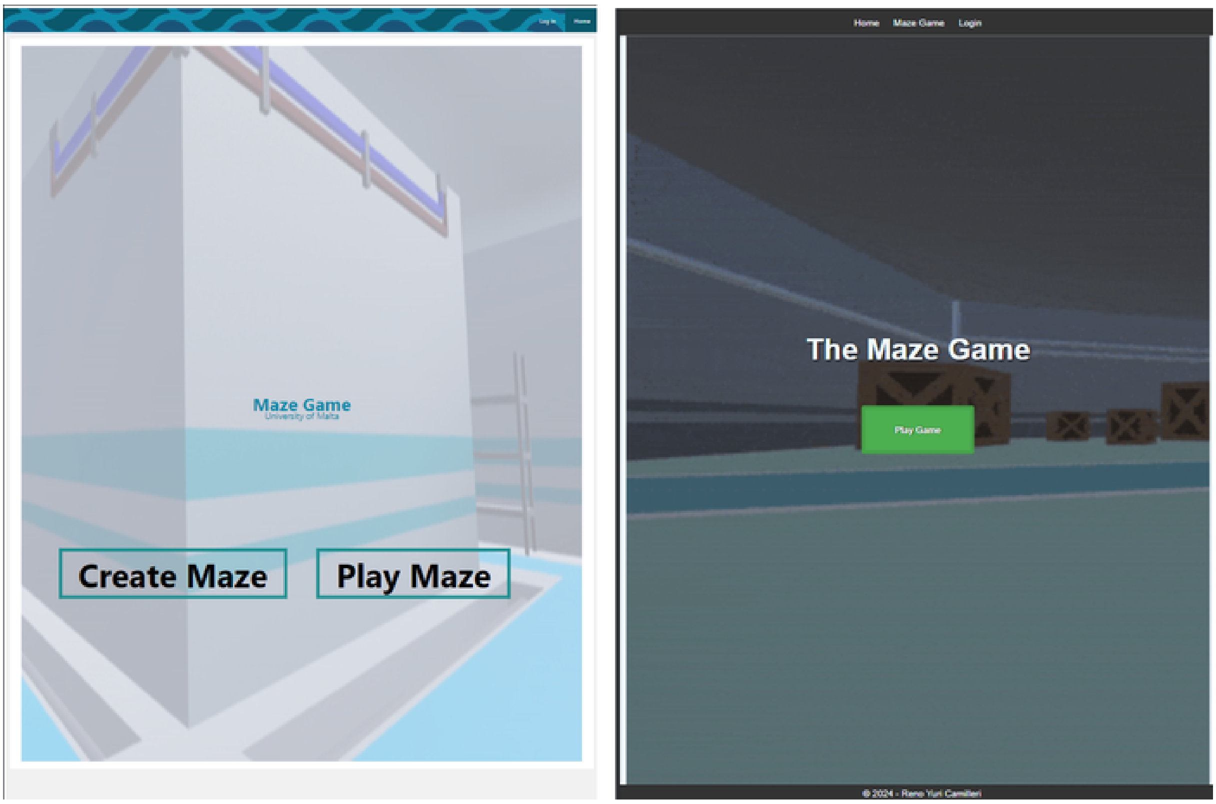


Figure B.2 Index Page - Old (Left) vs New (Right) Design for Tablet

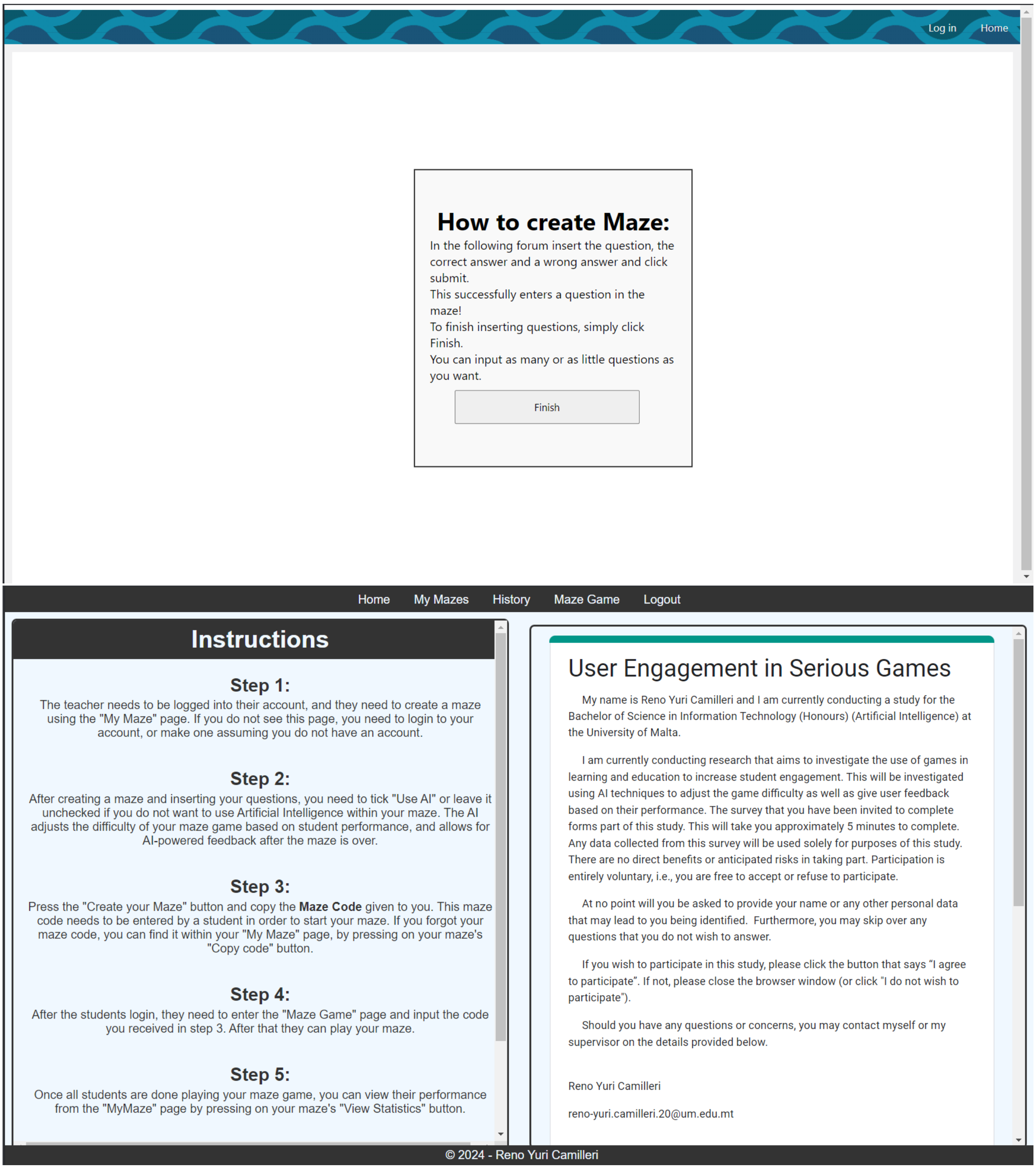


Figure B.3 Instructions Page - Old (Top) vs New (Bottom) Design for Desktop

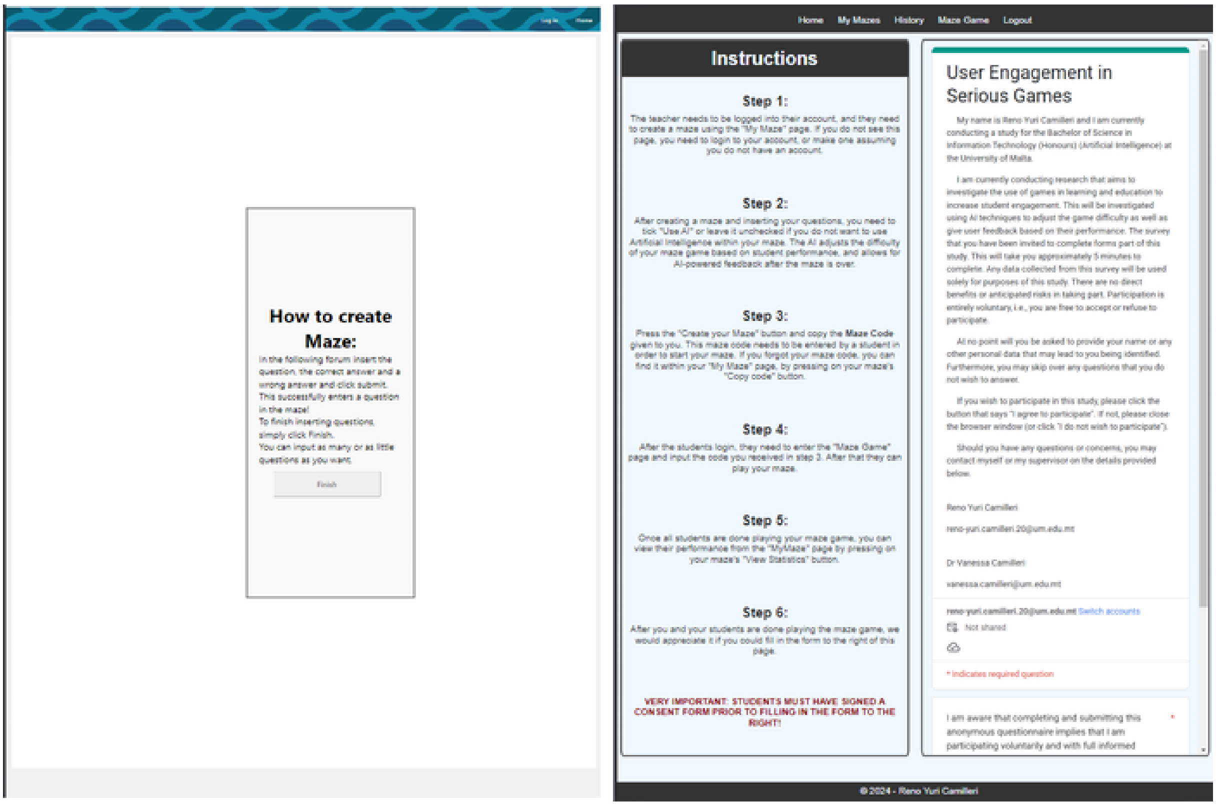


Figure B.4 Instructions Page - Old (Left) vs New (Right) Design for Tablet

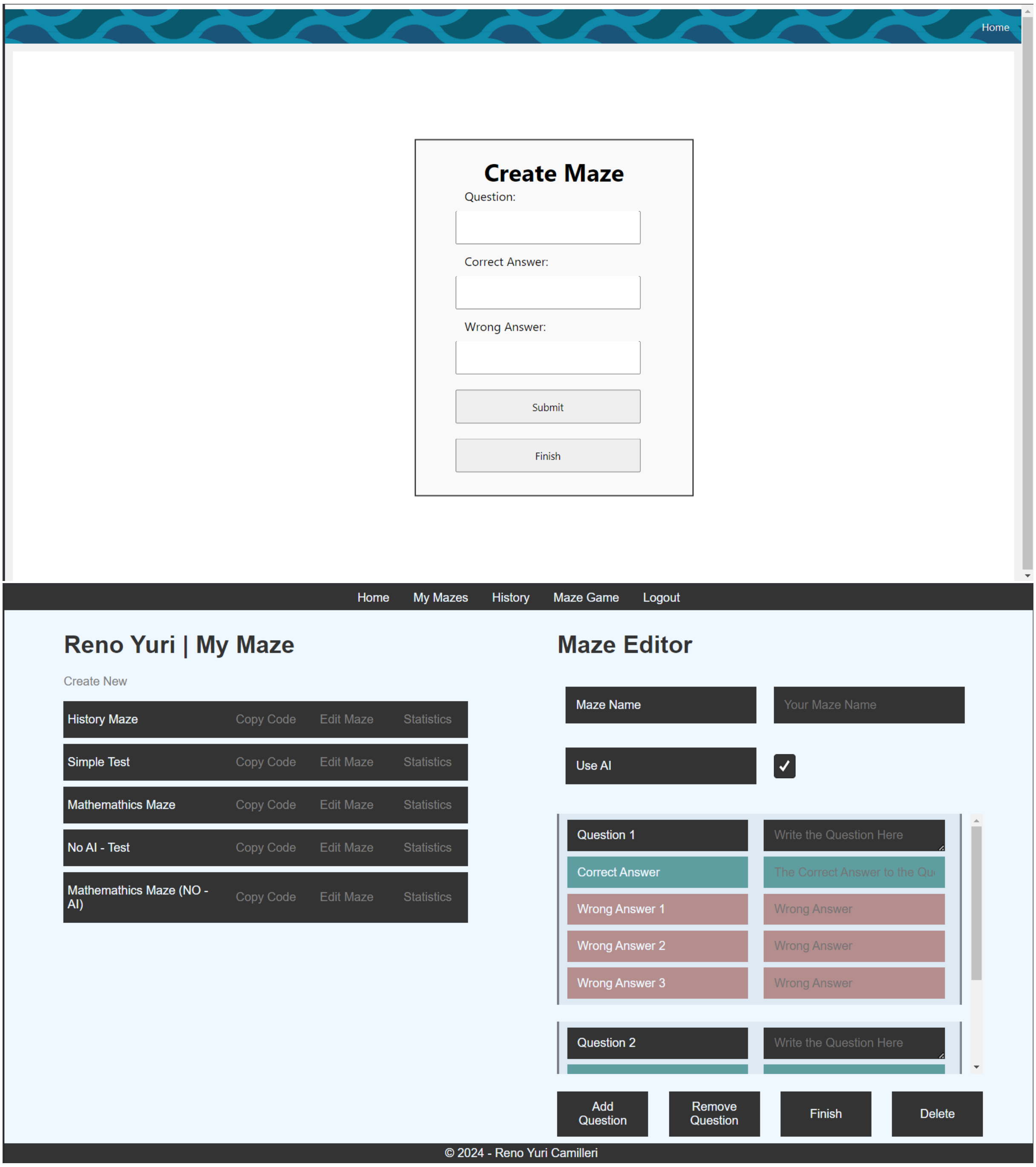


Figure B.5 Question Maker Page - Old (Top) vs New (Bottom) Design for Desktop

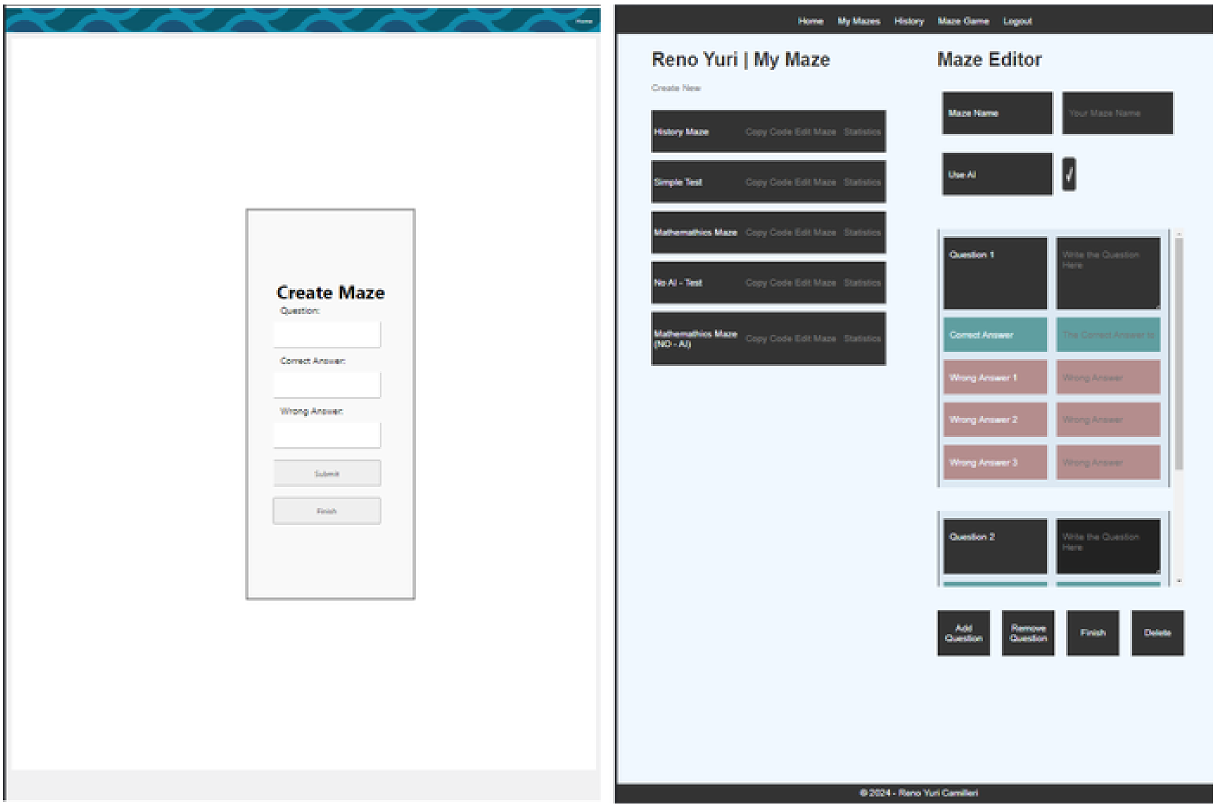


Figure B.6 Question Maker Page - Old (Left) vs New (Right) Design for Tablet

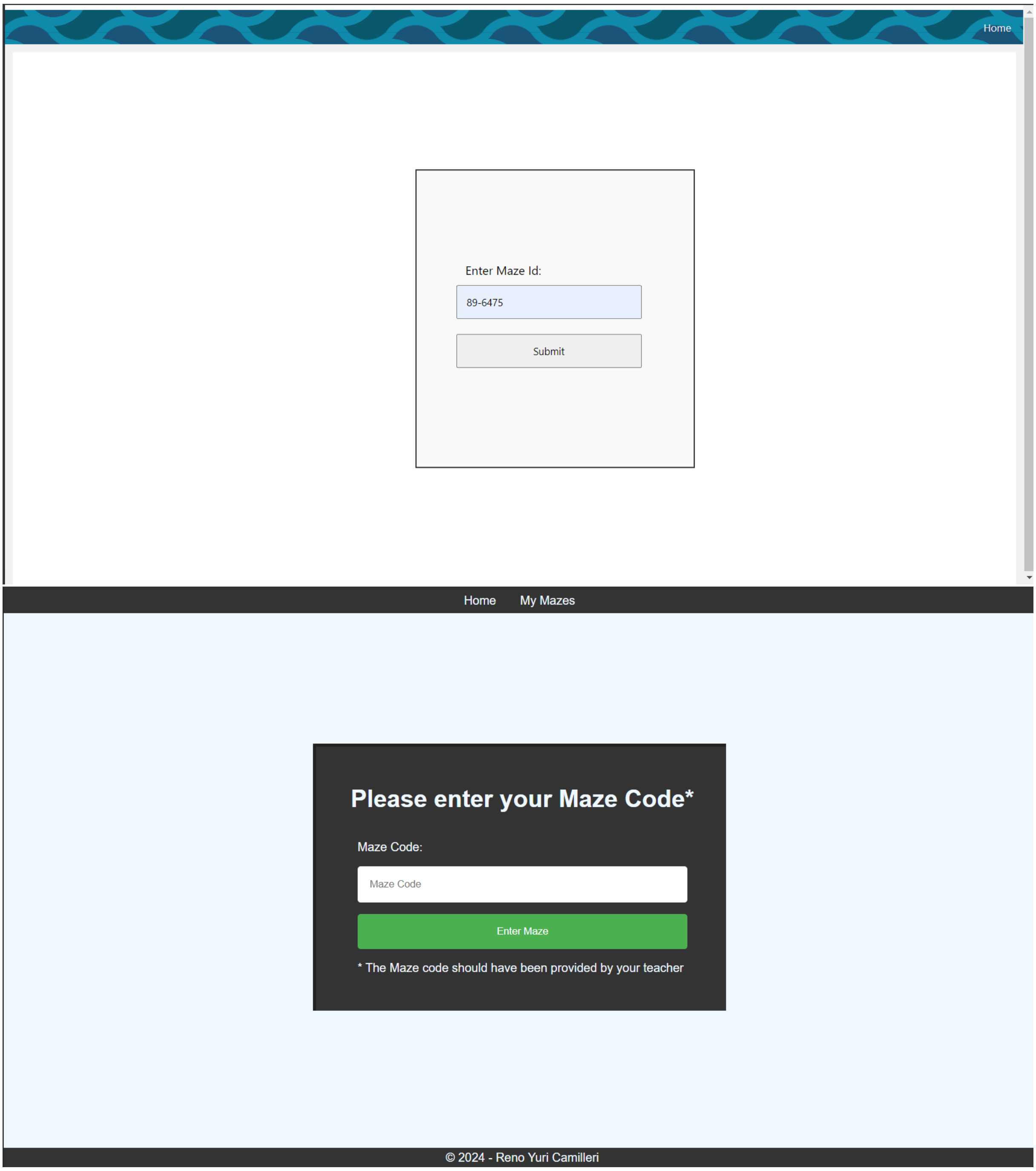


Figure B.7 Enter Maze Code Page - Old (Top) vs New (Bottom) Design for Desktop

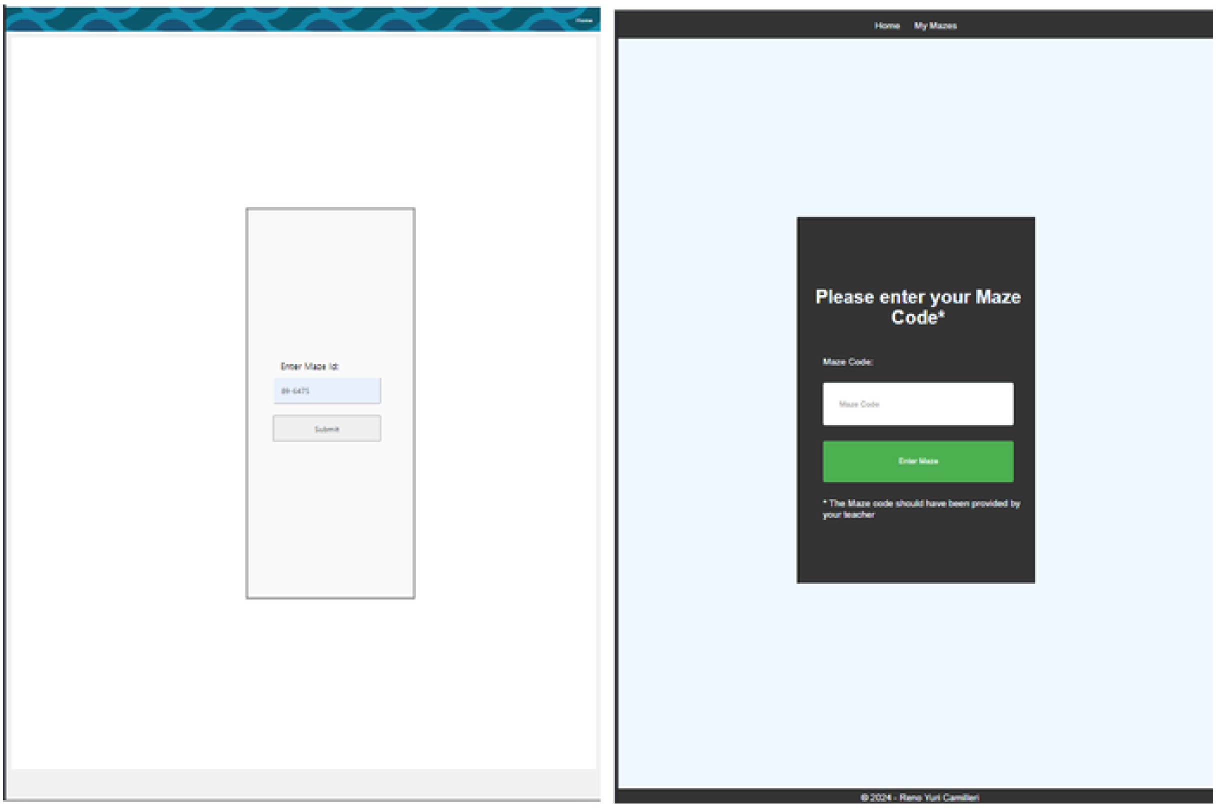


Figure B.8 Enter Maze Code Page - Old (Left) vs New (Right) Design for Tablet

B.2 Website Design Details

The original Maze Game web page featured an index page that scaled based on screen width and featured a lot of fixed options instead of dynamic options. As a result, while the website looked decent on PC monitors of certain sizes, it would be hard to view on other common classroom devices such as tablets or phones for instance. Furthermore, the original website did not process user data obtained from the maze and gave teachers a very simple way of creating mazes. The method given allowed teachers to write a question, right answer and wrong answer then move forward until the maze is ready, and then a code would be given. If the code was lost, or the teacher wanted to change a question or review the list of questions they made, it would not be possible within the old website.

This project aims to completely redesign and add a ton of functionality to the website, allowing teachers an easier and more user-friendly experience to create mazes, adding a new level of Artificial Intelligence in generating additional wrong answers to the maze, and providing users with review exercises, as well as giving users and teachers a way to view class performance in any given maze. Furthermore, while the website was made primarily for desktop and laptop users, this project makes use of dynamic sizes to ensure that each page is still easy to view and access on tablets and phones.

B.2.1 Index Page, Navigation Bar and Footer

Page Design (HTML & CSS):

All the pages within the Improved Maze Game website feature a similar design of splitting the page into 3 main sections, the navigation bar at the top, the main content in the centre, and the footer at the bottom of the page. In terms of the navigation bar, when a user is logged off it features 3 options: ”Home”, ”Maze Game” and ”Login”. While the Maze Game is shown as an option it is mainly there just to show that it will be an available option after login, since you need to be logged in to play the Maze game (To save data based on your performance), as such until you log in, the Maze Game option will simply direct you towards the login page.

After logging in the navigation bar will remove the ”Login” option, and instead add the following additional options: ”My Mazes”, ”History” and ”Logout”. The first option is intended for teachers and allows the teacher to access all the mazes they have created, more info later. The second option allows students to view the history of their completed mazes and all their attempts for a particular maze, as well as the page where a user can view review exercises. Logout is used to log out of your account. In addition to this: the ”Home” option is changed to direct a user to the instructions page instead of the Index Page, and the ”Maze Game” option now directs the user to the game verification page, where a student would need to insert a maze code to play the maze.

The Index Page in itself has a very simple design, the background is an animated GIF, showcasing how the Improved Maze Game looks like, as well as features one button ”Play Game”. This design was chosen to be similar to the design of the original Maze Game that also featured an animated GIF as a background. As previously mentioned, the user needs to be logged in to play the maze game, and since this page is only accessible if a user is not logged in, the user will simply be directed to the login page. Note: the login page features a link to the sign-up page.

Page Scripts (JavaScript & PHP):

Most pages within the Maze Game make use of the ”shared” JavaScript file. This file manages the navigation bar of most pages, and works as follows: the script first checks if a user has a cookie called ”user_id” set, if it is set then the user is logged in, otherwise the user is logged out. If the user is logged in, then the navigation bar will change as described in the Page Design above, to feature the new ”My Mazes”, ”History” and ”Logout” links, as well as change the links of the ”Home” page and ”Maze Game” page.

The Shared JavaScript file also holds the Logout() function which runs when Logout is pressed, this function creates a confirmation window asking the user if they are sure they want to log out, and if confirmation is given then a PHP file called ”process_logout" is run which simply clears the user's cookies, then the user is directed to the Index home page.

The Index Page has its own second JavaScript file called "indexScript" which checks if the user is logged in, if they are then they are redirected to the Instructions page. This script also contains the function used by the "Play Game" button to redirect the user to the login page.

B.2.2 Instructions Page

Page Design (HTML & CSS):

The instructions page as can be seen is made up of two scroll-able frames splitting the main section into two, left and right. On the left-hand side, there are some basic instructions for users and teachers to follow on how to use this website. On the right-hand side, there is an iframe embedding a Google form holding the "User Engagement in Serious Games" form which will be used in the evaluation section of our project.

B.2.3 Login and Sign up Page

Page Design (HTML & CSS):

Both the Login Page and the Sign-Up page feature a form aligned to the centre of the screen. A username and password are required for a user to access their account, whereas the Display Name is used to identify a user within their classroom, for example, let's say three users share the same name, they could put their name as their display name, but then have separate usernames. Within the sign-up page, there is also a "Confirm Password" field for a user to write down their password a second time if they misspelt their password on the first attempt.

The login page has a link to the sign-up page below the form, whereas the sign-up page has a link to the login page below the form. It is common practice to either make the sign-up page its own option in the navigation bar or else place it as an option within the login page (The latter being what this project did), as most users associate the two pages to be connected. Both forms have a button to submit at the end.

Page Scripts (JavaScript & PHP):

The login page makes use of the JavaScript file "loginScript" and the PHP file "process_login". Starting from the JavaScript, the script upon loading adds an event listener to the form so that if the "Enter" key is pressed login would be initiated, as a simple keyboard shortcut instead of having to press "Login". Together with this, the script features a Login() function which requests the PHP to check if the login details belong to an account.

If the PHP script returns a success status, then the user is taken to the instructions page, otherwise, the error returned by PHP is shown such as ”Invalid Username or Password”. After the error is shown, the password sent by the user is erased from the field for additional security. It is important that while JavaScript can be used to verify information since JavaScript runs locally, it can be manipulated by users with technical experience, so all verification checks should occur mainly in PHP since it is server-sided and difficult to manipulate by a user.

The PHP process_login script checks if the username and password match with the username and password placed inside the database, PHP's built-in ”password_verify()” function should be used to compare the password sent with the password saved since as an additional layer of security the user password obtained via sign up will be hashed.

The sign-up page makes use of the JavaScript file ”signupScript” and the PHP file ”process_newAccount”. Starting from the JavaScript, the script just like the login script allows a user to press enter to submit the form without needing to press the button manually. The script also features several functions to perform the following validation commands prior to passing on the details to PHP:

- Remove spaces from the start of the Display Name.
- Obtain all values from all the elements filled in by the user, and return an appropriate error if one of the elements is not filled in.
- The Display name should be between 1 and 30 characters, contain at least one letter and may only contain letters, numbers, and spaces.
- The Username should be between 6 and 20 characters, should not contain spaces and can only contain letters and numbers.
- The Password should be between 6 and 20 characters, should not contain spaces, should contain at least one number, should contain at least one letter and can only contain letters, numbers, ”!”, ”-”, or ”_”.
- The Username may not be the same as the password.
- The Password and the Confirm Password must be the same.

While this may not necessarily be the best way of verifying that an account has a very strong username and password, it should ensure a simple level of security for user accounts.

In the PHP script, in addition to server-side checks, the password is hashed using PHP's password_bcrypt function with a cost of 12, the cost was added as another server-side security feature to stop bots from brute-forcing the website by slowing down the response time. The password is then hashed, and then the database is checked for if the username sent already exists, if the username exists, then the process is aborted, otherwise, the new account is made using the details provided.

B.2.4 Maze Creator 'My Maze' Page

The My Maze page is split into two main aspects, which are the Maze list and the Maze Editor. When the page is opened only the Maze List can be viewed expanded and in the centre, so that teachers can view their mazes, and copy the maze code or view statistics for a particular maze without needing to see the editor for no reason. Should the editor be required either by pressing "Create New" or "Edit Maze" on one of the mazes, the Maze Editor would create open up an empty maze format, or open the respective maze.

When opening an existing maze, in addition to filling in all the values, the new AI-generated wrong answer is shown as an additional field. If "Use AI" is disabled then the new AI wrong answer is simply shown as "AI Disabled".

Once a new Maze is made or updated, a PHP script is run that either updates the list of questions or creates new ones. Depending on whether AI is enabled or not, the PHP script would also make a call to the OpenAI API to make use of GPT 3.5-Turbo to generate a new wrong answer, and that answer is saved in the database in addition to the other answers.

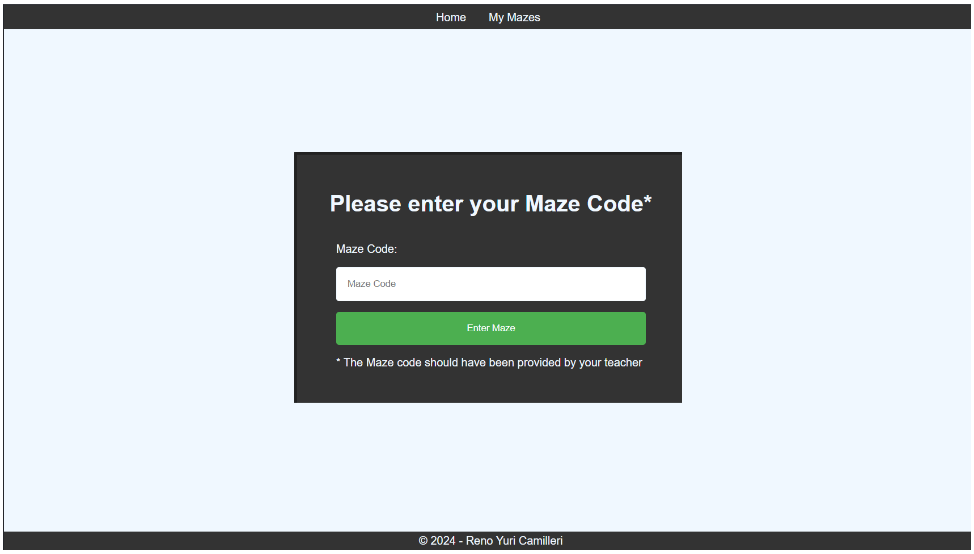
B.2.5 Maze Performance Page

The Maze Performance Page features the history of mazes that the user has done, to avoid cluttering the screen, the user's attempts are hidden and can only be viewed once a particular maze is selected, after the attempts are displayed, a user can view their performance in a maze's particular attempt. From the performance page, the user can view all the correct answers, the time it took them to answer the question, the points they made, and the number of mistakes they made assuming they made a mistake. At the top of the page under "Overall Performance" the user can go to the Maze Statistics Page to view the class performance for a particular maze. Additionally, if AI is enabled, at the bottom of the page the user can perform AI-powered review exercises (More info in the section "Building an AI Feedback Layer").

B.2.6 Maze Statistics Page

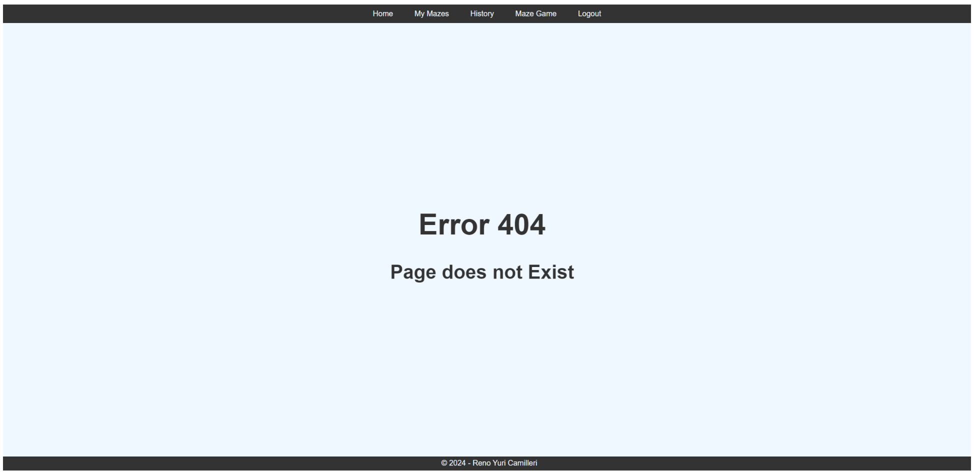
The Maze Statistics page features all the users that finished a particular maze, all placed in order of ranking from first to last place. The top 3 users have their colours set to Gold, Silver and Bronze respectively. In addition to this, the scores obtained by each user are shown as well as what attempts they currently have undergone. Badges achieved by users are also displayed here, the current badges available in the game are for the top 3 fastest players, and the top 3 players with the least mistakes. If two users are tied for a particular badge, they both obtain the same badge.

B.2.7 Game Verification Page



To stop users from accessing random Mazes they should not have access to, each maze employs a unique code, a teacher should send this code to their students, and the students would insert the code inside this page. The code is then checked if it is valid or not, if invalid an error would return, if valid then the maze game with that particular code is loaded.

B.2.8 404 Page



To make the page more user-friendly a custom 404 page was made, this page was set to open up in XAMPP if a user ever typed in a wrong web link.

B.2.9 Play the Game Page

Within this page the actual game loads up, and the user can play the game. It is worth noting that, all JavaScript functions mentioned in the Game section of the Methodology should be implemented in the JavaScript of this website.

Appendix C Detailed Overview of AI Algorithms

C.1 Overview of Reinforcement Learning

Moving onto the 3rd objective outlined, in addition to the ”GPT-3.5” algorithm mentioned in the ”My Maze” section of the website, which offers teachers an easier experience to create more options in the serious game environment, another algorithm that is user-focused would be the RL algorithm.

The RL algorithm is implemented as the AI algorithm mentioned within Chapter 4.2 and manages the Maze Generation, starting/ending the game, loading questions obtained from the website database, saving wrong answers in a list, converting lists to a JSON string, and exporting player performance data back to the database.

The RL algorithm chosen for this project was a Q-Learning algorithm with an Epsilon Greedy strategy, this algorithm was chosen as opposed to IRT and BKT for various reasons such as: the recommendations made to use RL algorithms within Background Research, as well as due to it being very simple to implement, yet being highly efficient performance wise. In addition to this RL is a much more commonly used algorithm in video games, and has increased in popularity tremendously over recent years.

The reason an AI approach was chosen as opposed to hard coding scores to decide what room to spawn was to future-proof the Improved Maze Game. If the values were to be hard-coded then it would be harder to edit the game if a developer wished to change the layout of various corridors and question rooms, then the developers would need to run the game multiple times and estimate how long it takes to beat each room, and replaced the hard-coded value manually, in addition to this, if new question room types were to be introduced, then apart from adding new if statements, each value in prior question rooms would also need to be adjusted. On the other hand, by using AI, a developer can temporarily increase the learning rate of the RL algorithm and the algorithm would take care of all these issues on its own while the developers are simply play-testing the game. In addition to this, as previously mentioned, the Q-learning algorithm is very simple, so in the long run, it would still be more code efficient than a multitude of if statements.

Within this project, the Q-Learning algorithm only uses 5 different states (”VeryLow”, ”Low”, ”Average”, ”High” and ”VeryHigh”) which represent the performance of a student, with 3 different actions (3, 4, 5) which represent the number of choices within a question room. The result of every state and action is the predicted score a user would gain from that instance. Using this predicted score, the AI estimates the median score a user would get by completing the maze and pushes each user to obtain that score.

The RL algorithm works as follows, at the start the Start() function is run to get the authorisation code (ie code containing the maze ID) from the website. Once this code is obtained a coroutine: ”GetQuestionsFromPHP()” is called using a link to the PHP function that loads question data, this link is accessed using the authorisation code previously obtained. The PHP function then returns a JSON containing three very important pieces of information: the QTable which is parsed and loaded within the program, the Questions which are also loaded within the program, and whether AI needs to be used. If AI doesn't need to be used, then the programs run normally without taking into account the RL algorithm, and at the end, an empty QTable is sent back to the program (Which does nothing) otherwise the RL algorithm is used, and at the end a QTable containing the updated values is sent back to the website to slightly increment the QTable towards the optimal RL QTable for the Improved Maze Game.

Once everything is loaded the function ”FirstQuestion()” is run, the reason this was made into its own function was to ensure that the program waits for everything to be loaded via the coroutine before running the rest of the program. This function calls the Maze Generator to generate the first question room. As previously mentioned, this algorithm is only in charge of making decisions and passing on its decisions to the Maze Generator, then the Maze Generator takes care of the rest. To that end, this script holds functions in place so that each time a score is obtained, the program identifies the user's performance as one of the 5 states previously mentioned, and then updates the QTable based on the score obtained.

In terms of the QTable used by the algorithm, it is saved within the database as its own table. In addition to the questions of a maze being sent to the game, the QTable is also sent as a JSON string to the maze game and is loaded as a QTable dictionary within the game. As the user plays the game the scores obtained are sent to the algorithm, which updates the QTable via a small learning rate, and at the end of the game the new QTable is sent back to the database, using a very small learning rate set by the developer, the database's QTable is slightly pushed towards the new QTable. This is done as an additional method of updating the performance of the QTable.

At the end of the program in addition to the QTable, a lot of user performance data is also sent which includes the: user mistakes (A list of all wrong choices), the score obtained in each question, the time taken to complete each question, the correct answer, question and question ID. The script makes use of an external class to encapsulate the mentioned data. Then using the Unity web request the data is sent in addition to whether or not AI was used and the maze ID to a PHP script called ”maze_finished.php” which processes the data obtained and saves it in appropriate tables.

C.2 Overview of the Feedback System

The AI feedback layer works using GPT 3.5 turbo, while its performance can be enhanced using the GPT 4 model, due to GPT 4 costing over 20 times more than the 3.5 turbo model (As of the writing of this thesis), it is not worth it from a financial standpoint to go for the GPT 4 model when the performance of the two is satisfactory with both versions of the model. The Feedback layer works as follows, all the questions of a particular maze are loaded, a question is chosen at random, and then the GPT API is called with the following prompt:

" "Given the following question: "(QUESTION)", use that question as a parameterised question, and generate a new question of about "(DIFFICULTY)", with a correct answer, 3 wrong answers, a hint and a short explanation as to why an answer is correct. Try and make a new question of the same genre as the parameterised question. Return your response as a JSON output with the following structure: "{"review": {"question": "YOUR_QUESTION", "correct": "YOUR_ANSWER", "incorrect_1": "YOUR_WRONGANS", "incorrect_2": "YOUR_WRONGANS", "incorrect_3": "YOUR_WRONGANS", "hint": "INSERT SHORT HINT", "why_correct": "INSERT SHORT EXPLANATION AS TO WHY CORRECT"} }" "

The Question parameter in bold is replaced by the question that is chosen, whilst the difficulty is set to one of the following strings based on the current performance of the user: "an easier difficulty", "a slightly easier difficulty", "the same level in terms of difficulty" or "a slightly harder difficulty". While these may not necessarily be the best prompt for engineering the feedback layer, from a lot of testing, these prompts generally produce the best outcome. The resulting JSON response of the GPT API has always been in the right format, whereas answers are correct most of the time (There have been a few cases where the response returned was not correct, please refer to Appendix D for more details). A JSON response by the API was also chosen to avoid making too many calls to the API, as more calls result in a higher expense for tokens, and take more time to process.

In terms of calling the API, the following changes were made: the "temperature" variable was set to 0.7, "top_p" was set to 0.7 and the "seed" is constantly randomly generated. The reason behind this choice was to increase the randomness of responses obtained, as the API would constantly generate the same parameter questions for the same question. If the temperature or top_p values are set too high, there is an increase in the chance of obtaining wrong answers, whilst having a small value results in repetitiveness. As such 0.7 was found after some testing to be a good value to balance out both aspects.

Once the JSON response is returned, javascript handles hiding the previous hint given by the AI assuming this isn't the first call. All the data is put into place, ie, a random button is chosen to hold the correct answer. The correct answer is placed there as well as changing the button to be marked to run a correct answer function, whereas all the other buttons are invalidated to run an incorrect answer function, as well as hold the wrong answer. The hint is added but made hidden, and can only be accessed if the user decides to press the view hint button. This is an optional thing a user can do to view more information.

If a correct answer is chosen then the program congratulates the user prints an explanation as to why an answer was correct and loads the next question. Otherwise, it informs the user that they were incorrect and explains what the correct answer was and why it was correct. During this process, the program also marks if the answer is correct or incorrect within javascript (as review exercises are based on the user's desire to learn so there is no need to allocate the server to handle this information).

The Feedback layer also makes use of scaffolding where if the user gets three more questions correct than incorrect, the difficulty increases, whereas on the other hand, if the user gets three more questions incorrect than correct the difficulty decreases, upon difficulty increase / decrease, this count is reset. The number of questions given to the user depends on the number of questions within the maze, but is never higher than 20, if a user desires they may attempt review exercises multiple times. 20 was chosen as a base number for this project. However, this value isn't particularly important as any number would work in this situation, but it is important not to make the number too high, as a user might get bored and want to stop performing these exercises. Similarly, if the number is too low the scaffolding wouldn't work properly.

Appendix D Quiz Given to Participants

D.1 Quiz Questions

Question 1

Arrange the four digits below to get the largest four digit number which is divisible by 5. ”5, 2, 3, 8”

Correct Answer: 8325
Wrong Answers; 5823, 8352 & 2835
AI Additional Wrong Answer: 2583

Question 2

A television programme starts at 10 minutes to ten. It lasts for 40 minutes. At what time does it finish?

Correct Answer: 10:30
Wrong Answers; 10:40, 11:30 & 11:40
AI Additional Wrong Answer: 11:00

Question 3

Each exterior angle of a regular polygon is 40°. How many sides does this polygon have?

Correct Answer: 9 Sides
Wrong Answers; 4 Sides, 5 Sides & 3 Sides
AI Additional Wrong Answer: ”6 Sides” **(NOTE: ”” quotations added by AI.)**

Question 4

A tank contains 12 litres of oil. The oil is emptied into containers holding 300 ml of oil. How many containers are filled?

Correct Answer: 40 containers
Wrong Answers; 30 containers, 300 containers & 80 containers
AI Additional Wrong Answer: 50 containers

Question 5

Express 200 grams as a percentage of 2 kilograms.

Correct Answer: 10%
Wrong Answers; 20%, 50% & 5%

AI Additional Wrong Answer: 25%

Question 6

What value of x makes the following equation true? 10 million = 10x

Correct Answer: 1,000,000
Wrong Answers; 100,000, 10,000 & 1000
AI Additional Wrong Answer: 100

Question 7

What prime number is larger than 30 and smaller than 40?

Correct Answer: 37
Wrong Answers; 39, 32 & 29
AI Additional Wrong Answer: 35

Question 8

James spends from 3:40pm to 4:35pm on his Maths homework and from 5:50pm till 6:25pm on his French homework. How much time does he spend on his homework for these two subjects?

Correct Answer: 90 minutes
Wrong Answers; 100 minutes, 115 minutes & 130 minutes
AI Additional Wrong Answer: 105 minutes

Question 9

Which of the answers below is the value of $5^2 \times 5^{-2}$

Correct Answer: 1
Wrong Answers; 5, 25 & 10
AI Additional Wrong Answer: 20

Question 10

Find the total cost of 15 pens at €2.25 each, and 15 diaries at €5.75 each.

Correct Answer: €120
Wrong Answers; €100, €150 & €105
AI Additional Wrong Answer: €130

Question 11

4.3 m = ? mm

Correct Answer: 4300 mm
Wrong Answers; 43,000 mm, 0.0043 mm & 43 mm
AI Additional Wrong Answer: 430 mm

Question 12

60 hours = ? days

Correct Answer: 2.5 days
Wrong Answers; 2 days, 3 days & 2.75 days
AI Additional Wrong Answer: 2.25 days

Question 13

When written in standard form, 3,500,000 = ?

Correct Answer: 3.5×10^6
Wrong Answers; 35×10^5, 7×5^6 & 3.5×10^5
AI Additional Wrong Answer: $350,000$

Question 14

A prime number between 15 and 20 is ?

Correct Answer: 17
Wrong Answers; 23, 13 & 15
AI Additional Wrong Answer: 14

Question 15

A television programme starts at twenty minutes to ten. It lasts for half an hour. What time does it finish?

Correct Answer: 10:10
Wrong Answers; 11:10, 11:30 & 10:30
AI Additional Wrong Answer: 10:20

Question 16

What is the value of 3.87456 km correct to the nearest metre?

Correct Answer: 3875 m
Wrong Answers; 3874.56 m, 3880 m & 3874.6 m
AI Additional Wrong Answer: 3873 m

Question 17

The population of a country is 2,673,840,000. What is this number correct to the nearest million?

Correct Answer: 2,674,000,000
Wrong Answers; 2,680,000,000, 2,700,000,000 & 3,000,000,000
AI Additional Wrong Answer: 2,670,000,000

Question 18

Write 38% as a fraction in its lowest terms.

Correct Answer: 19 / 50
Wrong Answers; 38 / 100, 9.5 / 1000 & 3 / 5
AI Additional Wrong Answer: 0.38

Question 19

Solve the equation 3x − 15 = 1 − x

Correct Answer: x = 4
Wrong Answers; x = 5, x = 15 / 4 & x = 6
AI Additional Wrong Answer: x = 10

Question 20

The probability that Lee is late for work is 0.12. What is the probability that Lee arrives at work on time?

Correct Answer: 0.88
Wrong Answers; 0.98, 0.92 & 0.82
AI Additional Wrong Answer: 0.22

D.2 Review Questions Generated

The review section is entirely managed by the GPT model, and as such there is no sequence in the order of review questions generated. While uncommon, there were cases where the GPT model gave incorrect answers. Below are 10 randomly taken review questions.

Review Question Given - Sample 1

Find the total cost of 10 pens at €1.50 each, and 10 diaries at €4.50 each.

GPT's Hint: Calculate the cost of pens and diaries separately before adding them together.

GPT's Correct Answer: €60

GPT's Wrong Answers; €50, €70 & €55

GPT's Answer Response: (Correct / Incorrect). To find the total cost, multiply the cost of 10 pens (10 x €1.50 = €15) and the cost of 10 diaries (10 x €4.50 = €45), then add the two amounts together (€15 + €45 = €60).

Review Question Given - Sample 2 (Wrong Ans)

A television programme starts at ten minutes past three. It lasts for 45 minutes. What time does it finish?

GPT's Hint: Remember to add the duration of the programme to the starting time.

GPT's Correct Answer: 4:00 PM

GPT's Wrong Answers; 3:45 PM, 4:15 PM & 3:45 PM

GPT's Answer Response: (Correct / Incorrect). Adding 45 minutes to 3:10 PM gives us 4:00 PM, which is when the programme finishes.

Review Question Given - Sample 3

Write 25% as a fraction in its lowest terms.

GPT's Hint: Think about the relationship between percentages and fractions.

GPT's Correct Answer: 1/4

GPT's Wrong Answers; 1/5, 2/5 & 3/5

GPT's Answer Response: (Correct / Incorrect). To convert a percentage to a fraction, you simply place the percentage as the numerator over 100 in the denominator. Then, simplify the fraction to its lowest terms.

Review Question Given - Sample 4

Arrange the three digits below to get the largest three digit number which is divisible by 2. 4, 7, 9

GPT's Hint: The largest digit should be placed in the hundreds place.

GPT's Correct Answer: 974

GPT's Wrong Answers; 947, 749 & 479

GPT's Answer Response: (Correct / Incorrect). To form the largest three digit number divisible by 2, the largest digit (9) should be placed in the hundreds place to maximise the number.

Review Question Given - Sample 5

Express 150 grams as a percentage of 1.5 kilograms.

GPT's Hint: To calculate the percentage, divide the given weight in grams by the weight in kilograms and then multiply by 100.

GPT's Correct Answer: 10%

GPT's Wrong Answers; 5%, 15% & 20%

GPT's Answer Response: (Correct / Incorrect). To express 150 grams as a percentage of 1.5 kilograms, we first convert both to the same unit (grams) by multiplying 1.5 kilograms by 1000 to get 1500 grams. Then, we divide 150 grams by 1500 grams and multiply by 100 to get 10

Review Question Given - Sample 6

Find the total cost of 20 notebooks at €3.50 each, and 10 pencils at €0.75 each.

GPT's Hint: To find the total cost, multiply the quantity of each item by its price and then add the results together.

GPT's Correct Answer: €77.50

GPT's Wrong Answers; €65.00, €82.50 & €70.00

GPT's Answer Response: (Correct / Incorrect). The correct answer is obtained by multiplying 20 notebooks by €3.50, which equals €70, and multiplying 10 pencils by €0.75, which equals €7.50. Adding these results together gives a total cost of €77.50.

Review Question Given - Sample 7

A perfect square between 20 and 30 is?

GPT's Hint: It is a number that when multiplied by itself gives the result.

GPT's Correct Answer: 25

GPT's Wrong Answers; 22, 28 & 30

GPT's Answer Response: (Correct / Incorrect). 25 is a perfect square because 5 multiplied by 5 equals 25.

Review Question Given - Sample 8

Solve the equation 2x + 7 = 3x - 5

GPT's Hint: Try isolating the variable x on one side of the equation.

GPT's Correct Answer: x = 12

GPT's Wrong Answers; x = -2, x = 5 & x = 8

GPT's Answer Response: (Correct / Incorrect). The correct answer is x = 12 because by moving the terms involving x to one side and simplifying, you get x = 12 as the solution.

Review Question Given - Sample 9

Each interior angle of a regular polygon is 140°. How many sides does this polygon have?

GPT's Hint: Think about the relationship between the interior angles of a polygon and the number of sides.

GPT's Correct Answer: 9

GPT's Wrong Answers; 6, 8 & 10

GPT's Answer Response: (Correct / Incorrect). In a regular polygon, the interior angle can be calculated using the formula (n-2) * 180 / n, where n is the number of sides. Setting this equal to 140° and solving for n gives us n = 9.

Review Question Given - Sample 10

7.6 km = ? m

GPT's Hint: Remember that 1 km is equal to 1000 m

GPT's Correct Answer: 7600 m

GPT's Wrong Answers; 760 m, 76 m & 76,000 m

GPT's Answer Response: (Correct / Incorrect). To convert from kilometres to meters, you need to multiply by 1000 since 1 km is equal to 1000 m. Therefore, 7.6 km is equal to 7.6 x 1000 = 7600 m

D.3 Instructions Given to Students

AI Student Instructions:

1. Make an Account
2. Login
3. Enter the Maze Game with the Code: 19_66179
4. Perform Post-Quiz Exercises
5. Make a simple 2-question quiz
6. Fill in the Survey (AI version)

Non-AI Student Instructions:

1. Make an Account
2. Login

3. Enter the Maze Game with the Code: 37_90667
4. Make a simple 2-question quiz
5. Fill in the Survey (Non - AI version)

Appendix E Results Obtained

All results featured hereunder have been changed to keep the users anonymous as part of this project's commitment to ethics. In addition, the responses are not placed in order, in other words, the first response to a particular question may or may not belong to the same individual as that of the first response to another question.

E.1 Results Obtained from the Game

AI Maze Game

Mathemathics Maze - Statistics

Position	Display Name	Points Obtained	Attempt Number	Badges
1	[redacted]	1438	1	
2	[redacted]	1409	1	
3	[redacted]	1394	1	
4	[redacted]	1212	1	
5	[redacted]	1118	1	
6	[redacted]	1102	1	

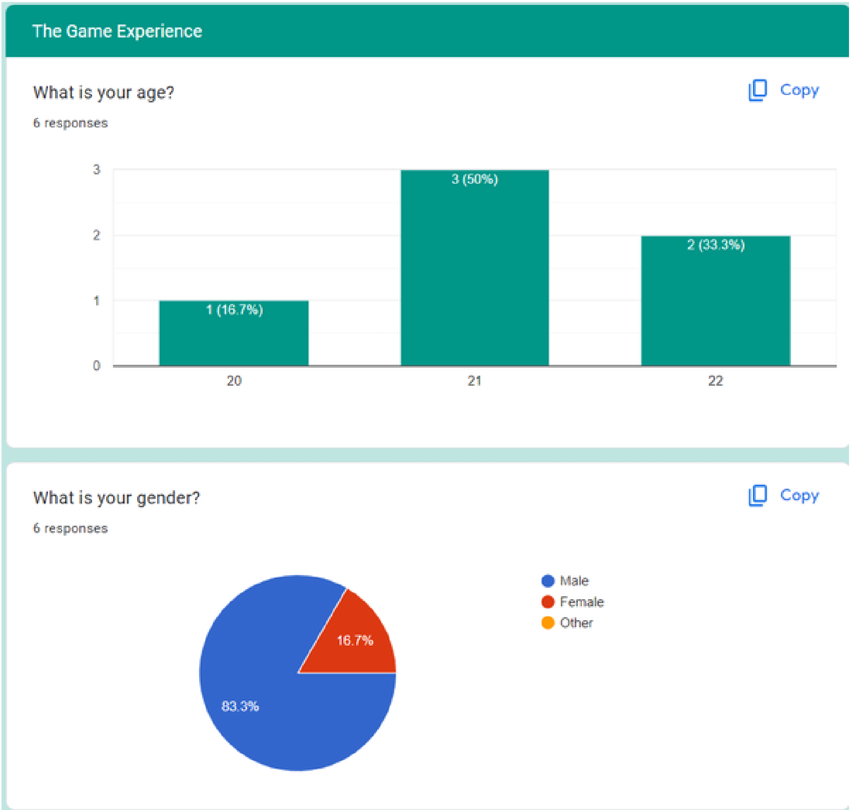
Non AI Maze Game

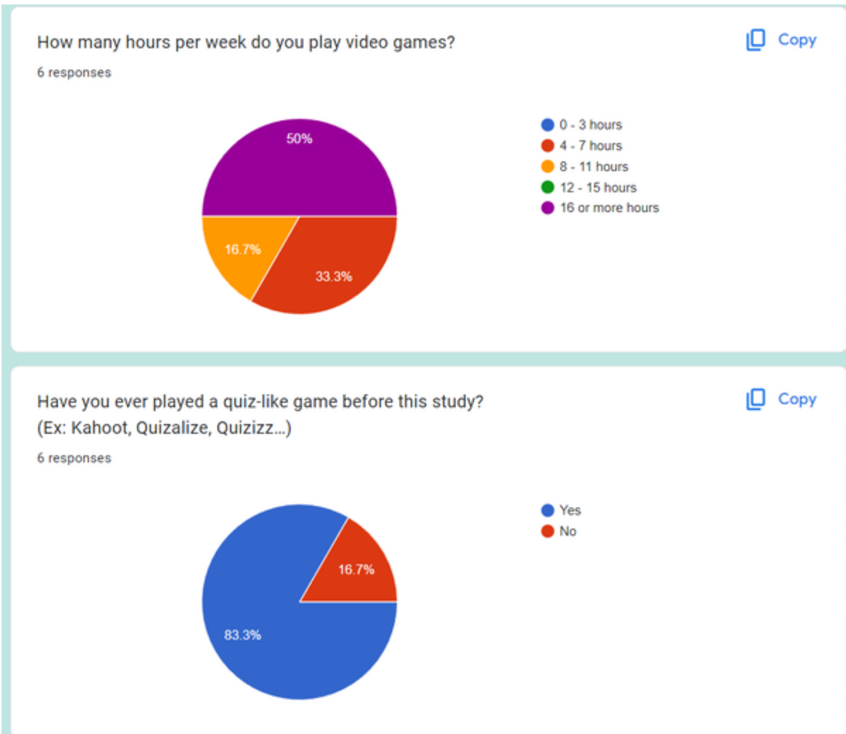
Mathemathics Maze (NO - AI) - Statistics

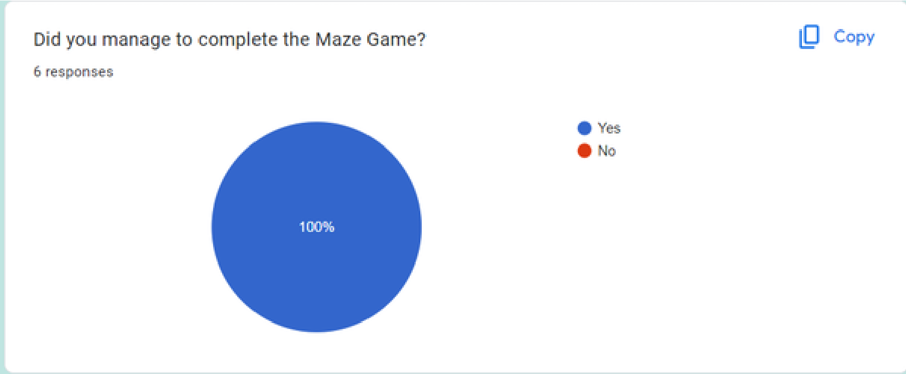
Position	Display Name	Points Obtained	Attempt Number	Badges
1	[redacted]	1538	1	
2	[redacted]	1460	1	
3	[redacted]	1380	1	
4	[redacted]	1246	1	
5	[redacted]	1135	1	
6	[redacted]	1021	1	
7	[redacted]	961	1	

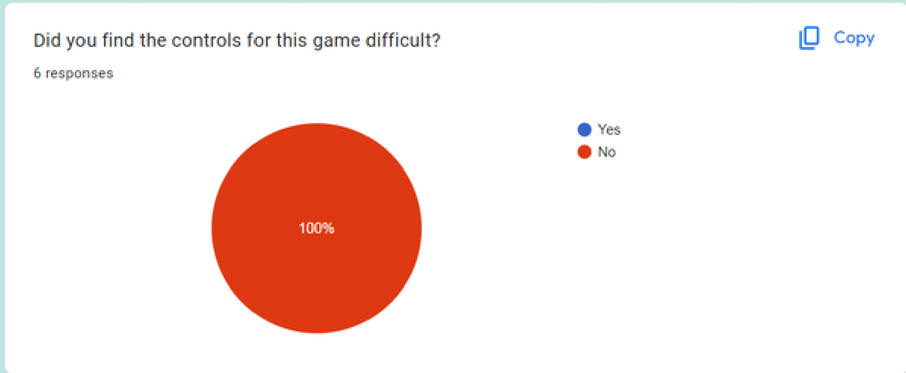
E.2 Results Obtained from the Surveys

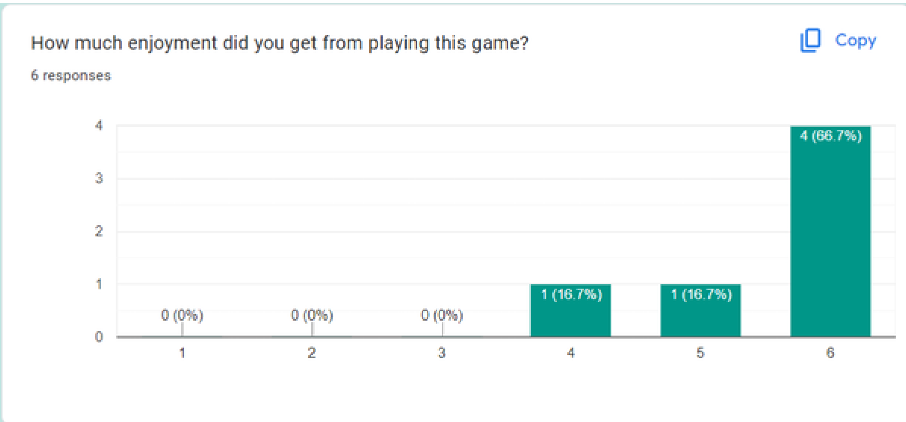
E.2.1 AI Group

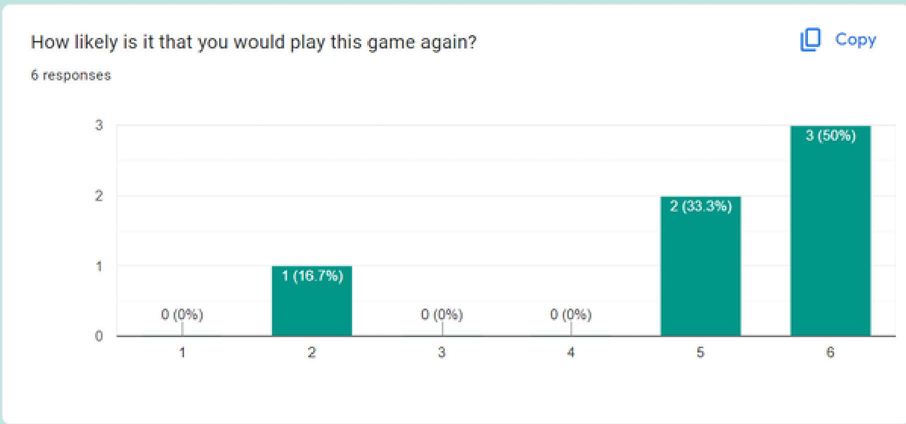


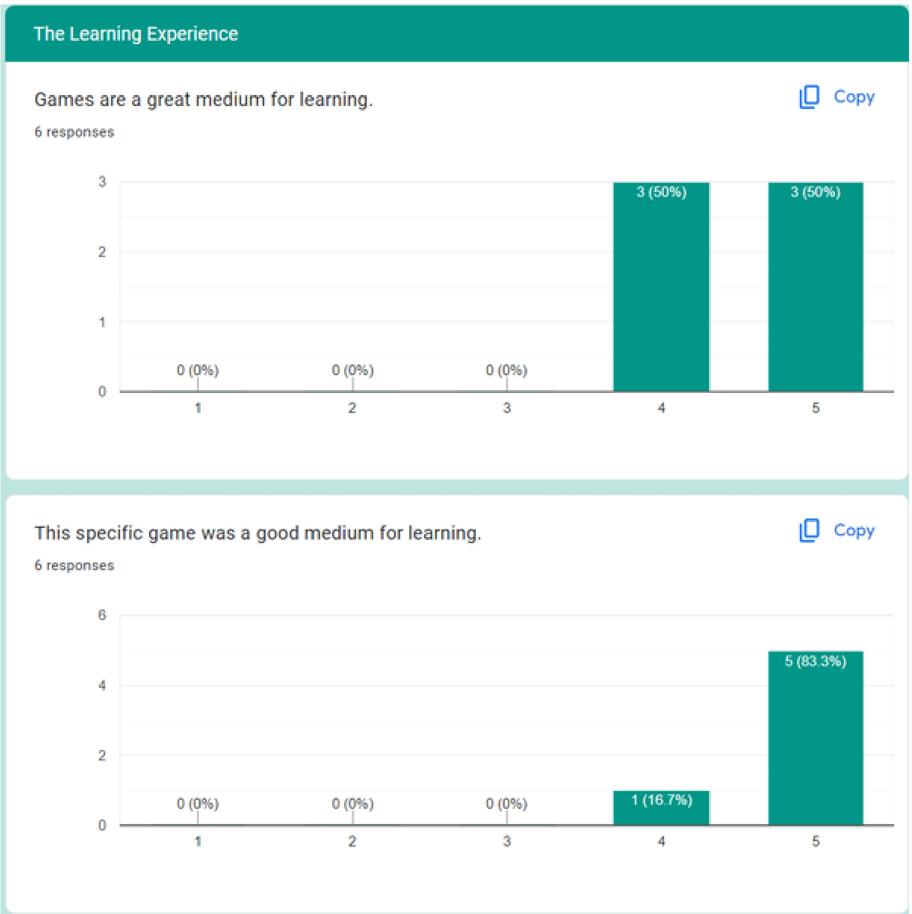


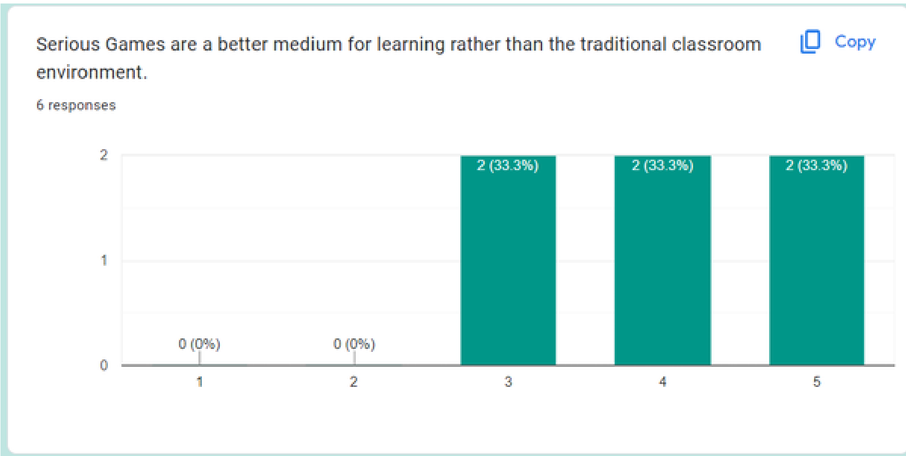

Did you manage to complete the Maze Game?
Copy
6 responses
Yes
No
100%


Did you find the controls for this game difficult?
Copy
6 responses
Yes
No
100%


How much enjoyment did you get from playing this game?
Copy
6 responses
0 (0%)
0 (0%)
0 (0%)
1 (16.7%)
1 (16.7%)
4 (66.7%)
1
2
3
4
5
6

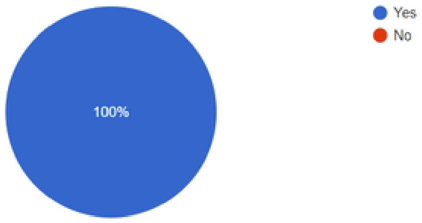

How likely is it that you would play this game again?
Copy
6 responses
0 (0%)
1 (16.7%)
0 (0%)
0 (0%)
2 (33.3%)
3 (50%)
1
2
3
4
5
6


The Learning Experience
Games are a great medium for learning.
Copy
6 responses
3
2
1
0
0 (0%)
0 (0%)
0 (0%)
3 (50%)
3 (50%)
1
2
3
4
5
This specific game was a good medium for learning.
Copy
6 responses
6
4
2
0
0 (0%)
0 (0%)
0 (0%)
1 (16.7%)
5 (83.3%)
1
2
3
4
5


Serious Games are a better medium for learning rather than the traditional classroom environment.
Copy
6 responses
2
1
0
0 (0%)
0 (0%)
2 (33.3%)
2 (33.3%)
2 (33.3%)
1
2
3
4
5

Did you perform the post-quiz review exercises?

6 responses



If yes, how did you find the exercises?

6 responses

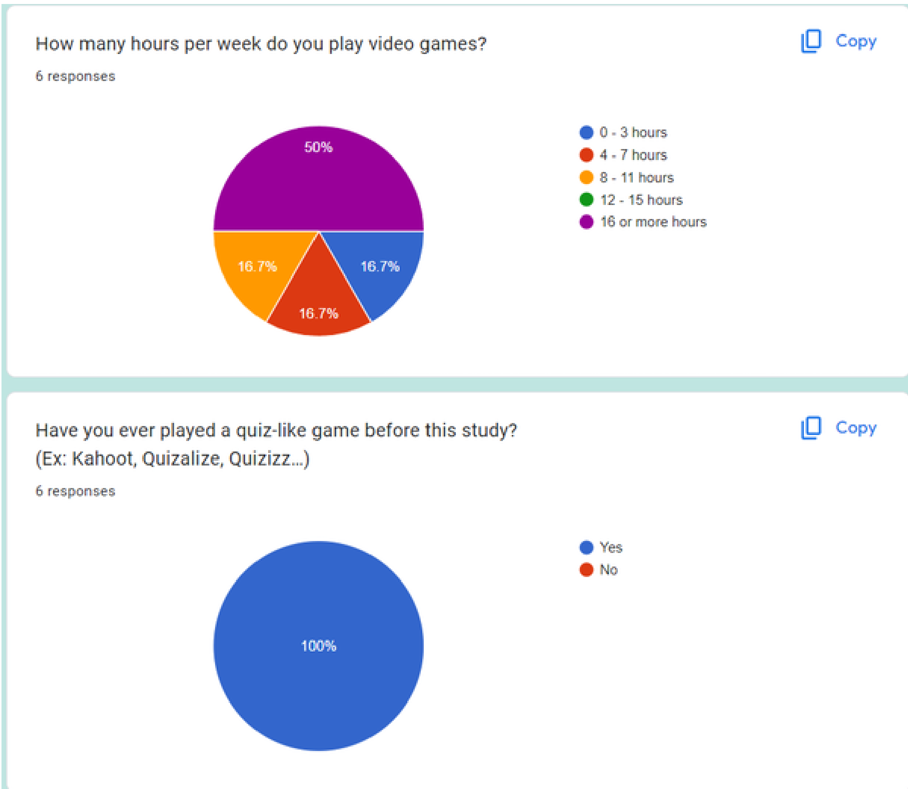
- Overall, they were good and accurate questions which increased in difficulty at times and were largely varied.
- Reasonably easy, some were challanging.
- They were a great source to figure out how I could have corrected my mistakes.
- pretty fun and good for learning
- quite nice!
- Convenient

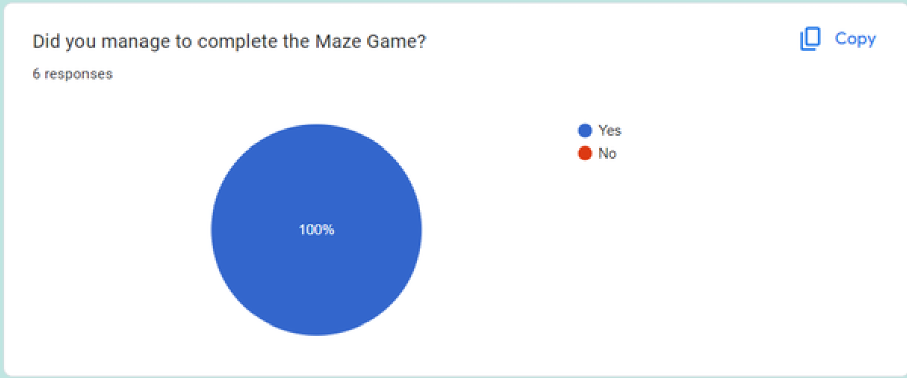
Is there any feedback you would like to give in regards to making this game better from an educational perspective?

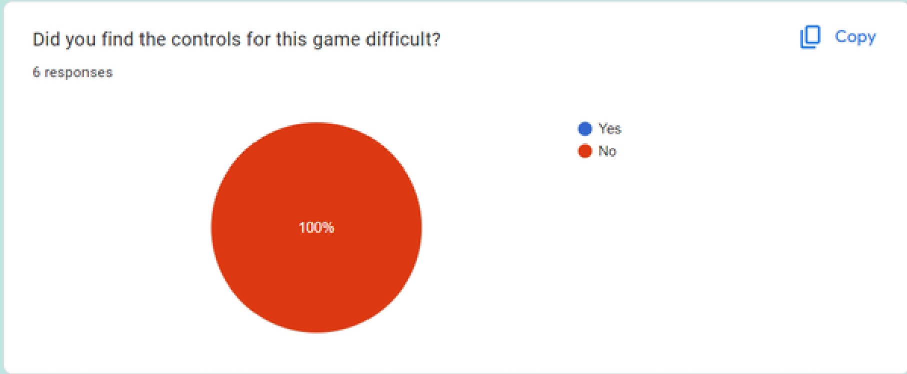
4 responses

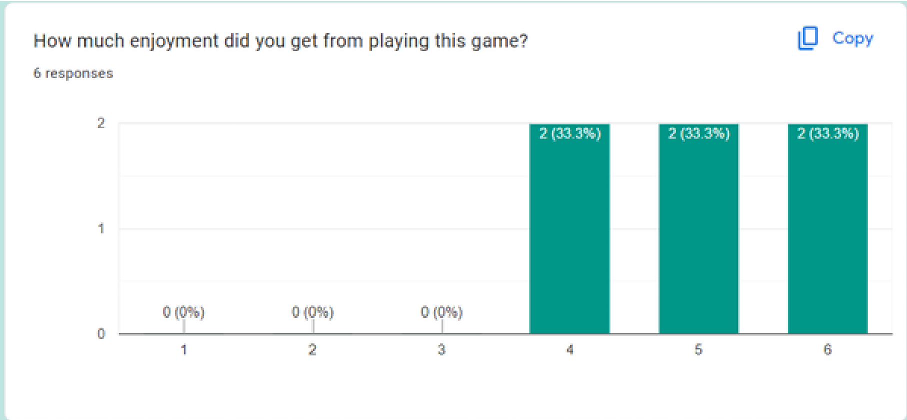
- The only issues I found were with AI. Perhaps another suggestion would be to add a pause function in the future for the Maze Game, especially since it has a timer and the player might be idle for a few minutes as they might need to take care of something urgent.
- Make other subjectspart of the game, such as english
- No feedback as of yet.
- nothing much to add on to this, great game

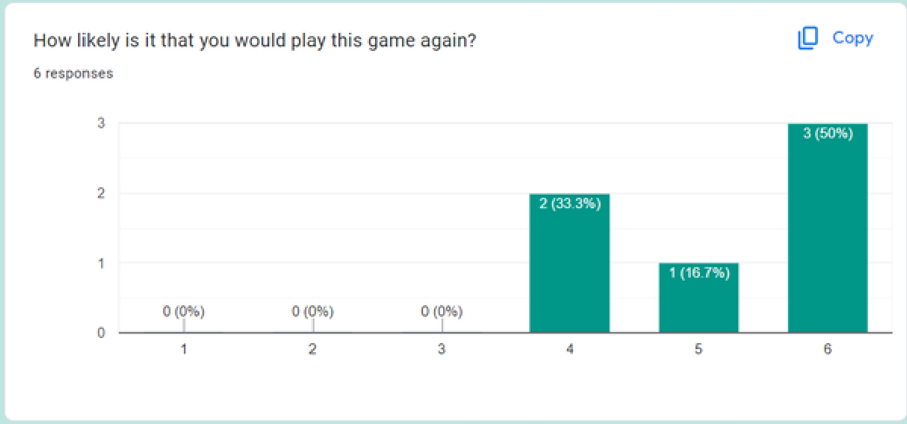
E.2.2 Non-AI Group

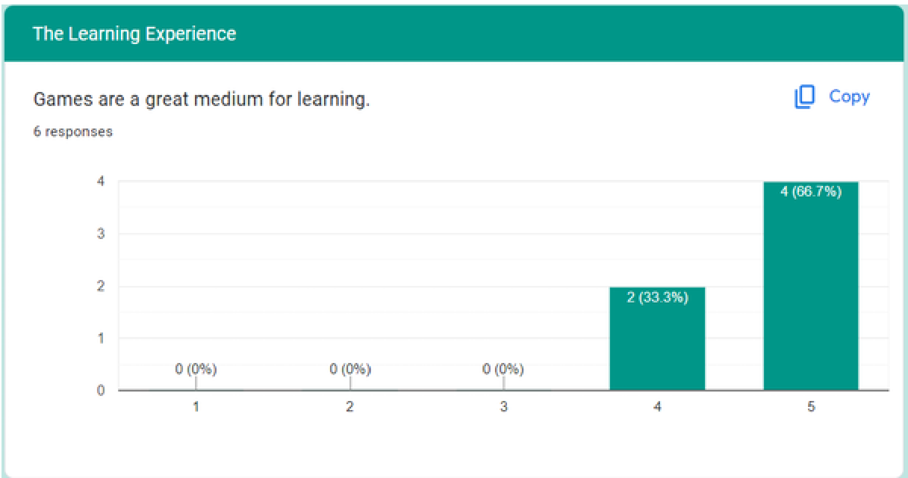


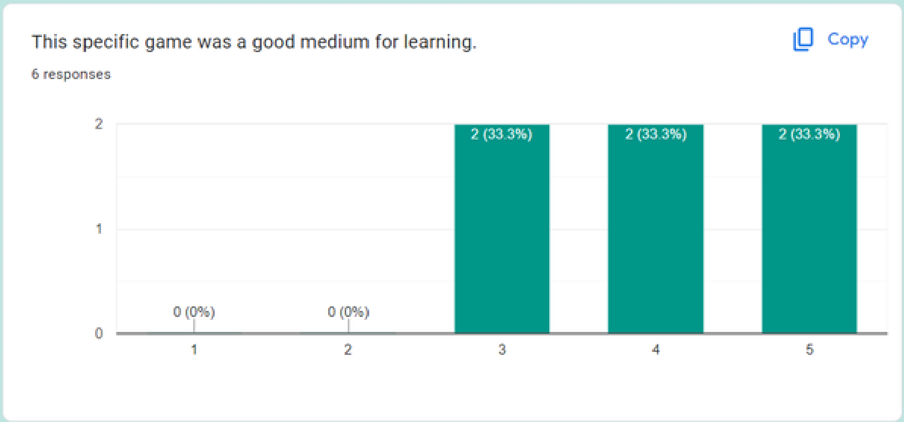

Did you manage to complete the Maze Game?
6 responses
Copy
Yes
No
100%

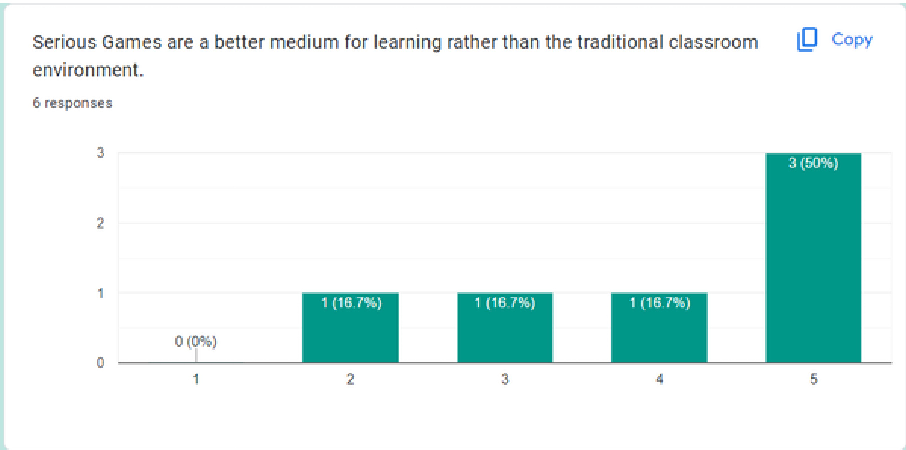

Did you find the controls for this game difficult?
6 responses
Copy
Yes
No
100%


How much enjoyment did you get from playing this game?
6 responses
Copy
0 (0%)
0 (0%)
0 (0%)
2 (33.3%)
2 (33.3%)
2 (33.3%)
0
1
2
1
2
3
4
5
6


How likely is it that you would play this game again?
6 responses
Copy
0 (0%)
0 (0%)
0 (0%)
2 (33.3%)
1 (16.7%)
3 (50%)
0
1
2
3
1
2
3
4
5
6


The Learning Experience
Games are a great medium for learning.
Copy
6 responses
4
3
2
1
0
0 (0%)
0 (0%)
0 (0%)
2 (33.3%)
4 (66.7%)
1
2
3
4
5


This specific game was a good medium for learning.
Copy
6 responses
2
1
0
0 (0%)
0 (0%)
2 (33.3%)
2 (33.3%)
2 (33.3%)
1
2
3
4
5


Serious Games are a better medium for learning rather than the traditional classroom environment.
Copy
6 responses
3
2
1
0
0 (0%)
1 (16.7%)
1 (16.7%)
1 (16.7%)
3 (50%)
1
2
3
4
5

Is there any feedback you would like to give in regards to making this game better from an educational perspective?

4 responses

- Make traversing the maze easier, like a bunny hop mechanism
- Option to have questions and potential answeres narrated to user.
- Increase time to read question/add an audio option to hear the question
- make mazes with different topics available

Appendix F Google Drive Link

The following Google Drive Link: **LINK TO DRIVE** contains a copy of this document, the serious game environment web server, as well as a unity package containing the unity development build of the game. In addition to this, there is a README explaining how to setup the web server.