UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en ___Ingenieria Informatica___

TRABAJO FIN DE GRADO

SpaceRaceEdu: Development of a Gamified Educational Tool for Self-Study and Evaluation

Leah Hadeed
Tutor: Juan Jesús Roldan Gómez
Ponente: Ivan Cantador

JUNIO 2021

SpaceRaceEdu: Development of a Gamified Educational Tool for Self-Study and Evaluation

AUTOR: Leah Hadeed
TUTOR: Juan Jesús Roldan Gómez

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2021

Resumen (castellano)

Este Trabajo Fin de Grado es una investigación en el campo de las tecnologías de la educación con respecto a la gamificación del aprendizaje y los exámenes. El software creado en esta investigación está diseñado para aumentar la participación de los estudiantes, fomentar el trabajo en equipo, mejorar la perspectiva educativa y servir como una forma psicológicamente menos invasiva de examinar el conocimiento de un estudiante frente a un examen escrito típico. Esta investigación considera las experiencias de docentes con diferentes años de experiencia en el campo educativo, así como una multitud de rangos de edad, para demostrar cuán efectiva es la educación gamificada en los niveles de primaria, secundaria y universidad.

El software desarrollado en esta tesis crea un entorno virtual limitado, inspirado en los juegos multijugador contemporáneos, en el que los alumnos trabajan en equipo para resolver las tareas planteadas por el profesor. El juego incorpora elementos de competencia limitada, así como elementos de cooperación. La competición se usa para aumenta el factor de entretenimiento de las asignaturas, mientras que la cooperación se usa para fomentar el trabajo en equipo y el buen espíritu deportivo. Las tareas antes mencionadas involucran versiones visualmente atractivas y gamificadas de preguntas de opción múltiple, soluciones matemáticas, categorización, ordenación e identificación. Estas tareas están diseñadas de tal manera que no afecten negativamente la voluntad de aprender de un estudiante, donde permiten reintentos monitoreados e ilimitados y se puede salir y regresar a aquellos problemas difíciles en un momento posterior del juego. Esto significa que los estudiantes no se desaniman a través de tácticas típicas de examen como la puntuación negativa en respuestas incorrectas, sino que se les anima, en un entorno seguro y divertido, a cometer los errores que les ayudarán a aprender.

Este software está diseñado para realizar un seguimiento de la cantidad de intentos que hace un estudiante para completar una tarea y la cantidad de tareas completadas por el estudiante, para que el profesor pueda medir la curva de aprendizaje de cada estudiante individualmente. Estas estadísticas estarán disponibles para futuras revisiones, lo que permitirá al profesor, usando su propia discreción, evaluar el desempeño de sus estudiantes.

El proceso de prueba se ha llevado a cabo a través de encuestas sobre videos de partidas limitadas. Estas pruebas iniciales han mostrado una respuesta positiva a la gamificación del temario principalmente en instituciones primarias y secundarias, y estudios de nivel universitario que requieren niveles más altos de memorización frente a experimentos de pensamiento independientes y desarrollo de ideas. La respuesta de los estudiantes también fue positiva, ya que la mayoría de los estudiantes optaron por las revisiones (si la materia lo permite) realizadas con este software debido al valor de entretenimiento y los aspectos competitivos.

Abstract (English)

This Bachelor Thesis is an investigation into the field of new technologies in education with regards to the gamification of learning and examinations. The software created in this investigation is designed to increase student participation, foment teamwork, improve educational outlook, and serve as a psychologically less invasive way to examine a student’s knowledge when compared to a typical written exam. This investigation considers the experiences of teachers with varying years of experience in the educational field as well as a multitude of age ranges, to demonstrate how effective gamified education is at primary, secondary, and university level.

The software developed in this thesis creates a limited virtual environment, inspired by contemporary multiplayer games, in which students work in teams to resolve tasks set out by the teacher. The game incorporates elements of limited competition as well as elements of cooperation. The cooperative aspects encourage teamwork and good sportsmanship while the competitive aspects increase the entertainment factor of typical class assignments. The aforementioned tasks involve visually appealing, gamified versions of multiple-choice questions, mathematical solutions, categorization, ordering and identification. These tasks are designed in such a way as to not negatively affect a student's desire to learn, wherein they allow for unlimited, monitored retries and can be exited and returned to at a later point in the game. This means that the students are not discouraged through typical examination tactics like negative scoring on incorrect responses, rather, they are encouraged in a safe, diverting environment to make the mistakes that will help them grow as problem solvers and human beings.

This software is designed to keep track of the number of tries a student takes to complete a task and the number of tasks completed by the student, as a way for the teacher to gauge the learning curve of each student individually. These statistics will be available for future revisions, allowing the teacher to, using their own discretion, evaluate the performance of their students.

The testing process was completed through a series of opinionative surveys regarding limited gamplay videos. These initial tests have shown a positive response to the gamification of subject matter mostly in primary and secondary institutions, and university level studies which require higher levels of memorization rather than independent thought experiments and idea development. The student response was also positive, in that most students opted for revisions (if the subject matter permits) done using this software because of the entertainment value and the competitive aspects.

Palabras clave (castellano)

Gamificacion, Educación, eLearning, Among Us, Multijugador, Software, Videojuego, MMO, Online, Classcraft, Kahoot, Minecraft Education Edition

Keywords (inglés)

Gamification, Education, eLearning, Among Us, Multiplayer, Software, Videogame, MMO, Online, Classcraft, Kahoot, Minecraft Education Edition

Acknowledgements

I would like to thank all the people who helped in the design phase of this project, without whom the idea would continue to be just that. Namely, my TFG advisor, Juan Jesus Roldan Gomez, who came to me with an idea for a project that touched my heart, Juan Martinez Alonso, for his help with some of the early design concepts, and Andrea Martin, for turning the design sketches into works of art.

I would also like to thank the many others who collaborated in a lesser degree to this project as beta testers and sources of information on the practical uses of such an application in schools: Khiana Mannix, Meaghan Louise Chipol, Alvaro Daimiel Martinez, Karolyn Edwards. Additionally, I would like to thank the institutions which have given me the formal training to be able to undertake such an ambitious project, Island Academy International School, and the Autonomous University of Madrid.

Finally, I would like to thank my family and friends for their constant support, and my loving boyfriend who continued to push me through the development of this TFG. My goal has always been to shoot for the stars (hence the rocket theme), and without them this project would not have left the ground.

TABLE OF CONTENTS

FIGURE INDEX

TABLE INDEX

1 Introduction

1.1 Motivation

This TFG report details the planning and progression of an investigation into the relevance and utility of *gamified*[1] learning tools when applied to the collegiate education system. This study comes on the heels of the global Covid-19 pandemic, and which emphasized the need for alternative methods of teaching and examining students in a way that promoted their interest in the subject matter and capture their attentions in environments where distraction may be present [1].

In many ways, the Covid-19 pandemic has reignited an interest in modifying the traditional educational system to reflect the needs of the modern world and increased *digitalization*[2] of society. It has also shed light on the changing views related to the subject matter, for example, how instant gratification or *micro satisfaction*[3] [2] methods of teaching (like through videogames) can lead to a higher participation rate among students as well as a higher task completion rate [3]. The official article from the Organization for Economic Co-Operation and Development (OECD) states:

> “While the educational community have made concerted efforts to maintain learning continuity during this period, children and students have had to rely more on their own resources to continue learning remotely through the Internet, television or radio. Teachers also had to adapt to new pedagogical concepts and modes of delivery of teaching, for which they may not have been trained.” (OECD, 2020, p.4)

This article makes the argument that the pandemic has simply accelerated transitions that were already taking place in the world of education [4].

One such shift was that in favour of more applicative learning styles, from visual and auditory to tactile and kinaesthetic [5]. Such changes follow a philosophical approach to education developed in the Renaissance and proposed by the philosopher John Amos in his novel Great Didactic (1627) [6] in which he states:

> “That, as far as is possible, instruction should be given through the senses, that it may be retained in the memory with less effort.” (Comenius, 1627, p.53)

This report delves into the stigmata surrounding *gamified* education through the creation and testing of such a tool, with requirements based on conversations with real life teachers. To further analyse its effectiveness, this report also compares the developed software and its results to those of similar software already used in both institutional and higher

[1] Gamification: The process of adding game-like aspects to a certain task to encourage participation.

[2] Digitalization: The process of digitizing operations and environments through the digitization of data/processes/documents

[3] Micro Satisfaction: A limited feeling of joy created by small victories, in this case, applied to a gaming environment.

educational levels of study. To limit the scope of this project, seeing as it requires the integration of many components, i.e., the gaming environment, the supporting server and API, the question database, and the teacher interface, this investigation will focus specifically on the creation of the frontend gaming environment. The end of this investigation should see the creation of a single player gaming unit with embedded connective functionality for a later conversion to multiplayer online gaming once the server technology is created. For more information on the additional aspects of the project, see the companion TFG: **SpaceRaceEdu: Desarollo de arquitectura y back-end para videojuego multijugador educativo**.

The main schools involved in the design and testing processes are the Autonomous University of Madrid (Collegiate level, Spain), the Polytechnical University of Madrid (Collegiate level, Spain), Island Academy International School (Secondary level, Antigua), and St. Joseph's School (Primary level, Gibraltar). The development process resides solely in the Autonomous University of Madrid.

1.2 Objectives

On a global scale, this project aims to offer an alternative to traditional teaching methods through the *gamification* of the evaluation and self-study process in a way that foments teamwork and communication students, as well as offering a simple continuous assessment method for teachers. Some of the challenges being confronted in this project are the stigmata associated to online learning and the perceptions among older generations of videogames and their influence on the younger generations.

On a more localized scale, this project delves into the creation of a videogame, based on criteria set by real world professors. Specifically:

- Compile opinions of various professors from different levels of experience, class sizes, disciplines, and academic teaching level, on what a gamified learning experience means to them.
- Design the basic structure of a game based on the aforementioned criteria. This includes the rules of the game, player control, graphical user interface (GUI), online communication, etc.
- Develop a functional single player prototype with a local filesystem for storing questions and backend variables (implemented separately).
- Test the prototype with the target audience and compile opinions, results, and complaints (if any).
- Evaluate the utility of such a tool.

This investigation serves as a practical approach to both videogame design and project management, as it requires communication with a backend developer and artist for the correct completion of the game.

1.3 Organization

This thesis is organized in 6 different sections, each detailing a specific part of the investigative process. The following are the headers of each section:

1. Introduction
   - Details the motives behind the investigation, intended goals and objectives, and the structure of the investigative document.

2. Current State of Affairs
    - Details the current environment, out of which the need for such an investigation arises, and the technologies that will be used for the execution of said investigation.
3. Design
    - Details the proposed design for the investigation based on the functional and non-functional requirements for the investigation, along with the statistical analysis of the proposed requirements with regards to a test population.
4. Development
    - Details the development/implementation process of the investigation.
5. Integration, tests and results
    - Details the integration of the system along with the results obtained from the testing phase of the investigation.
6. Conclusion and future works
    - Details the conclusions reached upon completion of the investigation.

Following the main 6 sections are the Bibliography and the Appendices.

This thesis makes references to certain documents required for typical game development, namely the Game Design Document, the Game Development/Structure Document, and the User Manual respectively. These can be found in Appendices A, B, and C. These documents contain more detailed information on the design and development decisions made in the development of this project, as well as the planning and distribution of external work, such as art and sound. The User Manual details the mode of operation of the constructed game. Appendix D details the raw results of the trial phase. Appendix E contains the initial design sketches for the player and map.

Note that all information pertaining to this investigation, this includes the investigation survey results, the developed software, and the example gameplay, can also be found at the following link:
https://drive.google.com/drive/folders/1D6xgVdOwgiu51CXZPsD0WLTBfD4wfwYb?usp=sharing

2 Current State of Affairs

2.1 Overview

On a global scale, this investigation aims to offer an alternative to traditional teaching methods through the gamification of the typical assessment process. Though there are many similar products on the market, this investigation is specifically designed to foment teamwork and communication, as well as act as an alternative to standard process of evaluating the learning curve of students throughout the academic year. On a more localized scale, this investigation delves into the creation of a videogame, based on criteria set by real world professors.

2.2 Online Learning

Over the past decade, many studies have been made to collect the opinions of teachers on the subject of online learning. One of these such opinions is that technology is stripping children of their attention spans [7]. The reason for this being that the screens, bright lights, and entertainment factor prove too distracting for the children to remain on task. Additionally, the faster pace of the information flow makes studying in the traditional sense more difficult because of the lack thereof of such pace. In the article, Graber D., a 7th grade teacher, cites studies made previously into this relationship between memory/retention and technology where the results have proven the negative correlation between the two, however, she follows this information with additional studies into the contrary, noting that the difference between them is the generation of students being affected. She states that younger generations are more readily adapted to this sort of environment, and that the integration of education and technology can offer new learning experiences to better prepare the children of today for the world of tomorrow, including but not limited to, how to balance time spent online and offline, how to evaluate the information they find online, etc.

Furthermore, recent studies show teachers have grown accustomed to the encroachment of technology on their classrooms, and even welcome it, thanks to some of the readily available software on the market [8]. Lynch M., an advocate for technology in schools, claims students participate more actively in subject matter when they are engaged in the way it is being taught. He offers a list of software already on the market to achieve such participation. His ideas go hand in hand with those of Huang W.H.Y. and Soman D., who, in their article state that the *gamification* of education does not directly affect the knowledge gained by the students, rather it increases their commitment and motivation, thereby indirectly helping to improve their knowledge and skills in the given area [9]. Their ideas on what elements to take into consideration when *gamifying* education are expanded upon later in this document. Kiryakova, G., Angelova, N., & Yordanova expand upon this even further, citing that the reason for this is the fact that learners nowadays are growing up with this technology and are more readily adapted to its integration into their daily lives [10]. As a result, teachers have had to adapt their teaching methods to keep up with the times. Even back in 2011, an article from *The Futurist* [11] states:

> “Some suggest taking Web-based learning one step further: Online social gaming may become the educational tool of choice.” (Cohen, 2011, p.1)

Despite the comments made by the article above suggesting that advancements in gamified learning were already taking place, not all movement towards this change was organic.

As stated previously, the Covid-19 pandemic also had a notable impact on the way society views the use of technology in classrooms and online learning in general, considering that students could not attend formal lessons and that teachers needed to find a way to continue to impart their knowledge without the comforts and familiarities of a traditional classroom. Some resorted to simple tools such as Microsoft Teams, Skype, and Zoom, for an experience similar to ‘face-to-face’ learning [12], while others took the plunge directly into more gamified software.

An investigation into the effect of the Covid-19 pandemic on education in Bhutan found that because of the pandemic and the inability to conduct ‘face-to-face’ learning, the need for online learning surged dramatically [12]. This article specifically states:

> “Several schools, colleges and universities have discontinued face-to-face teaching. There is a pressing need to innovate and implement alternative educational and assessment strategies.” (Pokhrel & Reshan, 2021, p.1)

And refers to another study done in India regarding the same topic, which further supports this viewpoint, adding [13]:

> “The COVID-19 pandemic has provided us with an opportunity to pave the way for introducing digital learning.” (Dhawan, 2020, p.1)

Both articles reference the software available for online learning and its impact on society. Though gamified educational software is not specifically referenced, other articles detail their expanded growth during the pandemic and the effectiveness of such software. A study from the Department of Industrial Engineering, Kangwon National University [14] found that:

> “The use of gamification provides gameful experiences through which learning experiences are reinforced. Furthermore, such experiences have a positive effect on learner motivation.” (Park & Kim, 2021, p.1)

The following subsections will go into more depth on examples of gamified learning software and its use cases.

2.3 Similar Platforms

This section delves into the already existent software in the field of gamified education. Some of the aspects to an educational game being referenced are:

- Platform: Device on which the software runs.
- Play Style: The number of players it would take to play (Individual/Team).
- Game Style: The flow of the game (Quiz, Experimentation, Puzzle Solving, VR Immersive).
- Duration: The span of time used to complete an execution of the software (short-term/long-term).

Name	Classcraft	Kahoot	Breakout EDU	Minecraft: Education Edition
Platform	PC	PC/Android/iOS	PC	PC/Android/iOS
Play Style	Individual	Individual (*)	Individual (*)	Individual (**)
Game Style	N/a (***)	Quiz	Puzzle	VR Immersive
Duration	Long-Term	Short-Term	Short-Term	Long-Term

Table 1: Comparison of Similar Software

(*) These games were designed with the intention of individual gameplay. However, the software can be used in groups at the discretion of the teacher.

(**) These games were designed with the intention of individual gameplay. However, being VR Immersive, the individual players can interact with one another through dialogue interfaces and collaborative actions.

(***) This software is designed not as a game but rather as an additional teaching tool and reward system for an in-class learning experience.

2.3.1 Classcraft

Classcraft is a hybrid learning platform that allows students to create an avatar and work towards teacher-set learning goals in a gamified format. Students are placed in teams to foment communication and teamwork. The motivation behind Classcraft is to provide real-world skills and rewards to learning in a way that makes the process enjoyable. The actions of the students in real life are what earn them rewards defined by the teacher, i.e., hints on tests, extra days for handing in assignments, etc. In the same way, the rewards can also be punishments if the student behaviour goes against that which the teacher is trying to foment, i.e., less days for handing in assignments.

Figure 1: Classcraft Logo [15]

The approach used by Classcraft can be summed up by the following statement found on the main website [15]:

> “Classcraft's motivational approach is based firmly on Self-Determination Theory, a well-researched psychological framework to explain what drives individuals to engage in specific activities. By fulfilling players' needs for exercising control, developing competency, and experiencing relatedness, games are effective external systems to foster intrinsic motivation. In other words, games are great at making you want to play them.” (Classcraft Studios Inc., 2021)

This approach allows the teachers to continue with their current method of teaching, but also integrate a system of rewards and consequences to the learning environment as though in a game. The software being developed in this investigation takes inspiration from the team aspects of Classcraft.

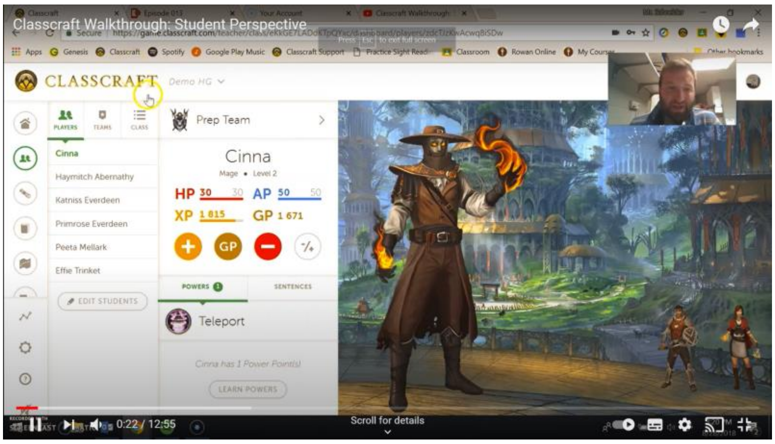


Figure 2: Example Gameplay Classcraft [16]

2.3.2 Kahoot

Kahoot is an online quiz application that allows teachers to create multiple choice styled quizzes and share them in class with their students to revise or test their knowledge on material seen in class.

Figure 3: Kahoot! logo [17]

The approach used by Kahoot! can be summed up by the following statement found on their website [17]:

> "Kahoot! is a global learning platform company that wants to empower everyone, including children, students, and employees, to unlock their full learning potential. Our learning platform makes it easy for any individual or corporation to create, share, and play learning games that drive compelling engagement. Kahoot! games can be played anywhere, in person or virtually, using any device with an internet connection." (Kahoot, 2021)

This approach allows teachers to implement typical multiple-choice tests on different digital devices, which permits students to actively participate from distinct locations, using passcodes to access the testing environment. Kahoot acts as revision software as well as examination software, similar to the software being developed in this investigation, however Kahoot remains simplistic in its visual implementation, and does not include the entertainment factor (avatars, map exploration) that this software intends to provide. One of the positive aspects of Kahoot that this software aims to imitate is the ability to include external objects such as pictures (and eventually audio) into the gameplay, (see Fig.4).

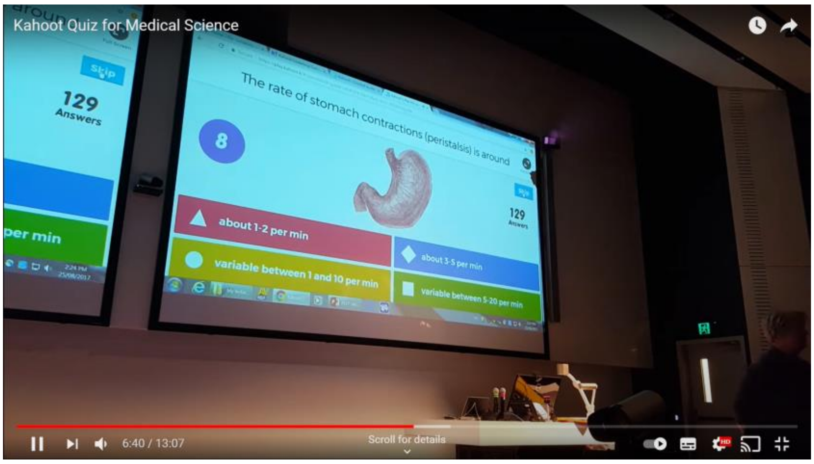

Kahoot Quiz for Medical Science
The rate of stomach contractions (peristalsis) is around
8
129
Answers
Skip
about 1-2 per min
about 3-5 per min
variable between 1 and 10 per min
variable between 5-20 per min
per min
6:40 / 13:07
Scroll for details

Figure 4: Example Gameplay Kahoot [18]

2.3.3 Breakout EDU

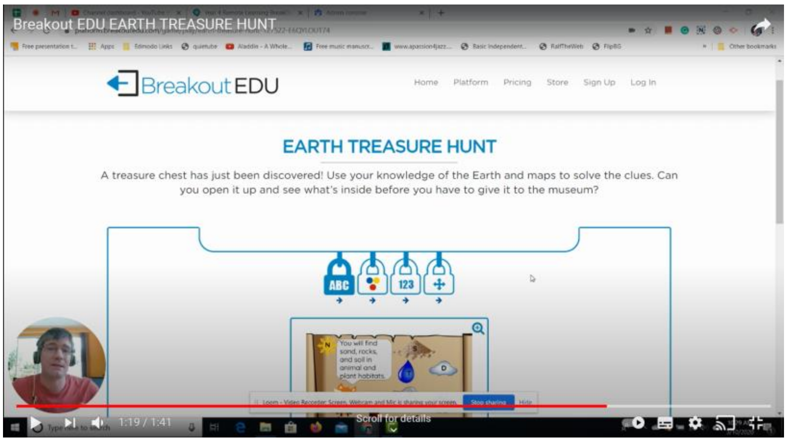
Breakout EDU is an online classroom environment with different puzzles that the students can solve as part of their school curriculum. The teachers can make use of the 'Design a game' functionality of the software to create minigames specifically designed to meet their classroom needs.

Figure 5: BreakoutEDU Logo [19]

The approach used by BreakoutEDU can be summed up by the following statement found on their website [19]:

> "Breakout games for both remote and in-person learning where students work collaboratively to solve puzzles." (BreakoutEDU, 2021)

Like Kahoot, Breakout EDU also contains test-based questions that serve for revision and examination. The questions use more involved visuals, Additionally, like Classcraft, Breakout EDU allows users to play on teams to complete the tasks.


Breakout EDU EARTH TREASURE HUNT

EARTH TREASURE HUNT

A treasure chest has just been discovered! Use your knowledge of the Earth and maps to solve the clues. Can you open it up and see what's inside before you have to give it to the museum?

You will find sand, rocks, and soil in animal and plant habitats.

Figure 6: Example Gameplay Breakout EDU [20]

2.3.4 Minecraft: Education Edition

Minecraft is a fantasy online virtual world that players can manipulate to their liking. The educational edition of Minecraft adds classroom styled elements to the virtual world based on set curricula but that can be modified to suit the needs of the teacher. Students are free to create their own avatars and explore the worlds created to suit their learning needs. Examples of the possible uses of MEE are: exploration of fictional worlds from recommended literatures, exploration of historical places, visualization of DNA strands and molecules, experimentation with elements based on their theoretical properties, etc.

Figure 7: Minecraft Education Edition Logo [21]

The approach used by Minecraft Education Edition can be summed up by the following statement [21]:

> “Minecraft: Education Edition empowers unique and creative learning experiences for educators and students alike.” (Minecraft: EE, 2021)

Minecraft Education Edition is the most unique of the games mentioned considering the software was initially developed without educational purposes in mind and was later adapted to suit a learning environment. Minecraft Education Edition also takes on the principles of *avatarization* and virtual reality, allowing students to create their own avatars and explore an interactive world with tasks similar to the ones being developed in this investigation.

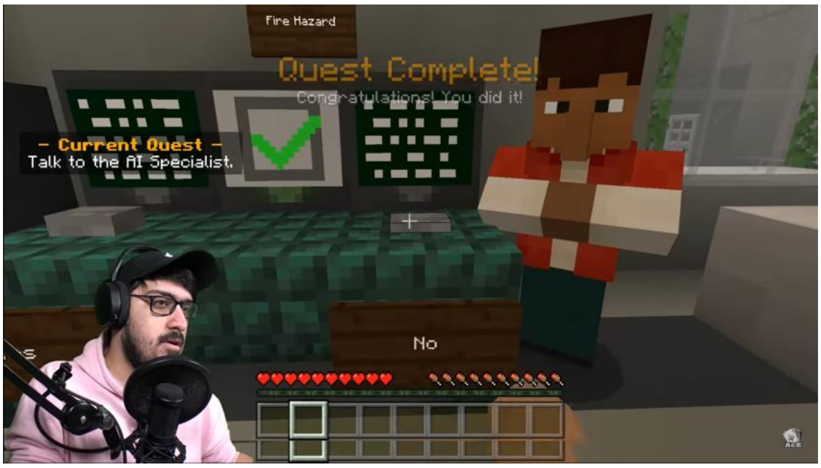


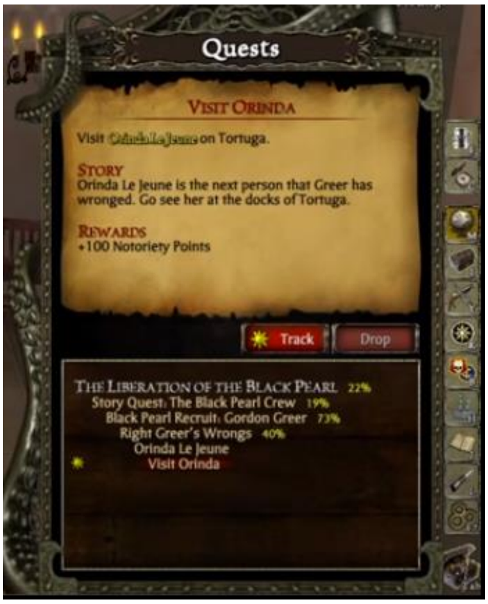
Figure 8: Example Gameplay Minecraft Education Edition [22]

2.4 Other Game Inspirations

Since the idea behind this software is to provide a captivating gaming experience, part of the research process entailed the discovery of universally appealing games with similar functionality to act as a starting point for the development process. As such, the games listed in this section are inspirations in both design aesthetic and functionality. They are multiplayer online games that include a quest/task-based gameplay in which the users can work together or go head-to-head in internal competitions.

2.4.1 Pirates of the Caribbean Online

Pirates of the Caribbean online was a Massively Multiplayer Online (MMO) videogame that appeared in the wake of the success of the original Pirates of the Caribbean film. The objective of the game was to explore the virtual world and complete a series of predetermined tasks to level up your avatar. These tasks ranged from speaking to an NPC, visiting a different isle, to battling against enemy NPCs, and were focused on advancing a predetermined plot. Once completed, these tasks earned a player gold and XP, which they could use to upgrade their avatar, purchase new items to enhance gameplay, etc.



Quests

VISIT ORINDA

Visit Orinda Le Jeune on Tortuga.

STORY
Orinda Le Jeune is the next person that Greer has wronged. Go see her at the docks of Tortuga.

REWARDS
+100 Notoriety Points

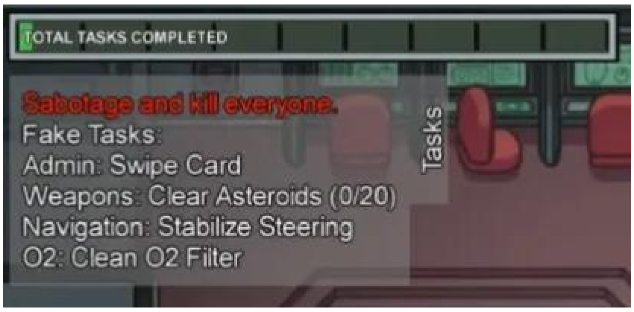
THE LIBERATION OF THE BLACK PEARL 22%
Story Quest: The Black Pearl Crew 19%
Black Pearl Recruit: Gordon Greer 73%
Right Greer's Wrongs 40%
Orinda Le Jeune
Visit Orinda

Figure 9: Pirates of the Caribbean Online Player Quest Board [23]

This project takes inspiration from the quest structure of this game (as shown above), and the way that the quests are used to give a deeper understanding into the life of pirates, which, in this case, translates to a deeper understanding of the subject matter.

2.4.2 Among Us

Among Us is a lightweight mobile/PC game that serves as the main inspiration for this project because of its versatility and accessible multiplayer gaming style, as well as its iconic status in contemporary gaming. Among Us is another example of a multiplayer online game with user specific tasks, but in this case, the players work together to discover an impostor on board a spaceship. Unlike the previous game, the tasks in this game are tracked on a global scale and are designed as a series of mini games that serve to occupy certain users while they discover who the imposter is.



Sabotage and kill everyone.
Fake Tasks:
Admin: Swipe Card
Weapons: Clear Asteroids (0/20)
Navigation: Stabilize Steering
O2: Clean O2 Filter

Figure 10: Among Us Player Task Board and Global Task Tracker

This project takes inspiration from the task structure and tracking mechanisms, and the concept of teamwork from this game. It also pays homage to the game's iconic cartoonish graphics to attract a larger audience. The tasks in Among Us are repetitive and serve no educational value, but they serve as a baseline for the mechanisms and logic used in this project's software.

3 Design

The design process considers research from articles about gamified education as well as opinions on what gamified education could look like in a modern classroom from a handful of professors from around the world. Some of the points to take into consideration are the progression of the game being developed, as it must engage the students without proving too distracting for the classroom, the ability to experiment, allowing students to feel free to try without the repercussions of failure (which could discourage the same participation it is trying to foment) and rapid feedback, which also helps with engagement [24].

3.1 Requirements

This section describes the requirements of the investigation and how they relate to the design in the following section. These requirements have been compiled through a series of surveys and meetings with professors from educational institutions who would like to implement this style of evaluation in their own classrooms. While some reunions were done in person, most of the interactions were done through services like Microsoft Teams, for holding reunions, surveymaker, for general opinions and to access a broader audience, and email for any basic questions about the system.

The following table shows the results of an opinionative survey given to different teachers from around the globe regarding their current teaching environment and their opinions on what role gamified education plays in their classrooms and contemporary life. This survey acts as an idea support bank for creative and design-related aspects of the investigation software, rather than as a sample with scientific merits, considering it does not consider a significant enough portion of the population. Nor does it mean to generalize gamified learning based on a diminutive sample size.

Table 2: Teacher Opinion Survey Results

Experience (YRS)	Class Size	Ususal Teaching Tools	Usual Examination Methods	Student Productivity (0-10)	Student Attention Span (0-10)	Student Grade Average (0-10)	What are the 5 most important facets (in your opinion) of a modern day education?	Give us your thoughts on what videogame-styled learning looks like to you.
0-2 years	10-20	Physical Labs/Hands on work, In class debates/Group learning activities	Group projects, Kahoot/Classcraft/Similar, Other	7	8	8	Class participation, Team building exercises/Group projects, Discussion-based activities, Interactive classrooms (Technology), Student progress	A videogame where the students/participants must answer a series of questions to make progress, similar to a trivial game.
0-2 years	10-20	Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, Documentaries, In class debates/Group learning activities	Formal written exam	8	7	8	Team building exercises/Group projects, Discussion-based activities, Interactive classrooms (Technology), Expert groups, Continuous evaluations	It would be an interest tactic to implement. An extension of using play in learning

0-2 years	0-10	Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, Documentaries, In class debates/Group learning activities	Essays/Reports, Group projects, Individual projects, Kahoot/Classcraft/Similar	7	8	8	Team building exercises/Group projects, Hands on work/Labs, Discussion-based activities, Expert groups, Student progress	it depends on what type of video games
10+ years	20-40	Whiteboard/Chalkboard	Essays/Reports	6	5	8	Individual study, Team building exercises/Group projects, Discussion-based activities, Interactive classrooms (Technology), Student progress	Interactive digital learning, videos, educational games or discussions
10+ years	10-20	Powerpoint/Other presentation software, Textbooks/Other book work, Documentaries, In class debates/Group learning activities	Formal written exam, Essays/Reports, Group projects, Individual projects	8	8	8	Class participation, Individual study, Discussion-based activities, Student progress	This method could Engage all students; but teachers would need to be comfortable with all facets of the technology
2-5 years	20-40	Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Other	Formal written exam, Individual projects	9	7	8	Team building exercises/Group projects, Hands on work/Labs, Discussion-based activities, Interactive classrooms (Technology), Student progress	It is an interactive way of learning, using fun techniques that most students are now familiar with and feel comfortable with. Students tend to enjoy learning in a fun way.
10+ years	10-20	Powerpoint/Other presentation software, Textbooks/Other book work	Formal written exam, Individual projects, Multiple choice tests	7	6	8	Class participation, Individual study, Discussion-based activities, Interactive classrooms (Technology), Student progress	Really engaging for the students
0-2 years	10-20	Whiteboard/Chalkboard, Powerpoint/Other presentation software	Multiple choice tests	8	6	7	Discussion-based activities, Student progress	I think it would be good for some students but it wouldn’t work for most kids. I feel like it can be distracting
0-2 years	20-40	Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, In class debates/Group learning activities	Formal written exam, Essays/Reports, Multiple choice tests	7	7	6	Team building exercises/Group projects, Hands on work/Labs, Discussion-based activities, Interactive classrooms (Technology), Student progress	I have no idea, i guess like among us? Would love to find out more about this!
2-5 years	10-20	Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, Documentaries, In class debates/Group learning activities	Essays/Reports, Group projects, Individual projects, Multiple choice tests, Kahoot/Classcraft/Similar	5	7	8	Class participation, Team building exercises/Group projects, Hands on work/Labs, Discussion-based activities, Interactive classrooms (Technology)	Playing games to learn a subject

An important aspect of this investigation was the obtention of real world, in-class experiences with regards to gamified learning to further specify the requirements and adhere to the necessities of real-life teachers. The following graphics show the results of a survey given to different teachers from around the globe regarding their current teaching environment and their opinions on what role gamified education plays in their classrooms and contemporary life.

The questions are shown in the upper left-hand corner and the possible responses are shown in the coloured legend to the right of the pie chart.

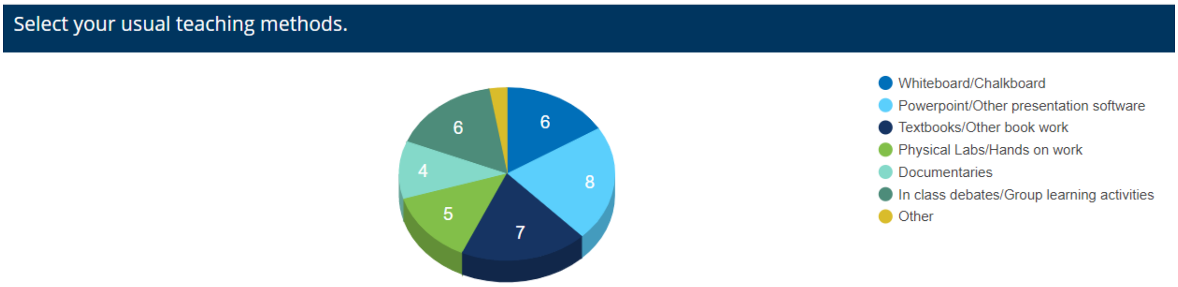


Figure 11: Survey Results - Usual Teaching Methods

This first question was aimed at discovering which teaching methods are currently implemented in classrooms around the world and to what extent. The most common methods of teaching continue to be presentations/slideshows, followed closely by textbook studies. Neither results surprising as they follow trends established well into the 1600s of accompanying text with picture/visuals to touch on multiple senses during the learning process. Surprisingly, the use of documentaries and instructional videos has dropped since their inception in 1910, when they were primed to be the learning technology of the future.

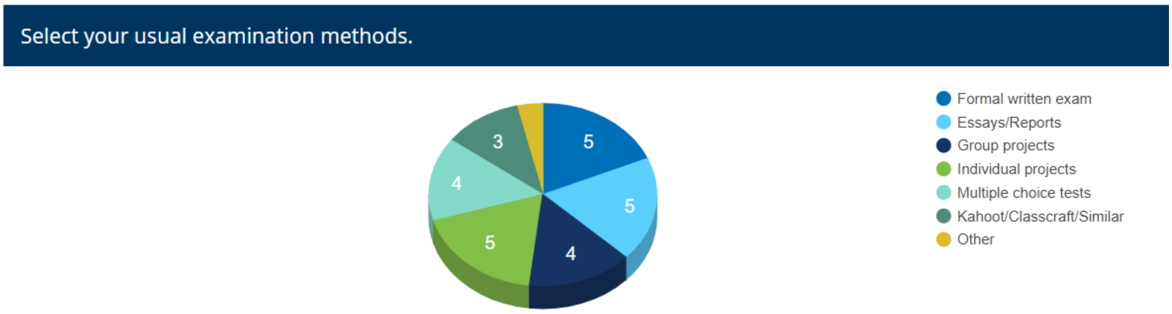


Figure 12: Survey Results - Usual Examination Methods

The second question aligns itself with the intentions of this investigation, focusing more on the ‘how’ of the examination process and the testing of a students knowledge. Again, the results were not surprising as they continue to follow trends established by the formal education system. What is interesting however is the appearance of digital examination software in a comparatively notable capacity. This shows that teachers are becoming more interested in these software and recognize its utility in the classroom.

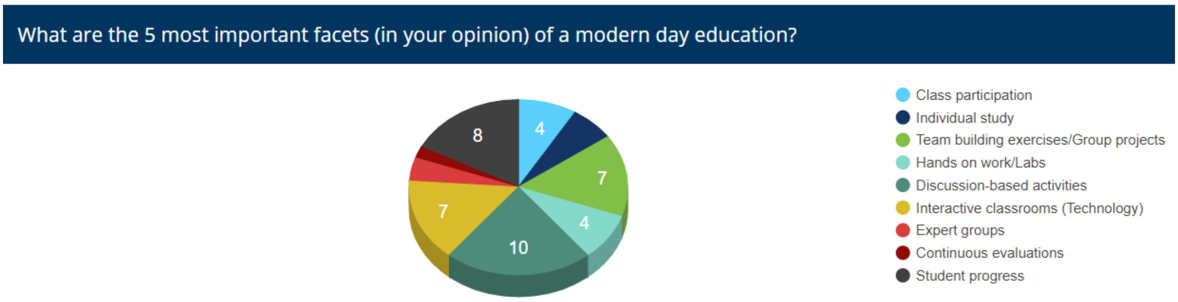

What are the 5 most important facets (in your opinion) of a modern day education?

Figure 13: Survey Results - Opinions on Modern Education

The third question focuses on the important aspects of a modern education as defined by teachers. Discussion-based activities leading the chart, going hand in hand with the third place ties of interactive classrooms and team building/group projects, demonstrates the emphasis being placed on sharing and communicating as a consortium rather than the individualistic approach (despite coming in second). As such, the software developed in this investigation should provide a mixture of group/team related aspects as well as individualistic components.

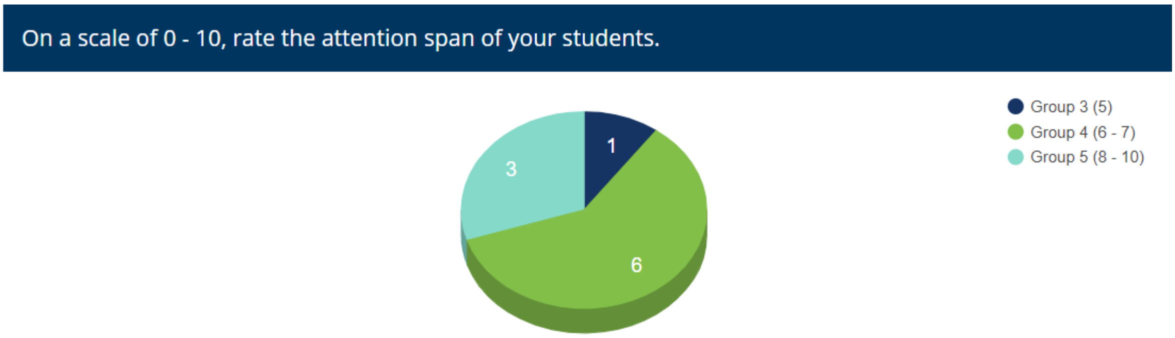
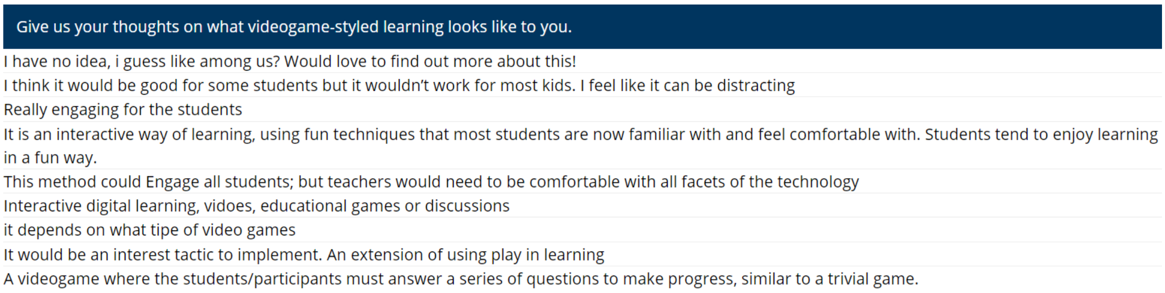

On a scale of 0 - 10, rate the attention span of your students.

Figure 14: Survey Results - Student Attention Spans

Another important aspect of the educational system where teachers sometimes require assistance is in participation/attention of the students in the class. This question was aimed at discovering how attentive teachers rate their students, as this investigation delves into a tool that aims at promoting such qualities in students. Interestingly enough, the majority of teachers consider that the attentiveness of their students is average, yet one of the main concerns (see Table 3) of the teachers is how to further engage their students in class.



Give us your thoughts on what videogame-styled learning looks like to you.
I have no idea, i guess like among us? Would love to find out more about this!
I think it would be good for some students but it wouldn't work for most kids. I feel like it can be distracting
Really engaging for the students
It is an interactive way of learning, using fun techniques that most students are now familiar with and feel comfortable with. Students tend to enjoy learning in a fun way.
This method could Engage all students; but teachers would need to be comfortable with all facets of the technology
Interactive digital learning, vidoes, educational games or discussions
it depends on what tipe of video games
It would be an interest tactic to implement. An extension of using play in learning
A videogame where the students/participants must answer a series of questions to make progress, similar to a trivial game.

Table 3: Results of Survey on Opinions of Videogame-styled Learning

The final question has to do with the ideas that professors have of what a videogame styled learning environment could look like. Most of the responses tend to focus on how they would affect the overall outlook on education rather than how they could be applied to the classroom. Most teachers appear to be in agreement that videogame styled learning could promote classroom participation, but are still apprehensive on the implementation of such tools.

The following section relates the functional and non-functional requirements of the software to be implemented in this investigation based on the results of the survey above and individual conversations with professors from the Autonomous University of Madrid and the Polytechnical University of Madrid.

3.1.1 Functional Requirements

The definition of functional requirements is "Any requirement which specifies *what* the system should do." [25] (Eriksson, 2021). This section describes the game specific functional requirements involved in the creation of this project.

Multiplayer Mode

FRMM-01 Simultaneous play for 2 – 8 players per sala at a time.
FRMM-02 Creation of different teams of players and assign a different sala to each for the match.
FRMM-03 Simultaneous reproductions of the main game map per team.
FRMM-04 The teams should have their completed and pending tasks tracked to gage the progression of the game.
FRMM-05 Only members of the same team should see each other represented on the map.

Players

FRPL-01 Players must move through the 2D map space both horizontally and vertically.
FRPL-02 Identification (name or ID) of each player must be visible to the other players at all times.
FRPL-03 Players must be able to interact with the objects throughout the 2D map space to be able to complete the assigned tasks.
FRPL-04 Each player has a random set of tasks.
FRPL-05 Players must have a record of their completed and pending tasks, and the number of tries taken to complete each task.
FRPL-06 The players must have a point of reference for globally completed tasks.

Tasks

FRTS-01 The tasks are represented on the map as interactable objects.
FRTS-02 Only the tasks assigned to the player should appear as interactable on that player's map.
FRTS-03 Only the tasks assigned to the player should appear on their task board.
FRTS-04 The tasks should take the form of interactive mini games.
FRTS-05 The tasks should come in a variety of visually appealing forms to avoid repetition.
FRTS-06 Each task has a question and a corresponding answer.
FRTS-07 There can be one or more correct answers.

FRTS-08 The tasks should reflect the environment/interactable that they are associated with.

FRTS-09 There must be a global registry of tasks, completed tasks, and pending tasks.

FRTS-10 These tasks must also be divided into teams and players.

FRTS-11 Once the task is completed, it should be removed from the player’s pending tasks list and the registry of this task should be added to the global completion of the team’s tasks.

FRTS-12 Each task should allow for retries of the same task.

FRTS-13 Each task should be able to be exited and returned to if not completed.

FRTS-14 Once a task is retried, a try count should be updated as part of user statistics (See FRGM-3).

Game Master

FRGM-01 The teacher must be able to create new games from within the game.

FRGM-02 The teacher needs an environment to be able to input new questions and their corresponding answers (question bank).

FRGM-03 The game should offer statistics on each match and each player.

FRGM-04 The game’s statistics should be tracked for the teacher to gauge the learning curve of the students (See FRGM-3).

FRGM-05 The game master should be able to define access to the game.

Match

FRMA-01 The match should be timed and run for no more than 30 minutes.

FRMA-02 Each match should have a reference number to locate statistics of the execution (See FRGM-5).

FRMA-03 The match does not require in-game communication, rather, it should allow for in-person communication or permit communication through other applications if the players are not located in the same physical area.

3.1.2 Non-Functional Requirements

The definition of a non-functional requirement is: “Any Requirement That Specifies How The System Performs A Certain Function.” [25] (Eriksson, 2021) This section outlines the non-functional requirements of the software pertaining exclusively to the game development and not the data collection or server requirements.

Graphics

NFGR-01 The graphics should be visually appealing and simple to encourage participation and further promote the concept of gamified learning.

NFGR-02 The game colours should be visually inoffensive and not representative of things that would otherwise be considered offensive to certain groups, regions, religions, etc.

NFGR-03 The characters should be simple and not representative of persons living or otherwise. However, they should be appealing enough to promote the sensation of *avatarization*.

NFGR-04 The characters may take inspiration from other visually appealing characters; however they should not be direct copies of pre-existing or copyrighted characters.

NFGR-05 The tasks visually represented as objects on the map should be more attractive/eye-catching than the background.

NFGR-06 The players should have a visual list of the tasks they need to complete (task board).

NFGR-07 The player’s Head Up Display (HUD) should show the progress made by their team and the global game progress through an animated progress bar.

NFGR-08 The tasks visually represented as objects on the map should be highlighted when a player with the same task ID in their task list enters their vicinity.

NFGR-09 The task scenes should be interactable screens with UI elements such as buttons, text boxes, and panels.

NFGR-10 The task scenes should have buttons that allow a player to return to the main scene regardless of whether or not they have completed the task.

NFGR-11 The buttons in the task scenes should all be clearly labelled.

NFGR-12 The text in the task scenes should be clearly readable, i.e. not with visually confusing fonts or colours that do not appear against their backgrounds.

NFGR-13 When a button is pressed, the pressed button animation should appear.

NFGR-14 For the scenes that require multiple responses, the buttons should clearly mark the different selected responses either by using a different colour or maintaining the pressed image.

NFGR-15 Buttons that serve the same purpose in the different tasks should have uniform graphics throughout the game (i.e. retry, return, check answer) so that the player knows what he or she is looking for.

NFGR-16 Correctly completed tasks should be highlighted as correct through some form of green visual signalling.

NFGR-17 Incorrectly completed tasks should be highlighted as incorrect through some form of red visual signalling.

NFGR-18 The tasks should allow for multiple forms of task representation (i.e. question/answer in the form of images and text).

NFGR-19 Completed tasks should be represented on the player’s task board in green font, while incomplete or incorrect tasks should be represented in red font.

Program Weight

NFPW-01 The complete game should be lightweight to be uploaded and stored on the corresponding server.

NFPW-02 The program should be built with the settings of a Web App or a standalone program.

Usability

NFUR-01 The game should use keyboard inputs for player movement and HUD interactions.

NFUR-02 The game should use mouse inputs for object interaction and task completion.

NFUR-03 Tasks should be separated into two categories, drag and drop, and button related, and should register the appropriate inputs.

NFUR-04 The drag and drop tasks should register a sustained click and maintain the object under the pointer.

NFUR-05 The button related tasks should register the individual clicks and sustained clicks as the same.

NFUR-06 The multiple-choice task (subsection of button related tasks) should allow for multiple answers. This should be made visually apparent by disabling the buttons when pressed.

NFUR-07 When a multiple-choice question is incorrect or retried, the retry button should reset the enabled/disabled status of all the buttons.

NFUR-08 The calculator tasks (subsection of button related tasks) should register up to five different button clicks.

NFUR-09 The drag and drop tasks should permit the user to drag the object to the nearest edge of the desired slot and snap the object into place automatically.

3.1.3 Diagrams

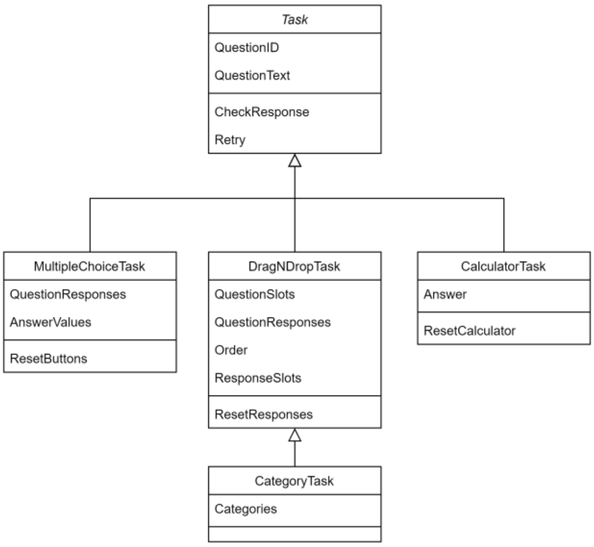


Figure 15: Class Diagram (Tasks)

Figure 15 shows the way that all the task scenes are related. Each task contains a QuestionID (int), which will be used to track the tasks done by the users, and a QuestionText (string), which could also be in image format, and will be the base of the task. The tasks then split into different categories depending on the type of response that is required. The tasks are described in detail below.

MultipleChoiceTask: allows for a single selection or multiple selection of responses. These responses have two aspects to them an identifying text and a boolean marker, which is used to check the given answer.

DragNDropTasks: This class of task is further separated into distinct types of drag and drop questions. The first is a simple ordering of responses, which are tracked by an integer placement. The second allows for the ordering responses by class, where the correction is based on a Boolean value per class.

CalculatorTask: The final task is a simple mathematic question which compares a 5-digit integer inputted by the student with the integer given as the correct response.

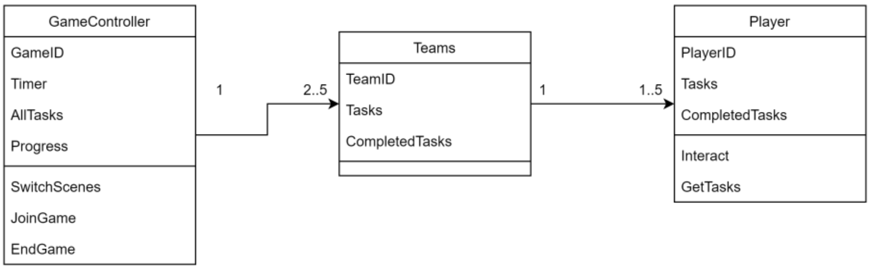


Figure 16: Class Diagram (Player Structure)

Figure 16 shows the relation between players in teams and the main game environment. In a single sala, the intended maximum number of teams is five, and on each team, the intended maximum number of players if five. This is dependent on the latency of the server that will be used to support the runtime. Each game will be timed and contain a tracker for the global number of completed tasks. Each team will have an ID and a list of tasks, as well as the number of tasks completed by the players on that team. Each player will have their own list of tasks to be completed.

3.1.4 Flowcharts

The flow diagram to the left (Fig. 17) shows the execution flow of the game for a single player.

The event marked **PLAYER_FINDS_OBJECT** is activated when a player enters the trigger zone of an object. In this case, the condition **OBJ_TASK == PLAYER_TASK** checks to see if any of the IDs of the player's tasks matches the object's task IDs. If so, the object is highlighted, and the player's task button is activated. If not, or if the player moves out of the trigger zone, the object sprite remains inactive.

The user may select the return button at any time. If selected, the player returns to the main map.

The **ALL_TASKS_COMPLETE** condition makes reference to all the tasks of all the players on a single team. When this condition is met, the end screen is shown and the game finishes. Otherwise, the players continue until the last task of a single team is completed.

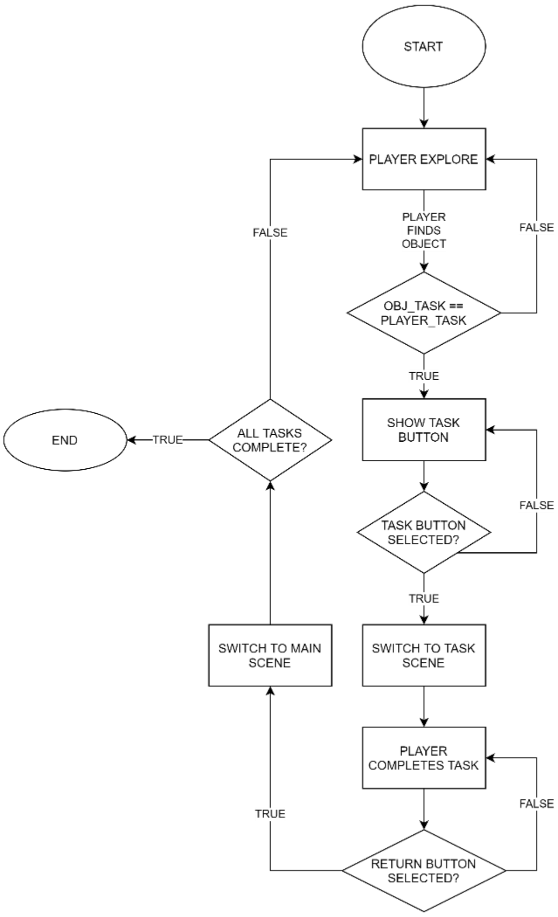


Figure 17: Gameplay Flowchart

3.1.5 Graphics & Artistic Inspiration

The main sources of inspiration for the artwork of this game, as previously mentioned, were the game Among Us, for colours, and the Despicable Me Minions. The reason both inspirations were chosen was for their iconic status in contemporary life, and their engagingly simple 2-Dimensional cartoonish graphics.

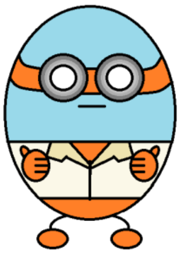


Figure 18: Player Sprite

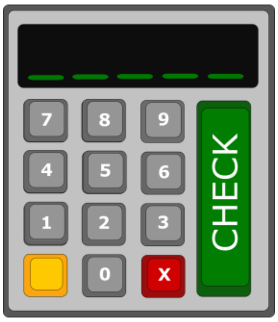


Figure 20: Calculator Sprite (Task)

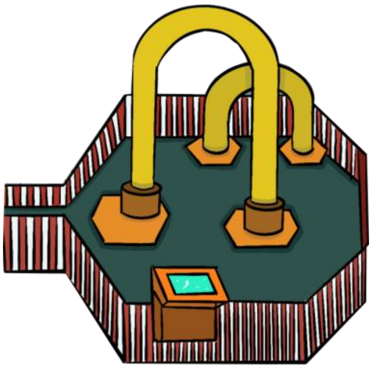


Figure 19: System Pumps (Object)

When considering the theme of the game, the most popular idea among those surveyed was the idea of a spaceship or space race (hence the name). This prompted the consequential endgame idea of having to be the first team to launch a rocket into space. To follow along with this theme, the player sprite was given a cartoon rendering of a space suit, the task objects take inspiration from elements found in space stations and research laboratories, and the tasks themselves take on a technical appearance, heavily featuring images of electronics and screens.

Figure 20 is a mock-up of the sprite that the players will use. This sprite was developed using Microsoft Paint. Considering that the players will be placed on teams, the jumpsuit worn by the player sprite will take on the colour of the team being represented, in this case, orange. The design is expanded on in the Game Design Document (Appendix A). Figure 19 is a mock-up of one of the elements of the numeric task and was developed using Inkscape. It is described in more detail in the Game Development Document (Appendix B). Figure 18 is an example of one of the objects being used to represent the tasks within the game and was created in Microsoft Paint. The objects are described in more detail in the Game Development Document (Appendix B).

The sprites were created using a combination of Microsoft Paint and Inkscape. The animation process was done through a series of built-in Unity tools, such as the sprite animator, and the OnClick button animator.

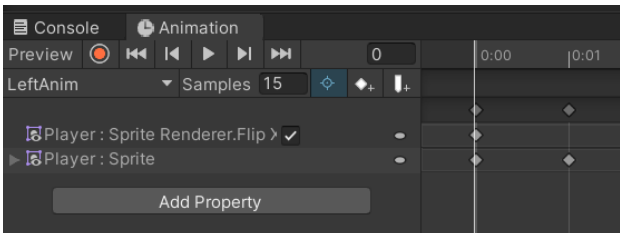

LeftAnim
Samples 15
Player : Sprite Renderer.Flip X ✓
Player : Sprite

Figure 21: Unity Sprite Animator

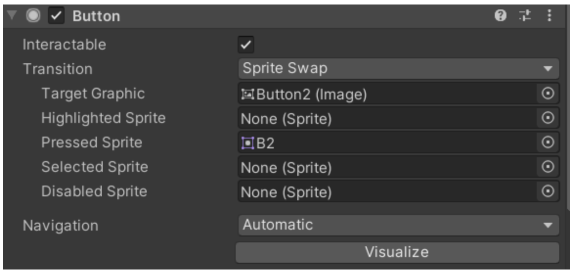

Button
Interactable ✓
Transition Sprite Swap
Target Graphic Button2 (Image)
Highlighted Sprite None (Sprite)
Pressed Sprite B2
Selected Sprite None (Sprite)
Disabled Sprite None (Sprite)
Navigation Automatic

Figure 22: Unity Button Animator

For more detail into the creative design process, see the Game Design Document (GDD) (Appendix A). For the initial sketches of both the character and map, along with task ideas and requirements, see Appendix E.

3.1.6 Component Integration

The following section describes the intended relation between the components that comprise the software.

The initial design requires the integration of an HTML page for the recompilation of question data, which is then sent to the database through the game server. These components will be expanded upon in the companion TFG. The game software developed in this TFG will be installed on end user (student) computers and will communicate positions and task completion with the server.

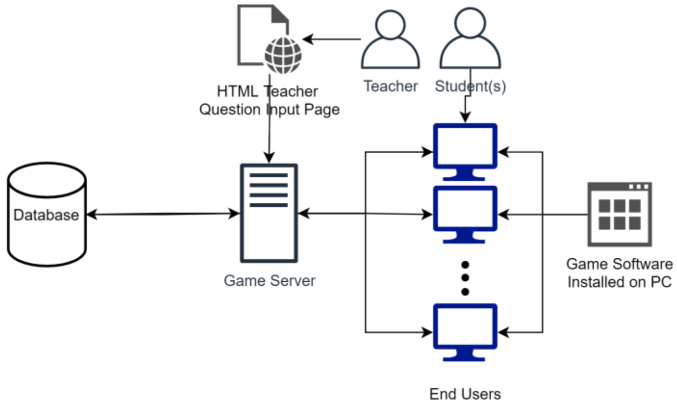


Figure 23: Game Components

4 Development

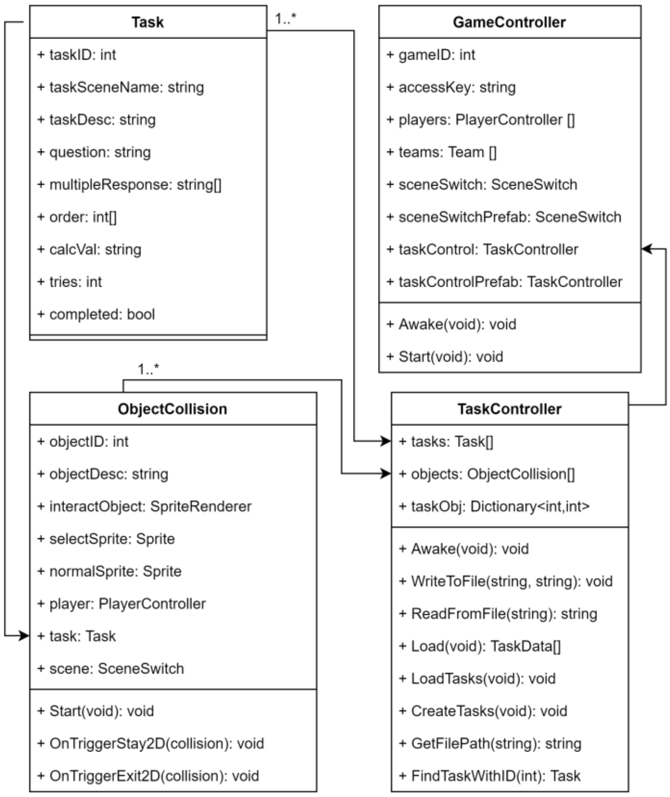
4.1 Code Development

The code for this project was developed in C# using the Microsoft Visual Studio 2019 code IDE with specific library imports for Unity game development. The programming paradigm used in the generation of this code was Object-Oriented because the native language of Unity is C#, which facilitates this type of programming. This also allows for the inheritance described in the diagram above, and a greater abstraction to each of the game components to reduce the complexity of the object code.

The design patterns used in this code are the Singleton and the Mediator. The Singleton pattern is used in objects such as the TaskController, as it needs to maintain the tasks throughout the game, despite the switching between scenes, as well as maintain a record of which objects have been assigned which tasks. The SceneSwitch object also uses this pattern, though the object itself is instantiated at game launch because it must be maintained through changing of scenes, but it also contains elements that are only known at the time of the scene switch (i.e., scene to switch to, task to switch to, etc.). The GameController object uses the Mediator design pattern because it acts as an intermediary for communications between various entities (player-object, player-player, player-team).

Additionally, the tasks and objects are created using specific interfaces so that the functionality is not overlooked in future iterations of this game. This also allows for the inheritance feature to be taken advantage of in C# Object-Oriented programming.

The final relationships between the code objects are as follows:



Task
+ taskID: int
+ taskSceneName: string
+ taskDesc: string
+ question: string
+ multipleResponse: string[]
+ order: int[]
+ calcVal: string
+ tries: int
+ completed: bool
1..*
GameController
+ gameID: int
+ accessKey: string
+ players: PlayerController []
+ teams: Team []
+ sceneSwitch: SceneSwitch
+ sceneSwitchPrefab: SceneSwitch
+ taskControl: TaskController
+ taskControlPrefab: TaskController
+ Awake(void): void
+ Start(void): void
1..*
ObjectCollision
+ objectID: int
+ objectDesc: string
+ interactObject: SpriteRenderer
+ selectSprite: Sprite
+ normalSprite: Sprite
+ player: PlayerController
+ task: Task
+ scene: SceneSwitch
+ Start(void): void
+ OnTriggerStay2D(collision): void
+ OnTriggerExit2D(collision): void
TaskController
+ tasks: Task[]
+ objects: ObjectCollision[]
+ taskObj: Dictionary<int,int>
+ Awake(void): void
+ WriteToFile(string, string): void
+ ReadFromFile(string): string
+ Load(void): TaskData[]
+ LoadTasks(void): void
+ CreateTasks(void): void
+ GetFilePath(string): string
+ FindTaskWithID(int): Task

Figure 24: Relationships Between Object Classes

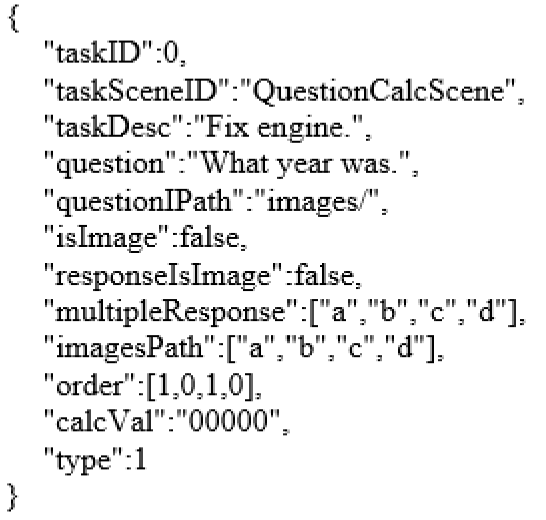
4.2 Object Structure

Object Structure refers to the way the objects in the game were created to handle the tasks and their corresponding scene switches. As demonstrated in Figure 24, the object ObjectCollision contains a reference to a task which is defined by the TaskController at the beginning of the game. As the TaskController object is a Singleton, the dictionary which contains the object IDs mapped to the task IDs is not deleted and therefore survives the scene switches. Each time the main map scene is started (calling the Start function), the objects are once again associated to their corresponding tasks, despite having been removed from the scene.

For the player interactions, the ObjectCollision tests whether the collision has come from an object with the *Player* tag and if the player has the object's task ID in their list of tasks. If so, and if the task has not been completed, the object sprite will switch to the highlighted sprite and activate the *Start Task* button located in the

4.3 Question Bank & Task Structure

The tasks are programmed in JSON and are read from local files. The initial idea was to have an HTML web page for the professors to input their questions and save the questions in JSON format to a NoSQL database for their in-game referencing. As the server is not currently functioning at the time of redaction, the JSON files are stored locally and take on the following structure:



```
{
   "taskID":0,
   "taskSceneID":"QuestionCalcScene",
   "taskDesc":"Fix engine.",
   "question":"What year was.",
   "questionIPath":"images/",
   "isImage":false,
   "responseIsImage":false,
   "multipleResponse":["a","b","c","d"],
   "imagesPath":["a","b","c","d"],
   "order":[1,0,1,0],
   "calcVal":"00000",
   "type":1
}
```

Figure 25: Example JSON Task

Visually, each task was designed to be its own scene where the user interacts with a canvas in various ways, such as dragging and dropping pieces into their correct slots, clicking buttons, and inputting numbers. The taskSceneID value determines which task scene to switch to. The question value refers to the task question. The multipleResponse value contains the possible answers to the question. The order value contains the given order of correctness of each answer in the multipleResponse value. In the case of the multiple-choice button task and the categories task, the order contains values in the range [0,1] in which 0 is false and 1 is true. In the case of the ordering task, the values can be in the range [0,3], where 0 is the first in the order and 3 is the last. The calcVal value is used for the calculator scene and contains the answer to a numerical question. The rest of the variables are unused at present and are configured for image capabilities.

In the case of the Multiple-Choice task, the variables from the JSON files used are the task ID, the task scene, the question, the multiple response values, and the order value, where order is an array of Boolean values in which the 1s represent correct values and the 0s represent incorrect values. This scene allows for single and multiple selections simply by modifying the number of 1s in the order array. In the case of the Calculator task, the variables used from the JSON files are the task ID, the task scene, the question, and the calcVal. The calculator scene only allows inputs of up to five digits, and they must be placed in written order from left to right. In the case of the Drag N Drop scene, the variables used are the task ID, the task scene, the question, the multiple response values, and the order value, with order being an array of integers to be verified in ascending order. Finally, regarding the categories scene, the values used are the task ID, the task scene, the question, the multiple response and the order, where order is an array of Boolean values and objects are classified into the binary classes.

For more information on the specific implementation of each task and the necessary actions the user must take to complete each task, see the game development document in Appendix B.

4.4 Gameplay

This section describes the gameplay from the perspective of a student (referred to in this case as player)

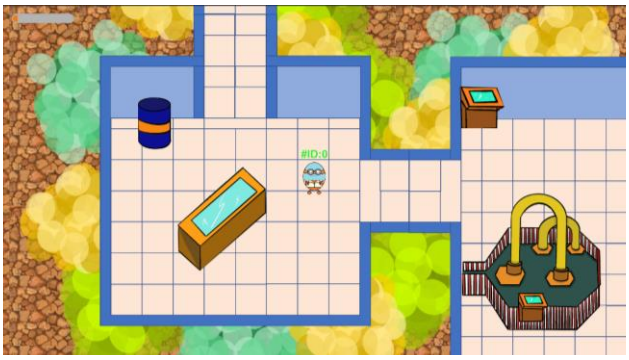


Figure 26: Entry Scene

The player starts in one of eight (8) rooms on the map (Figure 26). They use the arrow keys or the AWSD keys to move, exploring the map until they come upon an object that contains one of their assigned tasks. If the task IDs match, the *Start Task* button appears on the player's screen, as seen in Figure 27. Optionally, the player can select the **t** key on their keyboard to open their task board. This board shows the player which tasks have yet to be completed (red) and which have already been completed (green) and provides a brief description of the objects in which they can find their tasks (see Figure 28).

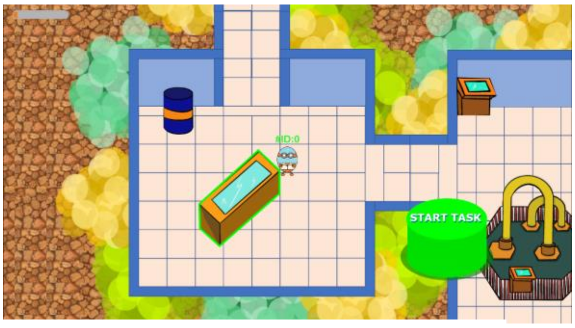
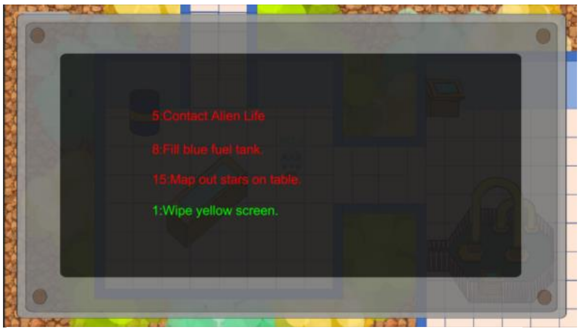


Figure 27: Task Button Activated



5:Contact Alien Life

8:Fill blue fuel tank.

15:Map out stars on table.

1:Wipe yellow screen.

Figure 28: Task Board Activated

The task screens can be visualized in the Game Development Document (see Appendix B). Once a task has been completed successfully, the progress bar located in the right-hand corner of the screen advances in a percentage comparative to the number of tasks left to complete by the team. Once all the team's tasks have been completed, the end game screen is shown, and the player can leave the application.

An example execution of the game can be found at the following link:
https://drive.google.com/drive/folders/1D6xgVdOwgiu51CXZPsD0WLTBfD4wfwYb?usp=sharing

5 Integration, tests & results

5.1 Integration

With regards to integration, as the backend server is pending implementation at the time of redaction, the software was not able to be tested to the fullest of its capabilities. Such aspects that could not be tested are the multiplayer implementation, the division of non-overlapping tasks, the division of teams, the global tracking of task completion of all players, and the player location tracking. Additionally, the teacher interface was designed to be an HTML page located on the server which allows them to introduce the questions in a user-friendly manner. That integrated feature can be found explained in more detail in the companion TFG: **SpaceRaceEdu: Desarollo de arquitectura y back-end para videojuego multijugador educativo**.

Regarding team and task implementation, the functionality was tested in single player mode during the unit tests described in the following section. The integration of tasks has been implemented through files for the purpose of testing, however the final question bank/task implementation will use a database located on the server and thus will need to be integrated in a future iteration of the project.

5.2 Tests

The tests were designed as acceptance tests for certain aspects of program functionality. The tests were performed at **runtime** (**RT**) and fall under the following categories: **Audio** (**AA**), **Animation** (**AN**), **Camera** (**CM**), **Collisions** (**CO**), **Launch** (**LC**), **Management** (**MM**), **Object** (**OB**), **Player** (**PL**), **Task** (**TK**), and **Team** (**TM**). Not all the tests could be performed seeing as the server is not operational at the time of redaction.

Table 4: Table of Tests

Test Identifier	Description	Classification
RT-AA01	Audio played during menu scene.	PENDING
RT-AA02	Background audio played during gameplay.	PENDING
RT-AN01	Player walks up, right, left, down, with left and right animation.	ACHIEVED
RT-AN02	Task button changes to pressed button sprite when touched.	ACHIEVED
RT-AN03	Objects change to highlighted sprite when it comes in contact with a player with the same task ID in their task list.	ACHIEVED
RT-CM01	Camera follows player who initiated game instance.	ACHIEVED
RT-CM02	Camera captures question screens fully.	ACHIEVED
RT-CO01	Object colliders trigger 'Start Task' button.	ACHIEVED
RT-CO02	Player collides with walls and objects.	ACHIEVED
RT-CO03	Player collides with other players.	ACHIEVED
RT-CO04	Player sprite maintains visual perspective when it collides with objects.	ACHIEVED
RT-CO05	Object collider disabled once the task has been completed.	ACHIEVED

RT-LC01	Object tasks are assigned on game launch.	ACHIEVED
RT-LC02	Players are assigned tasks on game launch.	ACHIEVED
RT-LC03	Object task dictionary is created on game launch to map objects to tasks.	ACHIEVED
RT-LC04	Player tasks appear on task board.	ACHIEVED
RT-MM01	Task completion is tracked globally.	ACHIEVED
RT-MM02	Completed tasks are displayed in green on the player's task board.	ACHIEVED
RT-MM03	Incomplete tasks are displayed in red on the player's task board.	ACHIEVED
RT-MM04	Players positions are tracked globally.	PENDING
RT-MM05	Scene switcher changes the scene based on the type of question.	ACHIEVED
RT-MM06	Scene switcher returns player to main map.	ACHIEVED
RT-MM07	Objects are assigned their original tasks after the scene switch.	ACHIEVED
RT-MM08	Game finishes when a team completes all the required tasks.	ACHIEVED
RT-MM09	Game is timed.	PENDING
RT-PL01	Player moves using **AWSD** keys.	ACHIEVED
RT-PL02	Player task board opens using **t** key.	ACHIEVED
RT-PL03	Player tasks are maintained through scene switch.	ACHIEVED
RT-PL04	Player task tries are recorded.	ACHIEVED
RT-PL05	Player task completion is recorded.	ACHIEVED
RT-PL06	Player task board shows required number of tasks (4).	ACHIEVED
RT-PL07	Player ID is not modified through scene switch.	ACHIEVED
RT-TK01	Multiple choice scene maintains selection pressed.	ACHIEVED
RT-TK02	Multiple choice scene allows for multiple selection.	ACHIEVED
RT-TK03	Multiple choice scene compares user selection with response correctly.	ACHIEVED
RT-TK04	Multiple choice scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK05	Multiple choice scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK06	Multiple choice scene allows users to try again when they answer incorrectly.	ACHIEVED
RT-TK07	Multiple choice scene contains a return button for users to abandon the incomplete task.	ACHIEVED
RT-TK08	Multiple choice scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK09	Multiple choice scene shows question and multiple answers from task details.	ACHIEVED
RT-TK10	Categories scene pieces drag and drop correctly.	ACHIEVED
RT-TK11	Categories scene pieces snap into place in category box.	ACHIEVED

RT-TK12	Categories scene compares user selection with response correctly.	ACHIEVED
RT-TK13	Categories scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK14	Categories scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK15	Categories scene allows users to try again when they answer incorrectly.	ACHIEVED
RT-TK16	Categories scene contains a return button for users to abandon the incomplete task.	ACHIEVED
RT-TK17	Categories scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK18	Categories scene shows question and multiple answers from task details.	ACHIEVED
RT-TK19	Categories scene pieces appear as picture or text.	PENDING
RT-TK20	Drag N Drop scene pieces drag and drop correctly.	ACHIEVED
RT-TK21	Drag N Drop scene pieces snap into place in category box.	ACHIEVED
RT-TK22	Drag N Drop scene compares user selection with response correctly.	ACHIEVED
RT-TK23	Drag N Drop scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK24	Drag N Drop scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK25	Drag N Drop scene allows users to try again when they answer incorrectly.	ACHIEVED
RT-TK26	Drag N Drop scene contains a return button for users to abandon the incomplete task.	ACHIEVED
RT-TK27	Drag N Drop scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK28	Drag N Drop scene shows question and multiple answers from task details.	ACHIEVED
RT-TK29	Drag N Drop scene pieces appear as picture or text.	PENDING
RT-TK30	Calculator scene buttons are pressable.	ACHIEVED
RT-TK31	Calculator scene prohibits more than 5 characters in answer.	ACHIEVED
RT-TK32	Calculator scene compares user selection with response correctly.	ACHIEVED
RT-TK33	Calculator scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK34	Calculator scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK35	Calculator scene allows users to try again when they answer incorrectly.	ACHIEVED
RT-TK36	Calculator scene contains a return button for users to abandon the incomplete task.	ACHIEVED

RT-TK37	Calculator scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK38	Calculator scene shows question from task details.	ACHIEVED
RT-TK39	Calculator scene question appears as picture or text.	PENDING
RT-TM01	Teams are automatically assigned with even numbers of players on each team.	PENDING
RT-TM02	Team progress is tracked in progress bar.	ACHIEVED
RT-TM03	Opposition progress is tracked in progress bar.	ACHIEVED

5.3 Bugs

The following table details the bugs and errors that were found during the testing phase.

Table 5: Table of Bugs

FUNCTIONALITY	**DESCRIPTION**	**CLASSIFICATION**
Animation	The player does not stop moving despite the input being null.	**BUG – Fixed 06.05.21** The player animator uses additional variables to monitor movement.
Animation / Collisions	Objects don't highlight when the player enters the object trigger range.	**BUG – Fixed 28.04.21** The collider was updated to track player ID and task ID.
Collisions	The object colliders do not trigger the 'Start Task' button.	**ERROR – Fixed 28.04.21** A new controller was created to register the collisions.
Collisions	The player sprite goes over the objects, but the perspective is incorrect.	**BUG – Fixed 26.04.21** The player collider was expanded to avoid perspective errors.
Player	The player needs to open personal task board by selecting **t** key, but key does not register on PlayerTaskController.	**ERROR – Fixed 28.04.21** The game controller was modified to register player input on keyboard (**t**).
Tasks	The calculator scene does not register more than four digits and the number that is calculated is not in the correct order.	**ERROR – Fixed 28.04.21** The loops for the text boxes were not properly instantiated.
Tasks	The image swap mechanism allows for images to be used in place of both questions and answers.	**ERROR – Fixed 11.05.21** The images folder was created, and all images must be placed in it to be able to be accessed. (NOTE: This will be tracked by server).
Tasks	Tasks do not survive transition	**ERROR – Fixed 06.05.21**

	from one scene to another.	The task manager was included in the game manager, which does not get destroyed when the scenes are switched.
Tasks	Player tasks not assigned upon game initiation.	**ERROR – Fixed 06.05.21** The game controller instantiates tasks and assigns them to the task controller singleton object.
Tasks	Player tasks are reassigned on switching scenes.	**ERROR – Fixed 09.05.21** The player tasks were written to local files and read from them once assigned. (NOTE: Server should monitor this).
Tasks	Player tasks do not register as completed on task board.	**BUG – Fixed 09.05.21** The task completion is now tracked by the task controller through IDs rather than the scene switcher.
Tasks	Images do not appear in image swap containers.	**ERROR – Unsolved**
Tasks	DragNDrop images grow if return is selected multiple times.	**BUG – Fixed 05.06.21** Return button disabled before selection is checked.
Teams	Player task completion not registered on team task completion bar.	**ERROR – Fixed 15.05.21** The team uses player tasks and player task completion marker.
Teams	Player not assigned to team.	**ERROR – Fixed 15.05.21** Game controller assigns players upon creation to team.
Scene	Endgame scene does not register scene change.	**ERROR – Fixed 10.06.21** Removed unregistrable variables.
Scene	End panel does not appear when team completes tasks.	**ERROR – Fixed 10.06.21** Added team name to the sceneswitch module to register team name upon completion.

5.4 Results

The initial user tests were performed by students in primary, secondary, and university level, as well as by teachers working at those levels. The tests were carried out by providing sample questions from past examinations or exercises depending on the students' level and allowing the students to explore the map after a brief demonstration of gameplay. As a result of the continued Covid-19 pandemic, the tests were not able to be carried out in person, however this further demonstrates the utility of such a tool in modern learning environments, highlighting its ability to be used remotely. The following sections delve deeper into the results achieved during the trial phase.

5.4.1 Questions Regarding Gameplay

To achieve a standardized response to the trial phase, the questions below were asked upon completion of the trial. The questions deal with both the implementation of the software and its usability in the classroom from the points of view of the teachers and the students.

Questions for students

- How does playing the game compare to taking an actual test? Do you feel like you would get a better grade on an exam styled this way or on a traditional exam?
- How would you rate the gameplay? How intuitive is the game? How entertaining is it?
- How does being able to play with teammates help/hinder the learning experience?
- How does being able to compete against classmates help/hinder the learning experience?
- How comfortable would you be playing this game in the classroom? How would you use this game in the classroom?
- Do you feel you would do better or worse taking an exam in this format? Why?
- Would you use this software to study?

Responses

For the sake of brevity, not all responses are noted in this section. The rest of the responses can be found in Appendix D.

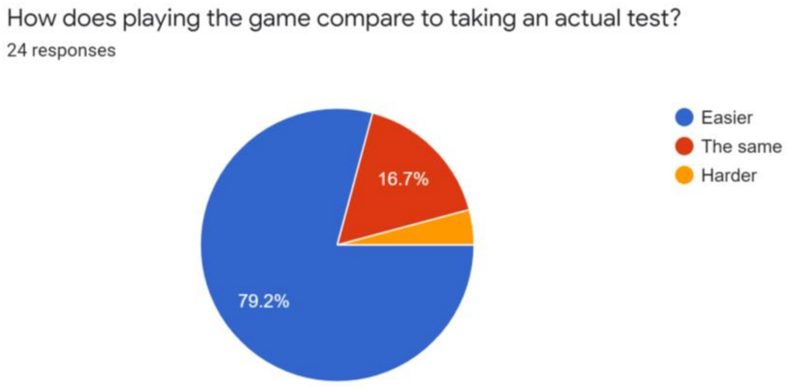


Figure 29: Results - How does playing the game compare to taking an actual test?

As seen in the chart above, most of the sample find this form of testing easier than traditional testing methods. In the short answer section of the questionnaire, the students use words like fun, entertaining, and relaxing, which all fall in line with the aims of the game. They also found it easier to complete the tasks when given in this format versus when given in traditional styled multiple choice.

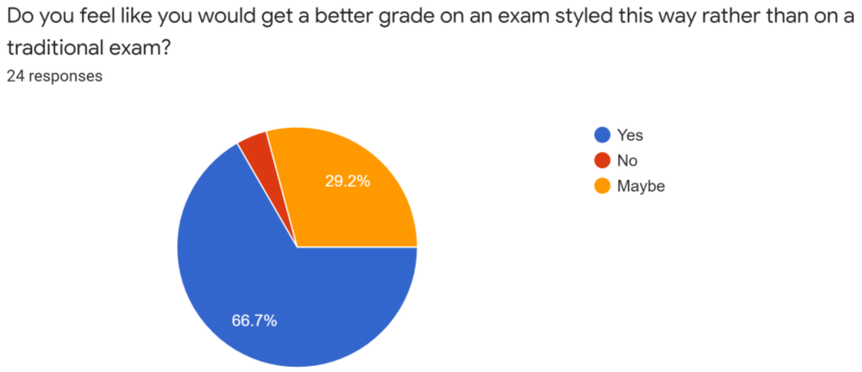


Figure 30: Results - Do you feel like you would get a better grade on an exam styled this way rather than a traditional exam?

Again, the students show their preference for the investigation software over traditional examination methods. As these results are based on feelings, further, possibly psychological studies should be done to fully qualify the effect of a person's emotions towards the examination versus the game in this particular instance. However, as seen in many of the reference guides researched in the initial phases of project development, there is a strong correlation between the way a test is perceived and how well a student performs on it.

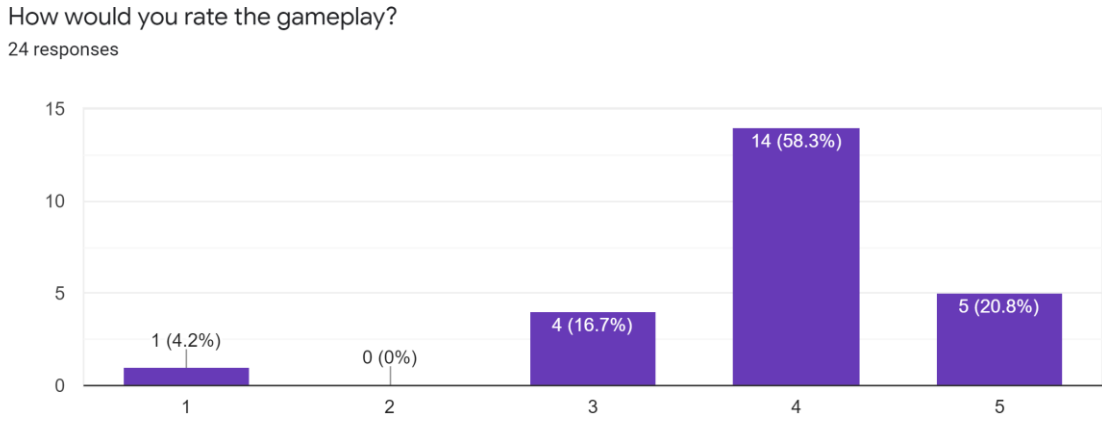


Figure 31: Results - How would you rate the gameplay?

As explained to the students, gameplay refers to the flow, entertainment factor, completeness, and relative easiness of the game. Thus, interpreting the results of the diagram, students appear to be moderately satisfied with the gameplay.

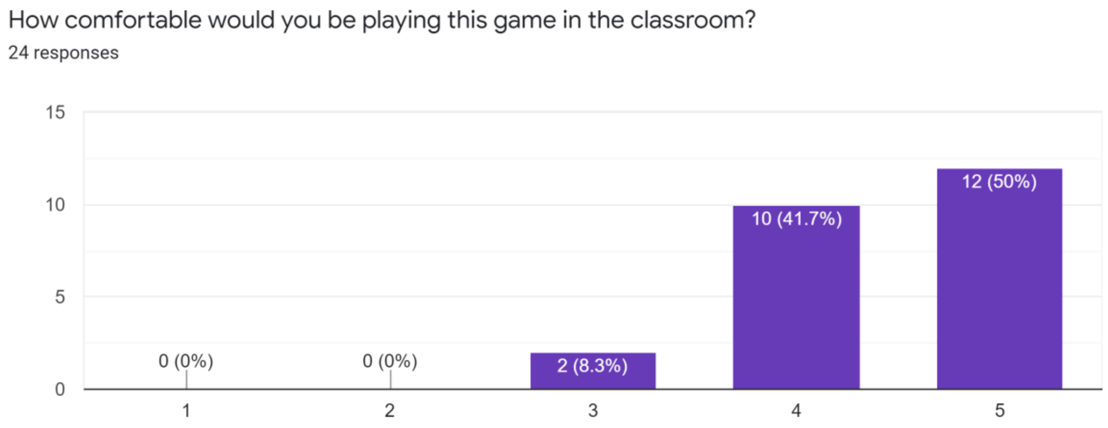


Figure 32: Results - How comfortable would you be playing this game in the classroom?

The results of this question show that the students would be moderately to very comfortable playing this game in class. In the written portion of this question, some mention the ability to play against their classmates and how that motivates them, while others enjoy the team building aspects of the software, mentioning their preference for working with their friends.

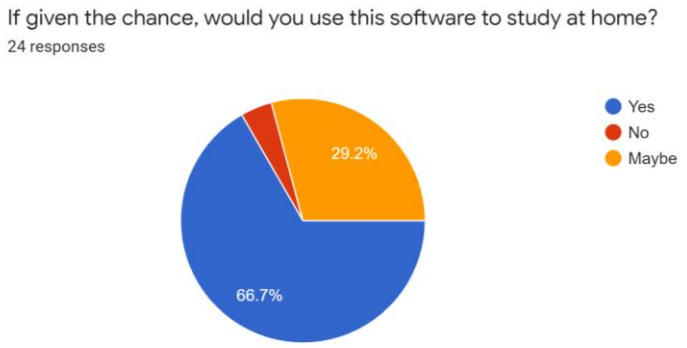


Figure 33: Results - If given the chance, would you use this software to study at home?

The final question stems from the need for online learning caused by the pandemic. As a result, the students responded positively towards it, having seen the value and potential that such a software could have in their education and in maintaining the atmosphere of teamwork even from a distance.

Questions for teachers

- Can your students use computers in class?
- Would gamified education be easily applied to your subject?
- Do you think it would provide benefits to learning?
- How could this style of questioning be implemented in your classes?
- How applicable is this software to your classes?
- Would you use this software for exams or reviews?
- In your opinion, would this encourage teamwork in your class?

- Would you give your students the option to use this software outside of the classroom?

Responses

For the sake of brevity, not all the responses have been placed in this section. The missing responses can be found in Appendix D.

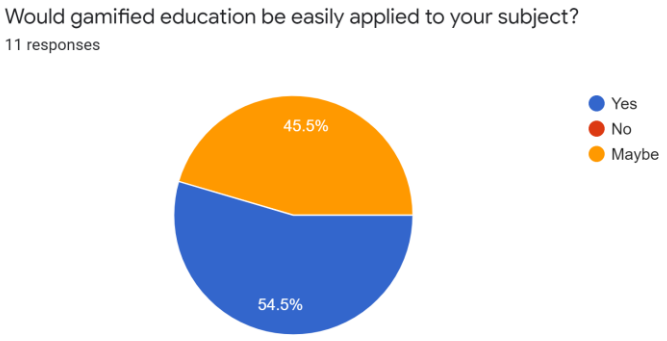


Figure 34: Results - Would gamified education be easily applied to your subject?

Surprisingly, most teachers interviewed are still hesitant to how gamification could be applied to their subjects, yet none have explicitly rejected the proposal. One of the main comments regarding the gamification of subject matter was that it would be more easily applied to primary education.

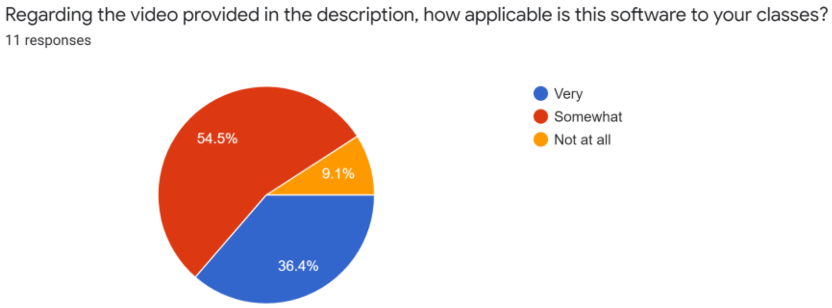


Figure 35: Results - How applicable is this software to your classes?

The teachers in the fields of upper-level education and teachers with more years of experience fell into the **somewhat** and **not at all** categories. As with the previous question, this is because university and upper-level education was considered harder to translate into simple multiple-choice styled tasks. As the software was not tested in person by these teachers, due to the global pandemic, most made a point to state that their results could change pending an in-person trial with multiple users.

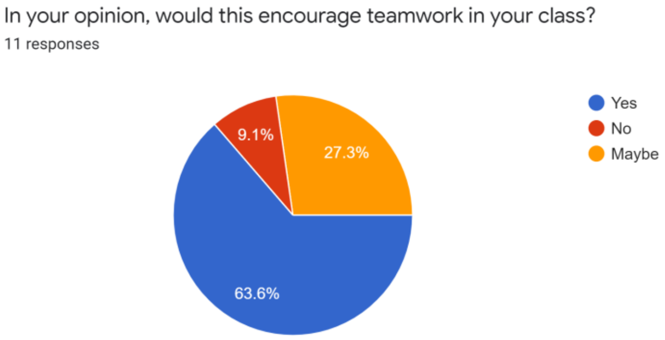


Figure 36: Results - In your opinion, would this encourage teamwork in your class?

Again, this question was heavily influenced by the fact that the trials were not held in person, and though most teachers could imagine the effects the software would have on their students in the multiplayer environment, the ones who selected the **maybe** and **no** options spoke solely about the version (single player) shown in the gameplay videos.

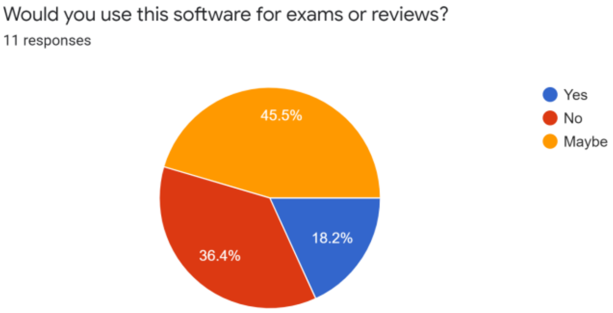


Figure 37: Results - Would you use this software for exams of revisions?

The results of this question were more varied than previously anticipated. This could be attributed both to the gameplay video, which only demonstrates the single player version of the software and in a limited capacity, or because the specific subject matter of the teachers does not allow for proper examination through multiple choice styled questions. One of the comments regarding this question was the inclusion of a short answer section for the questions that cannot be tested in the current manner.

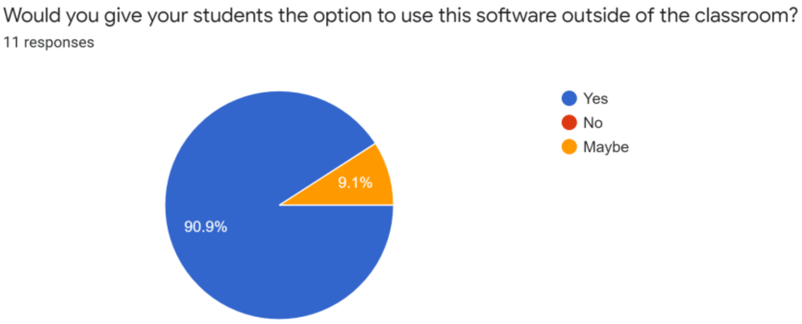


Figure 38: Results - Would you give your students the option to use this software outside of the classroom?

The final question regards the use of online learning in situations that do not permit in class learning, such as the pandemic. The response, as a result of this, was positive, which begs the question of how the pandemic will usher in a new age of education. For now, sufficed to say the software has demonstrated its merit in the increasingly isolated world during and post-pandemic.

5.4.2 Criticisms of Software

The general response was positive, although most teachers agreed that they would need to physically test the software in their own classrooms to see its effectiveness. Over 90% agreed to take part in future Beta trials. The criticisms focused on aspects of usability. For example, the lack of a guiding map to be able to locate the tasks, rather than having to explore and lose time not knowing where to go. Regarding gameplay, some teachers mentioned wanting to implement other types of questions, similar to games like Duolingo, which implement audio tests as well, and matching tasks. Others were sceptical about the style of questioning for their subject matter, as they consider it too complex to be reduced to multiple-choice style. Some teachers also considered the style of questioning to be too simplistic for university level education, but agreed that for lower levels, the game would offer benefits to both in-class and distance learning.

Other comments made were regarding the multiplayer integration of the game. Some teachers asked for the possibility to assign different teams to different games and using passcode access. Others asked for the ability to track the student progress through an external database, as they would the questions. These aspects will be fixed when the server is implemented, considering that they were already part of the game requirements.

5.4.3 Overall Results

One of the main takeaways from this experiment is that teachers are still hesitant to use this software as examination software, for a variety of reasons, such as the multiplayer aspects and the limited question style, in addition to the distracting elements that come with being a video game. They do however see a future in distance study and general, in-class revisions for this software, the latter being one of the intended uses, and the former resulting from the changed views of learning brough about by the global pandemic. For this, the teachers would require the use of the server and a question creation module, as at the moment, the software requires them to have basic knowledge of JSON, whereas the intended final product would not.

Regarding students and parents (the latter demographic emerging from the primary school demographic who were also searching for ways to educate and entertain their children during the pandemic), the results were an overwhelming success. The students enjoyed the character and the play style but thought that four tasks were too little. This caused them to want to play multiple times to see if they could complete all the different styles of tasks. The secondary and collegiate demographics both responded well to this form of questioning and found it easier to complete tests in this manner rather than in formal multiple-choice tests, because the gaming style made them feel more relaxed while taking the test. Additionally, considering the game takes inspiration from other popular games, the learning curve was minimal.

6 Conclusions & future work

6.1 Conclusions

This investigation began with the intent of creating an alternative to the standard student evaluation and self-study process. As seen in the previous sections, much of the functionality was implemented and follows the initial design idea set out in the Game Design Document (Appendix A). Despite the lack of server integration, the users were able to test the alpha version of the software successfully and were content with the usability and overall game experience. This alpha version included single-player mode with customizable task data. The alpha version brandished the four different task modes described above, with completion and attempt tracking capabilities.

Moreover, the typical phases of software development (requirement acquisition/analysis, design, development, test) were completed successfully, with all the required documentation (intermediate and final). These documents can be found in appendices A through C. As the project is still being developed, not all the functionalities could be completed/tested in the real-world setting, and as such, the development cycle will continue, taking on an iterative waterfall-styled management. This also implies that the lifecycle of this project software will take on additional phases that surpass the development of this investigation.

Furthermore, with respect to the project software, it can be concluded that there is still much work to be done, specifically in the areas of multiplayer integration and usability, however the initial version has tested positively in all the test demographics. The global pandemic provided an interesting backdrop on which to test this software because of the restrictions and limitations it brought with it. Despite this, the software has proven itself to be useful, even in its current, limited state.

6.2 Knowledge Gained

Thanks to this investigation, many of the aspects of project management learned in university have become consolidated knowledge.

With regards to the design and analysis phase of the project, the most crucial lesson learned was the importance of honing the requirements of a project. The more precise the requirements, the easier it is later to construct the project without having to guess whether or not an element is important. In addition, having to speak with the teachers and students both in the initial design phase and the final testing phase has brought important lessons regarding how to speak to clients and how to gauge their reactions.

Finally, regarding the technical implementation of this investigation (software and game development), the knowledge acquired relates to the correct implementation of object-oriented code and how this can affect the performance and memory usage of a program. In addition, the proper organization of the code came from the clarification of how to design class diagrams and algorithmic flow charts. Finally, this project forced lessons in the programming of C# which, though seen in class, was not greatly expanded upon.

6.3 Difficulties Encountered

The main issues encountered in this investigation stemmed from having to design a multiplayer game without the server side implemented. The trials were not as extensive as a result, and though the code works in single player mode, the response times, player tracking, question bank code, etc. have not been tested and therefore cannot be considered fully implemented.

Another set of difficulties arose in the form of the same pandemic that promoted interest in this investigation. Due to the continued restrictions of the covid pandemic, this software could not be tested on a larger scale and in physical classrooms for safety reasons.

6.4 Future work

Initially, some of the game controllers should be fixed to remove bugs and better their performance and stability, the colliders should be edited to reflect a more realistic gameplay, and additional task types should be added to reflect the original scope of the project. The functionality that was not implemented in this iteration of development should also be included in subsequent iterations. This includes server integration, to allow for the intended multiplayer online mode, statistical analysis, and timed games. In conjunction with server integration, an HTML page or compatible set up program should be created and hosted on the server to offer a user-friendly interface for the teachers to create and modify the question banks for their games.

Aside from bug fixes, a map should be created for the users who are not able to locate their assigned tasks. This could be implemented with another panel and highlighted markers showing where the tasks are located. Moreover, the statistics module should be expanded upon (though more can be done when server functionality is integrated) so the teachers may visualize the student data. It could also be expanded upon by using other factors that teachers deem useful with regards to student evaluations, such as completion time and student profiles, or even by creating modules that allow students to be able to evaluate their own progress.

As for the grander future of this project, the software would need to be adapted to make it more accessible to classrooms that are not well equipped technologically, i.e., classrooms lacking in computers or that do not have access to a stable internet connection. Finally, integration of Android/iOS functionality and cloud would potentially increase the accessibility and minimize latency. Additionally, this project has the opportunity for slight adjustments in gameplay in the form of the *avatarization* of the players, a reward system, and additional game enhancements, such as the ability to interact with other teams in the form of sabotage. The idea behind these modifications is increased participation and engagement.

References

[1] UNESCO. (2020, September 23). *How teachers are leading efforts to Ensure learning never stops During COVID-19 education disruption*. Retrieved May 09, 2021, from https://en.unesco.org/news/how-teachers-are-leading-efforts-ensure-learning-never-stops-during-covid-19-education

[2] Roberts, K. J. (2010). *Cyber junkie: Escape the gaming and internet trap*. Center City, MN: Hazelden.

[3] Huang, W. H. Y., & Soman, D. (2013). Gamification of education. *Report Series*: *Behavioural Economics in Action*, 29.

[4] Schleicher, A. (2020). *The impact of covid-19 on education insights from education at a glance 2020.* Retrieved from oecd.org website: https://www.oecd.org/education/the-impact-of-covid-19-on-education-insights-education-at-a-glance-2020.pdf

[5] AbilityPath. (2020, March 10*). Children's learning styles*. Retrieved May 01, 2021, from https://abilitypath.org/ap-resources/childrens-learning-styles/

[6] Comenius, J. A., & Keatinge, M. W. (1896). *The great didactic of John Amos Comenius, now for the first time Englished*. A. & C. Black.

[7] Graber, D. (2014, April 30). *Kids, tech and those shrinking attention spans*. Retrieved April 28, 2021, from HuffPost: https://www.huffpost.com/entry/kids-tech-and-those-shrinking-attention-spans_b_4870655

[8] Lynch, M. (2017, August 13). *8 must have gamification apps, tools, and resources.* Retrieved April 28, 2021, from TheTechAdvocate: https://www.thetechedvocate.org/8-must-gamification-apps-tools-resources/

[9] Huang, W. H. Y., & Soman, D. (2013). *Gamification of education.* Report Series: Behavioural Economics in Action, 29.

[10] Kiryakova, G., Angelova, N., & Yordanova, L. (2014). *Gamification in education*. Proceedings of 9th International Balkan Education and Science Conference.

[11] Cohen, A. M. (2011). The gamification of education. *The Futurist*, *45*(5), 16.

[12] Pokhrel, S., & Chhetri, R. (2021). A Literature Review on Impact of COVID-19 Pandemic on Teaching and Learning. Higher Education for the Future, 8(1), 133–141. https://doi.org/10.1177/2347631120983481

[13] Dhawan, S. (2020). Online Learning: A Panacea in the Time of COVID-19 Crisis. Journal of Educational Technology Systems, 49(1), 5–22. https://doi.org/10.1177/0047239520934018

[14] Park, S., & Kim, S. (2021). Is Sustainable Online Learning Possible with Gamification?—The Effect of Gamified Online Learning on Student Learning. *Sustainability*, *13*(8), 4267. doi:10.3390/su13084267

[15] Classcraft Studios Inc. (2021). *Classcraft Home*. Retrieved April 20, 2021, from Classcraft: https://www.classcraft.com

[16] YouTube. (2018). *Classcraft Walkthrough: Student Perspective. YouTube.* https://www.youtube.com/watch?v=8hZkXdgrQVQ&ab_channel=Mr.SchneiderMr.Schneider

[17] Kahoot! Inc. (2021). *Learning games: Make learning awesome!* Retrieved April 21, 2021, from Kahoot: https://kahoot.com/

[18] YouTube. (2017). *Kahoot Quiz for Medical Science. (2017). YouTube.* https://youtu.be/P9Fm6cBe6HA.

[19] BreakoutEDU. (2021). *Learn more: What is breakout edu?* Retrieved May 01, 2021, from https://resources.breakoutedu.com/learn-more-2020

[20] YouTube. (2020). *Breakout Edu Earth Treasure Hunt. YouTube.* https://www.youtube.com/watch?v=PdJdYTkFbHo&amp;ab_channel=SethiDeClercq

[21] Mojang Studios (2021). *Homepage: Minecraft: Education Edition.* Retrieved April 28, 2021, from https://education.minecraft.net/

[22] YouTube. (2019). *So I Tried Minecraft Education Edition.. YouTube*. https://www.youtube.com/watch?v=L4TUnW_gWvA&amp;ab_channel=AlexACEAlexACEVerified.

[23] MMOHuts. (2010). *Pirates of the Caribbean Online Gameplay - First Look Hd. YouTube. YouTube.* https://www.youtube.com/watch?v=D5CXc0hl5BY&amp;t=183s&amp;ab_channel=MMOHutsMMOHuts.

[24] Stott, A., & Neustaedter, C. (2013). Analysis of gamification in education. *Surrey, BC, Canada*, *8*, 36.

[25] Eriksson, U. (5 de April de 2021). *Why is the difference between functional and Non-functional requirements important?* Retrieved April 20, 2021, from ReQtest: https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/

[26] Goldstone, W. (2009). *Unity game development essentials*. Packt Publishing Ltd.

[27] Watkins, R. (2016). *Procedural content generation for unity game development*. Packt Publishing Ltd.

[28] Bond, J. G. (2014). *Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C*. Addison-Wesley Professional.

[29] Calabrese, D. (2014). *Unity 2D Game Development*. Packt Publishing Ltd.

Glossary

API	Application Programming Interface
Auditory	A style of learning which favours listening/hearing.
Avatarization	The process of recreating one’s self-image into that of an online avatar. This avatar holds personal value to its real-life counterpart because of the customization and personalization made to it.
Gamification	The process of adding game-like elements to tasks to make them more appealing and encourage participation.
GDD	Game Design Document
GUI	Graphical User Interface
Kinaesthetic	A style of learning which favours doing and moving.
Micro satisfaction	A dopamine induced brain state of short duration which stems from small achievements like those of playing games.
Question Bank	The repository of questions and answers used in the implementation of the game.
SCRUM	A style of project development that focuses on bursts of work known as sprints.
Tactile	A style of learning which favours touch.
Visual	A style of learning which favours seeing.
VR	Virtual Reality.

Appendices

A Game Design Document

Space Race

An Educational Odyssey

16/03/2021

Design Document

Escuela Politécnica Superior,
Universidad Autónoma de Madrid
May 2021

Contents

1 Game Development Team

Designers: Leah Hadeed, Juan Martinez, Juan Jesus Roldan Gomez
Frontend Developer: Leah Hadeed
Backend Developer: Juan Martinez
Artists: Leah Hadeed, Andrea Martin
Animator: Leah Hadeed

2 Game Overview

Title: Space Race
Platform: PC Standalone + Web
Genre: Platform, Mini-game, Educational
Age Restrictions: 6+
Players: Multiplayer Online (2-30)
Space Race is a third-person educational arcade game where the player is part of a team of space engineers looking to discover new planets. To fulfil their dreams of space travel, each of the team members must complete a series of tasks before they are able to launch the rocket. These tasks involve responding correctly to multiple choice typed questions. The first team to launch their rocket wins.

3 High Concept

The mingineers are a small race of minions that are trying to launch their rockets into space to discover other worlds. Each mingineer is part of a team that seeks to send their rocket off before their rivals. The game is set in a space station with nine different rooms, each with different tasks that need to be completed before the rocket can be launched.

The idea behind the game is to create a space for students to interact with each other while they resolve problems set by the teacher as either study exercises or even evaluation exercises. Each student takes on the role of a scientist trying to help his or her team send their rocket off to space before the other team. Each member has their own tasks to be completed and the group on a whole has one or two joint evaluation questions to resolve before they can finally launch their rocket. These tasks are designed to be questions relating to themes and topics studied in class and they follow one of six patterns (see Game Structure).

4 Game Objectives

The objective of the game is for each team to finish all their assigned tasks, individual and grouped, to be able to launch a rocket into space. This game is designed for those professors looking for a different way of evaluating their students in a way that foments teamwork, participation, and offers a diverting way of capturing their students' attention. As the professor is the one who determines the questions, this multifaceted game can be used either for personal or group evaluations as the need arises.

The game consists of teams of no more than six (6) players who need to work together to finish all their assigned tasks, each of which are related to one of the questions in the Question Bank, as stipulated by the professor. Each Sala (match) will have no more than four (4) teams. Each team is also given the option of 'sabotaging' the other teams in their Sala by adding more tasks to their workload. The saboteurs however must be careful, because to be able to sabotage, they must answer extra questions of their own, and in doing so, risk the other team advancing, and adding more tasks to their own workload.

Once everyone on the team has completed their tasks, the team can enter the launch room and fire their rocket off into space to colonize other planets. Each match has a time limit of thirty (30) minutes, and if no team manages to complete their tasks by that time, the winner is the team with the most tasks completed. The game will keep track of the statistics per match by way of an associated database.

4.1 Benchmarks

The benchmarks for the game prototype are as follows:

1. **Iteration 1 (non-playable) 30.03.2021:**
- Creation of main menu.
- Creation of player and main animations.
- Associate scene changes between main menu and main game.
- Create colliders.

2. **Iteration 2 (single player) 15.04.2021:**
- Creation of backgrounds.
- Creation of interactable objects and tasks.
- Animations for final rocket scene.
- Single Sala.

3. **Iteration 3 (multiplayer) 15.05.2021:**
- Integration with server system.
- Integration with question bank.

4. **Iteration 4 (beta version) 30.05.2021:**
- Beta testing.
- Bug fixes.
- Multi-Sala.

5. **Iteration 5 (production) 15.06.2021:**
- Final version.
- Bug fixes

5 Game Rules

The game follows the basic principles of a boardgame, on a multiplayer 2D platform arena. Much like other similar boardgames (Cluedo), this game contains a series of room each littered with missions (tasks) to be performed by each of the players on a team. Each team plays over their own version of the same map, and each Sala contains up to four of these replicated maps.

The players can move based on compass directions (north, northeast, east, southeast, south, southwest, west, northwest) however the animations take place on a one-dimensional plane (1D, left/right). Each player can interact with objects around the room that contain their specific tasks (pre-assigned at random at the beginning of the game based on a question bank created by their professors). These questions follow one of six (6) patterns based on their question type.

Each team must work to solve all their questions and fill the task bar to be able to access the launch bay and set off their rocket before the other teams in their Sala. The winning team is the one that sets off their rocket first, or, if no team manages to set off their rocket, the team with the most completed tasks.

6 Game Structure

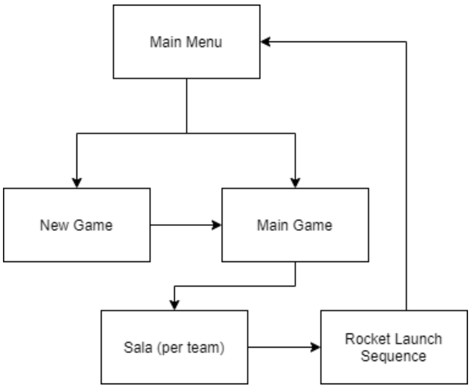


Figure 39: Gameplay flowchart

The structure of the game follows the typical arcade styled map where the game taxes place in a single scene. Players are placed on teams of minimum six (6) and a Sala is created for each team. The professors can log on and create new games, while the students can only join the game. Once a team reaches the stipulated requirements for winning the game, the scene shifts to the rocket launch sequence and the game ends.

6.1 Code Structure

For this project, the code is object-oriented to allow for maximum abstraction and easier implementation. As this project will be coded in Unity, the language of code is C#. Unity facilitates inheritance; therefore, each object will inherit their main class properties from the abstract entity, and for each type of object there will be a controller that monitors all the objects created in a certain game.
In particular, the classes are the following:

- ***Task*:** A question with answer to be attached to an object.
- ***Player***: An instance of a player for the multiplayer game.
- ***Game***: The new game object created by the teacher.
- ***QuestionBank***: The container for all the questions for the specific game instance.
- ***PlayerController***: Controller for all player instances in the game.
- ***TaskController***: Controller for all tasks in the game.
- ***GameController***: Controller for the game.

These scripts are responsible for propagating changes throughout the game and the effects of the users' actions on the game objects, for example, removing questions from the question bank, and declaring the win to the team whose tasks have been completed.

6.2 Game Controls

The controllers follow the Singleton design pattern on a per-Sala basis. Since the game is multiplayer, each different Sala must have the same internal controllers. The question bank is coded on a per-game basis, and thus also uses a Singleton. These controllers interact with the rest of the objects in the game on a pub/sub basis using the Observer/Subscriber notification pattern.

7 Gameplay

7.1 Controls

The game controls fall into two categories: movement and interaction. For all movements, the players use either the arrow keys or the AWSD keys, and the player can move in one of eight directions. The interaction is controlled by the mouse and is activated by clicking the highlighted object on the screen.

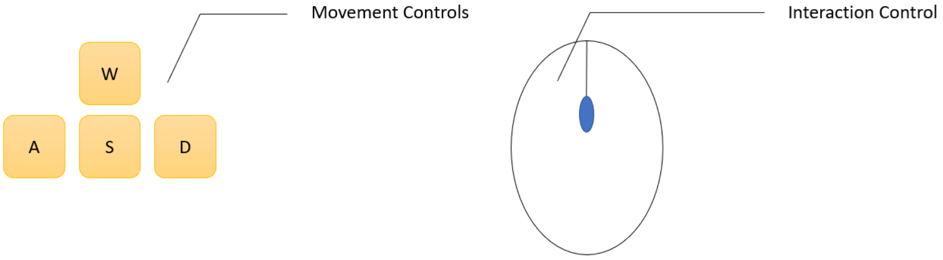


Figure 40: Description of controls

7.2 Camera

Since the map is fixed, the camera will be displaced to follow the user as they move through the scene. This way, the camera will be fixed to the transform of the user, allowing for a third person view styled game.

7.3 HUD (Head Up Display)

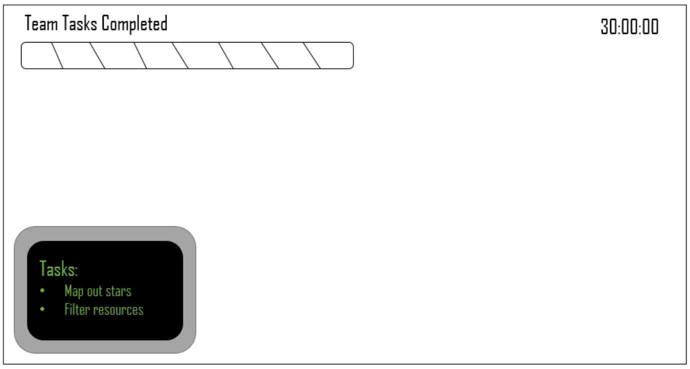


Figure 41: Example Player HUD

The Head Up Display (HUD) shows the global percentage of tasks completed by a single team (top left-hand corner) as well as the list of pending tasks for the user (bottom left-hand corner). In the top righthand corner sits the timer for each game. This is a countdown timer that marks how much time is left in this match. Though there are other aspects still to be contemplated, for now, like interaction buttons, for now, the objects with which a player can interact will appear in the game and not on the HUD. Since the game is space-related, the fonts and styles of HUD objects must follow the same space related theme, which is why they are set in terminal-styled colours and designs.

7.4 Animations

The animations for the game are all created with the Unity animator. To do so we join the sprites using the animator at a speed of 15fps. The players all have an animation controller which considers their current positions and the direction in which they are moving. This way, the animation flips to reflect the change in direction, however the player animation will only move left and right. These animations were developed using the Unity Blend Trees feature.

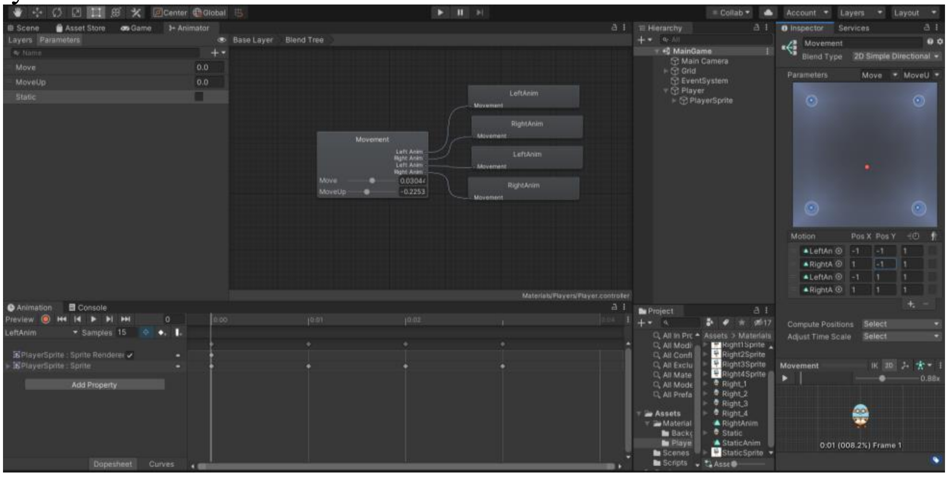


Figure 42: Unity Animation Pane (Blend Tree)

7.5 Menu

The menu will be click interactable. This means that the options will be selected with the mouse or the touchscreen. The menu is its own scene and will allow for the options to create, or join a game depending on the type of user and the type of login information provided.

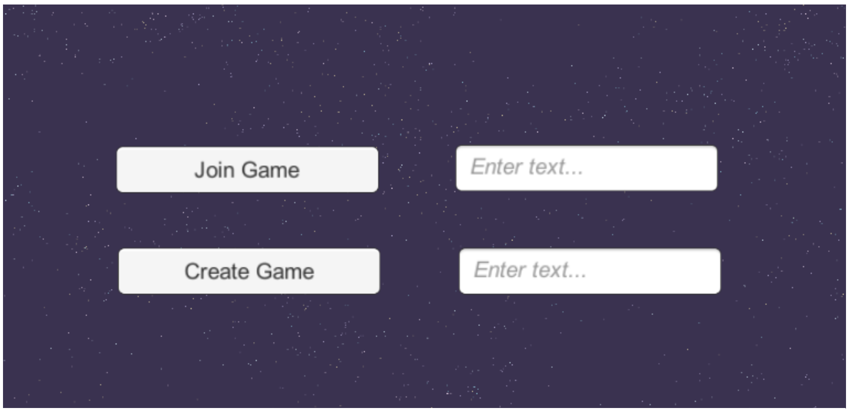


Figure 43: Main Menu Reference

As for the *Graphical User Interface* (GUI), we use a *UI Canvas* as the main node, with child panels that change the function of the menu as per necessity. The first screen is shown above. If the player chooses to create a game, they must log in with a teacher's password protected account. Otherwise, if they choose to join a game, they simply input their character's username as their login.

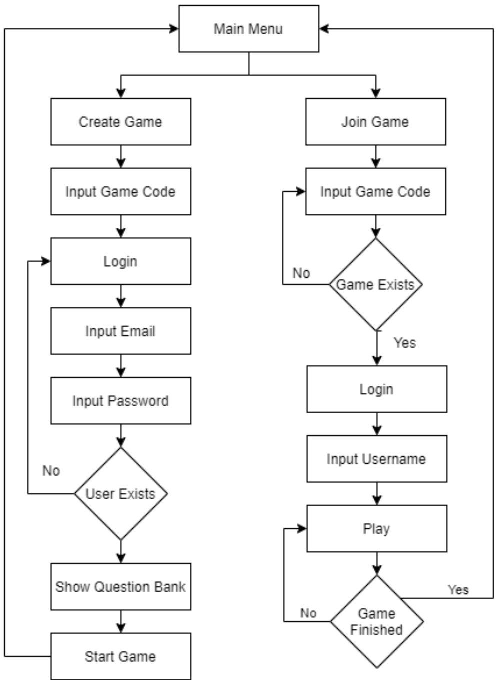


Figure 44: Main Menu Flow Diagram

The diagram on the left shows the typical flow of a game from the main menu perspective. As we can see, there are two main paths that a user can follow. The first path reflects a professor/teacher, who has a password protected account and can sign in to create/add/delete question banks and attach them to new games. The second path reflects the typical student user who does not have a saved/password protected account and can access created games through a professor provided code.

8 Player

The players will be assigned a different colour based on their team and Sala, but aesthetically, all of them follow the same basic pattern, shown on the right (Figure 39). The different colours depend on the team assigned at random at the beginning of the game. These colours are **Red, Orange, Yellow, Green, Blue, Purple.**

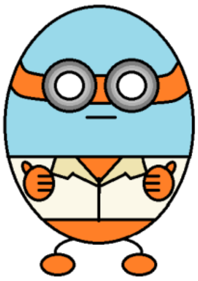


Figure 45: Example of player in orange

8.1 Player Movements

The players can move in each of the 8 cardinal directions and are coded as the following:

- **EE**: East.
- **NE**: North East.
- **NN**: North.
- **NW**: North West.
- **SE**: South East.
- **SS**: South.
- **SW**: South West.
- **WW**: West

9 Art

9.1 Background

The background is a spread of stars over which the main map is placed to make it seem like the players are in a space station in outer space.

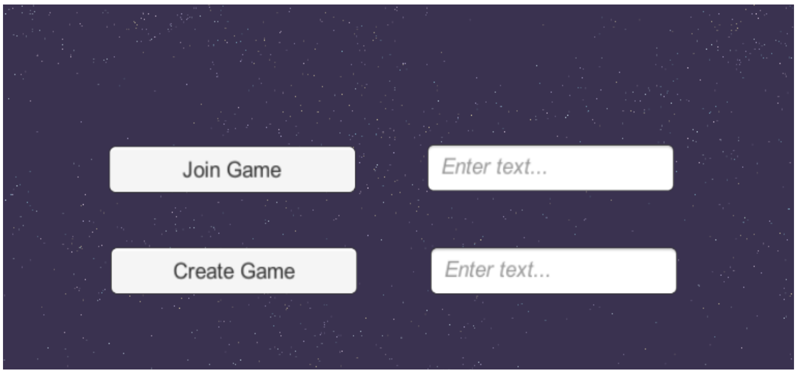


Figure 46: Main menu depicting starry background.

9.2 Foreground

The foreground contains the main map for the game. This map contains nine (9) different rooms where tasks can be allocated from the question bank. Each room contains objects that can be interacted with as part of a given task.

9.2.1 Rooms

The following is the official list of rooms on the map:

- Mission Control
- Cafeteria
- Engine Room
- Laboratory
- Communications
- Security Headquarters
- Electrical Room
- Defence Room
- Rocket Launchpad

The rooms are designed to have navigable spaces between all the objects for the players to move around and hallways that connect each of them to one another.

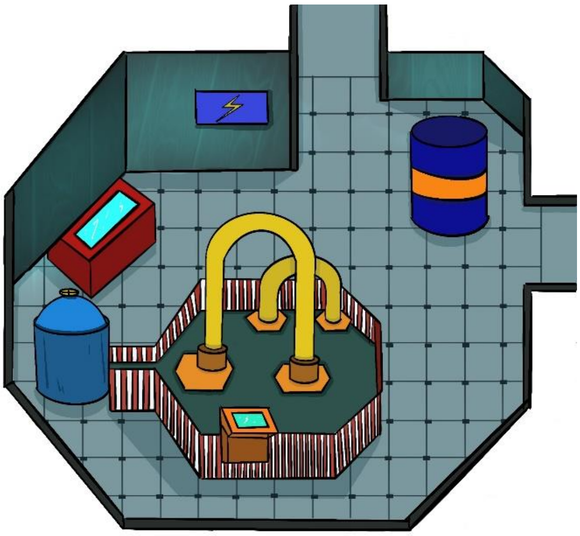


Figure 47: Example room (Engine Room)

9.2.2 Objects

The following is a list of objects that will be copied and spread throughout the rooms:Computer

- Telephone
- Satellite
- Monitor
- Maps
- Battery Panel
- Engine
- Test Tubes
- Speaker
- Communications Equipment
- Table
- Bins
- Motor
- Touch Panel

The objects are designed to have bright colours as to attract the players' attention.

10 Audio

For the audio section of this document, we will differentiate two possible sound types: Background, and Event. Background sounds are the sounds that are played in the main menu scene and the during the main gameplay. Event sounds are the sounds that are played when the user completes an action that warrants a sound. These divisions are reflected in the following sound table to distinguish the sound use cases as well as their scene cases.

SOUNDS

Name	**Category**	**Description**
menu_bg.wav	Background	The sound that is played in the main menu and the waiting area.
main_bg.wav	Background	The sound that is played throughout the game as the users complete their tasks.
menu_select.wav	Event	The sound that is played when the player selects one of the buttons in the main menu.
player_walk.wav	Event	The sound played when the player walks.
player_interact.wav	Event	The sound played when the player chooses to interact with an object.
question_correct.wav	Event	The sound played when a task is completed correctly.
question_false.wav	Event	The sound played when a task is completed incorrectly.
timer.wav	Event	The sound played when the timer reaches the five minutes remaining mark.
launch.wav	Event	The sound played when a team launches a rocket.

11 MVP (Minimum Viable Product)

11.1 Art & Animation

The game should include the entire playable map with limited object animation and full player movement animation.

11.2 Audio

The game should minimally include the main background sound.

11.3 Controls

The game should allow for user control through the movement keys and interaction using the mouse/pointer capabilities.

11.4 Objects

The game must make use of objects that connect to a question bank on the same server in which the game is hosted. These questions must be able to be assigned at random both to players as tasks, and objects whose functionality is dependent on said tasks. The player must be able to interact with these objects through the push of a button/click.

11.5 Platforms

The game needs to be able to support a single team with up to six simultaneous connections. The game must be able to function as a web-based application and have support for at least the Firefox browser. The game must have functional use of the controls stipulated in previous parts of this document.

11.6 Question Bank

The game must be able to access a registry (can be either JSON file or database) hosted on the same server which acts as a question bank for the tasks in the game. This question bank should be the result of an HTML static webpage hosted on the server that allows the professors to introduce questions and answers into a suitable question format.

11.7 Servers

The game should minimally be able to host six simultaneous connections (single team of multiplayer connectivity). The players should all be able to react/respond to server changes in real-time/near real-time. The players' movements must be tracked and registered on the server.

B Game Structure Document

SpaceRaceEdu

An Educational Odyssey

21/04/2021

Structure Document

Escuela Politécnica Superior,
Universidad Autónoma de Madrid
May 2021

Contents

1 Project Description

1.1 Introduction

This document serves as a detailed report on the creation process of the SpaceRaceEdu Videogame, as well as a comparison between the planning and structure outlined in the Game Design Document (GDD) and the actual progression of work. This document also serves as a log report for the system tests and bugs, as well as an architecture/user manual for the implemented game. This document is divided into the following five (5) sections: Project Description, Project Management, Analysis and Design, Implementation, and Tests.

1.2 General Project Description

This version of the game serves as an initial prototype which allows for multiplayer, limited sala game execution. Essentially, the game allows for a maximum of eight (8) players, determined by TCP Connections, to connect to a single version of the game space and interact with their surroundings. The map is structured into eight (8) different rooms, as stated in the GDD, in which each user will have an allotted four (4) tasks to complete. These tasks take the form of educational mini games. Once all tasks are completed, the game finishes.

1.3 Achieved Functionality

In the original Game Design Document, the proposed functionality included bidimensional player movement with mouse interactions, interactable tasks, multiplayer support, multi-sala support, and an integrated question bank with automatic task assignation. Considering the backlog due to the game art section of this project, this prototype contemplates a more limited scope.

Specifically speaking, this prototype brandishes the following functionality:

- Player functionality
  - Bi-directional movement using keyboard keys.
  - Animated player movement.
  - Mouse interaction with objects.
  - Collision detection.
- Task functionality
  - Animated task board, accessed through keyboard key (t).
  - Four (4) different task types.
  - Global task tracking.
  - Individual task tracking.
  - Retry capabilities & tracking.

1.4 Development Environment

The main environment used for the game development was Unity, with Microsoft Visual Studio 2019 for script editing. To create the sprites and scenes, Inkscape, Microsoft Powerpoint, and Microsoft Paint were used (listed in order of most usage). For the animation of the sprites, the native Unity Sprite Animator was used. For project planning, the task planner Trello was used. The software BeepBox was used to create the 8-bit background music.

1.5 Team and place of project development

The development team is comprised of Leah Hadeed as the main frontend developer and artist, Juan Martinez as the main backend developer, and Andrea Martin as the co-artist. The project was mostly developed online using Unity Hub and a combination of Discord, Teams, and Whatsapp for project design and follow up meetings. These meetings involved Sprint-styled planning, aided by the Trello planning tool mentioned above.

1.6 Additional Resources

1.6.1 Graphics

As mentioned above, the graphics were created specifically for this project using the aforementioned tools, with the exception of the dirt background, which was found at https://www.pinterest.com/pin/402438916677486224/. The floorplan was created using powerpoint. The objects were created using paint and inkscape. The buttons and screens were created using inkscape.

1.6.2 Server

As previously mentioned, the server environment and connection API were developed by Juan Martinez as part of a separate investigation. The necessary code for the API was developed based on the requirements of the game to be able to track certain statistics, movements, elements, etc.

1.7 Project Management

As described previously in the Game Design Document, the following calendar details the intended benchmarks for the game development:

1. **Iteration 1 (non-playable) 30.03.2021:**
   a. Creation of main menu.
   b. Creation of player and main animations.
   c. Associate scene changes between main menu and main game.

2. **Iteration 2 (single player) 15.04.2021:**
   a. Creation of backgrounds and colliders.
   b. Creation of interactable objects and tasks.
   c. Animations for final rocket scene.
   d. Single sala.

3. **Iteration 3 (multiplayer) 15.05.2021:**
   a. Integration with server system.
   b. Integration with question bank.

4. **Iteration 4 (beta version) 30.05.2021:**
   a. Beta testing.
   b. Bug fixes.
   c. Multi-sala.

5. **Iteration 5 (production) 15.06.2021:**
   a. Final version.
   b. Bug fixes

However, as a result of delays in the artwork, the entire schedule has been adjusted to reflect the setbacks. This new version also includes design aspects that were previously not contemplated but have proven to be necessary in the implementation of the software. Examples of such deviations are the use of different controllers for the game, the tasks, and the players, as well as the modification of certain flow elements, like buttons and other UI elements. Therefore, the following is the updated version of the benchmark planner:

1. **Iteration 1 (non-playable) 30.03.2021:**
   a. Creation of main menu.
   b. Creation of player and main animations.
   c. Player movement controller.
   d. Associate scene changes between main menu and main game.

2. **Iteration 2 (non-playable) 15.04.2021:**
   a. Creation of task scenes.
   b. Main map with colliders and objects.
   c. Creation of interactable object scripts.
   d. Player GUI.

3. **Iteration 3 (single player) 15.05.2021:**
   a. Integration with server system.
   b. Integration with question bank.

4. **Iteration 4 (multiplayer) 30.05.2021:**
   a. Beta testing.
   b. Bug fixes.
   c. Multi-sala.

5. **Iteration 5 (production) 15.06.2021:**
   a. Final version.
   b. Bug fixes

2 Analysis & Design

This section describes the outline of the game structure, namely the items used in the design and analysis phase of project development (class diagrams, flow diagrams, design patterns, etc.), as well as any possible deviations from the original plan for the project.

2.1 Requirements

Requirements were compiled from an initial survey completed by a series of teachers across the globe (Table 10) and from a series of conversations with the professors of the Universidad Autonoma de Madrid and the Universidad Politecnica de Madrid.

Select your usual teaching methods.	Select your usual examination methods.	On a scale of 0 - 10, rate the productivity of your students.	On a scale of 0 - 10, rate the attention span of your students.	On average, what grades do your students achieve using your current examination tactics? (10 = 100%)	What are the 5 most important facets (in your opinion) of a modern day education?	Give us your thoughts on what videogame-styled learning looks like to you.
Physical Labs/Hands on work, In class debates/Group learning activities	Group projects, Kahoot/Classcraft/Similar, Other	7	8	8	Class participation, Team building exercises/Group projects, Discussion-based activities, Interactive classrooms (Technology), Student progress	A videogame where the students/participants must answer a series of questions to make progress, similar to a trivial game.
Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, Documentaries, In class debates/Group learning activities	Formal written exam	8	7	8	Team building exercises/Group projects, Discussion-based activities, Interactive classrooms (Technology), Expert groups, Continuous evaluations	It would be an interest tactic to implement. An extension of using play in learning
Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, Documentaries, In class debates/Group learning activities	Essays/Reports, Group projects, Individual projects, Kahoot/Classcraft/Similar	7	8	8	Team building exercises/Group projects, Hands on work/Labs, Discussion-based activities, Expert groups, Student progress	it depends on what type of video games
Whiteboard/Chalkboard	Essays/Reports	6	5	8	Individual study, Team building exercises/Group projects, Discussion-based activities, Interactive classrooms (Technology), Student progress	Interactive digital learning, vidoes, educational games or discussions
Powerpoint/Other presentation software, Textbooks/Other book work, Documentaries, In class debates/Group learning activities	Formal written exam, Essays/Reports, Group projects, Individual projects	8	8	8	Class participation, Individual study, Discussion-based activities, Student progress	This method could Engage all students; but teachers would need to be comfortable with all facets of the technology
Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Other	Formal written exam, Individual projects	9	7	8	Team building exercises/Group projects, Hands on work/Labs, Discussion-based activities, Interactive classrooms (Technology), Student progress	It is an interactive way of learning, using fun techniques that most students are now familiar with and feel comfortable with. Students tend to enjoy learning in a fun way.
Powerpoint/Other presentation software, Textbooks/Other book work	Formal written exam, Individual projects, Multiple choice tests	7	6	8	Class participation, Individual study, Discussion-based activities, Interactive classrooms (Technology), Student progress	Really engaging for the students
Whiteboard/Chalkboard, Powerpoint/Other presentation software	Multiple choice tests	8	6	7	Discussion-based activities, Student progress	I think it would be good for some students but it wouldn't work for most kids. I feel like it can be distracting
Whiteboard/Chalkboard, Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, In class debates/Group learning activities	Formal written exam, Essays/Reports, Multiple choice tests	7	7	6	Team building exercises/Group projects, Hands on work/Labs, Discussion-based activities, Interactive classrooms (Technology), Student progress	I have no idea, i guess like among us? Would love to find out more about this!
Powerpoint/Other presentation software, Textbooks/Other book work, Physical Labs/Hands on work, Documentaries, In	Essays/Reports, Group projects, Individual projects, Multiple choice	5	7	8	Class participation, Team building exercises/Group projects, Hands on work/Labs, Discussion-	Playing games to learn a subject

Table 6: Teacher Opinion Survey Results

2.1.1 Functional Requirements

2.1.1.1 Multiplayer Mode

FRMM-1 Simultaneous play for 2 – 8 players per sala at a time.

FRMM-2 Creation of different teams of players and assign a different sala to each for the match.

FRMM-3 Simultaneous reproductions of the main game map per team.

FRMM-4 The teams should have their completed and pending tasks tracked to gage the progression of the game.

FRMM-5 Only members of the same team should see each other represented on the map.

2.1.1.2 Players

FRPL-1 Players must move through the 2D map space both horizontally and vertically.

FRPL-2 Identification (name or ID) of each player must be visible to the other players at all times.

FRPL-3 Players must be able to interact with the objects throughout the 2D map space to be able to complete the assigned tasks.

FRPL-4 Each player has a random set of tasks.

FRPL-5 Players must have a record of their completed and pending tasks, and the number of tries taken to complete each task.

FRPL-6 The players must have a point of reference for globally completed tasks.

2.1.1.2 Tasks

FRTS-1 The tasks are represented on the map as interactable objects.

FRTS-2 Only the tasks assigned to the player should appear as interactable on that player's map.

FRTS-3 Only the tasks assigned to the player should appear on their task board.

FRTS-4 The tasks should take the form of interactive mini games.

FRTS-5 The tasks should come in a variety of visually appealing forms to avoid repetition.

FRTS-6 Each task has a question and a corresponding answer.

FRTS-7 There can be one or more correct answers.

FRTS-8 The tasks should reflect the environment/interactable that they are associated with.

FRTS-9 There must be a global registry of tasks, completed tasks, and pending tasks.

FRTS-10 These tasks must also be divided into teams and players.

FRTS-11 Once the task is completed, it should be removed from the player's pending tasks list and the registry of this task should be added to the global completion of the team's tasks.

FRTS-12 Each task should allow for retries of the same task.

FRTS-13 Each task should be able to be exited and returned to if not completed.

FRTS-14 Once a task is retried, a try count should be updated as part of user statistics (See FRGM-3).

2.1.1.2 Game Master

FRGM-1 The teacher must be able to create new games from within the game.

FRGM-2 The teacher needs an environment to be able to input new questions and their corresponding answers (question bank).

FRGM-3 The game should offer statistics on each match and each player.

FRGM-4 The game's statistics should be tracked for the teacher to gauge the learning curve of the students (See FRGM-3).

FRGM-5 The game master should be able to define access to the game.

2.1.1.2 Match

FRMA-1 The match should be timed and run for no more than 30 minutes.

FRMA-2 Each match should have a reference number to locate statistics of the execution (See FRGM-5).

2.1.2 Non-Functional Requirements

2.1.2.2 Graphics

NFGR-1 The graphics should be visually appealing and simple to encourage participation and further promote the concept of gamified learning.

NFGR-2 The game colours should be visually inoffensive and not representative of things that would otherwise be considered offensive to certain groups, regions, religions, etc.

NFGR-3 The characters should be simple and not representative of persons living or otherwise. However, they should be appealing enough to promote the sensation of *avatarization*.

NFGR-4 The characters may take inspiration from other visually appealing characters, however they should not be direct copies of pre-existing or copyrighted characters.

NFGR-5 The tasks visually represented as objects on the map should be more attractive/eye-catching than the background.

NFGR-6 The players should have a visual list of the tasks they need to complete (task board).

NFGR-7 The player's Head Up Display (HUD) should show the progress made by their team and the global game progress through an animated progress bar.

NFGR-8 The tasks visually represented as objects on the map should be highlighted when a player with the same task ID in their task list enters their vicinity.

NFGR-9 The task scenes should be interactable screens with UI elements such as buttons, text boxes, and panels.

NFGR-10 The task scenes should have buttons that allow a player to return to the main scene regardless of whether or not they have completed the task.

NFGR-11 The buttons in the task scenes should all be clearly labelled.

NFGR-12 The text in the task scenes should be clearly readable, i.e. not with visually confusing fonts or colours that do not appear against their backgrounds.

NFGR-13 When a button is pressed, the pressed button animation should appear.

NFGR-14 For the scenes that require multiple responses, the buttons should clearly mark the different selected responses either by using a different colour or maintaining the pressed image.

NFGR-15 Buttons that serve the same purpose in the different tasks should have uniform graphics throughout the game (i.e. retry, return, check answer) so that the player knows what he or she is looking for.

NFGR-16 Correctly completed tasks should be highlighted as correct through some form of green visual signalling.

NFGR-17 Incorrectly completed tasks should be highlighted as incorrect through some form of red visual signalling.

NFGR-18 The tasks should allow for multiple forms of task representation (i.e. question/answer in the form of images and text).

NFGR-19 Completed tasks should be represented on the player's task board in green font, while incomplete or incorrect tasks should be represented in red font.

2.1.2.2 Program Weight

NFPW-1 The complete game should be lightweight to be uploaded and stored on the corresponding server.

NFPW-2 The program should be built with the settings of a Web App or a standalone program.

2.1.2.2 Usability

NFUR-1 The game should use keyboard inputs for player movement and HUD interactions.

NFUR-2 The game should use mouse inputs for object interaction and task completion.

NFUR-3 Tasks should be separated into two categories, drag and drop, and button related, and should register the appropriate inputs.

NFUR-4 The drag and drop tasks should register a sustained click and maintain the object under the pointer.

NFUR-5 The button related tasks should register the individual clicks and sustained clicks as the same.

NFUR-6 The multiple-choice task (subsection of button related tasks) should allow for multiple answers. This should be made visually apparent by disabling the buttons when pressed.

NFUR-7 When a multiple-choice question is incorrect or retried, the retry button should reset the enabled/disabled status of all the buttons.

NFUR-8 The calculator tasks (subsection of button related tasks) should register up to five different button clicks.

NFUR-9 The drag and drop tasks should permit the user to drag the object to the nearest edge of the desired slot and snap the object into place automatically.

2.2 Class Diagrams

The following subsection outlines the different classes implemented to achieve the desired game functionality, as well as the reason for the chosen design.

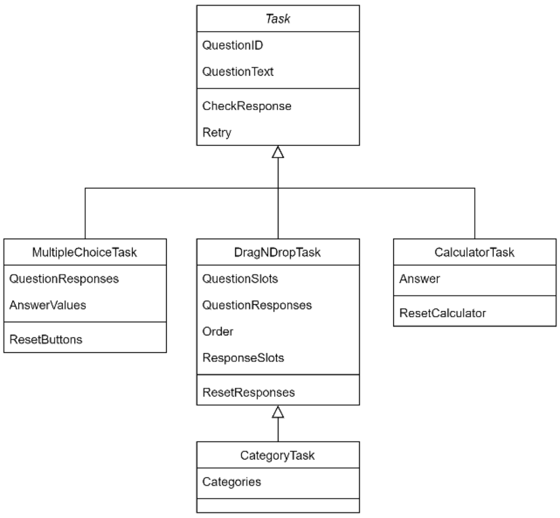


Figure 48: Task Class Diagram

The Task Class Diagram (Fig. 42) outlines the structure of the Task object in the game. The Task class in itself is an abstract class which the other classes inherit from. Each class share the Question ID and text variables, as well as the Check Response and Retry methods. For this reason, the abstract task class was created.

The classes that inherit from this class use different variables to control the responses depending on how the task itself was designed visually. For example, some of the tasks need Boolean variables for multiple response, while others are simply numeric answers.

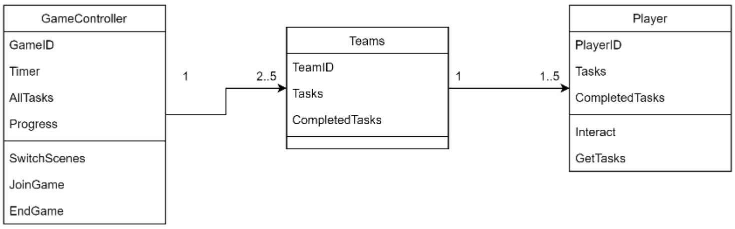


Figure 49: Player Class Diagram

The Player Class Diagram (Fig. 43) describes the Player class and the methods it contains. This diagram also demonstrates how the players are related to one another through teams, as well as how they are controlled on a global scale by the game controller.

The players contain a list of personal tasks which each contain a marker that determines their completion status.

2.3 Flow Charts

The flow diagram to the left (Fig. 44) shows the execution flow of the game for a single player.

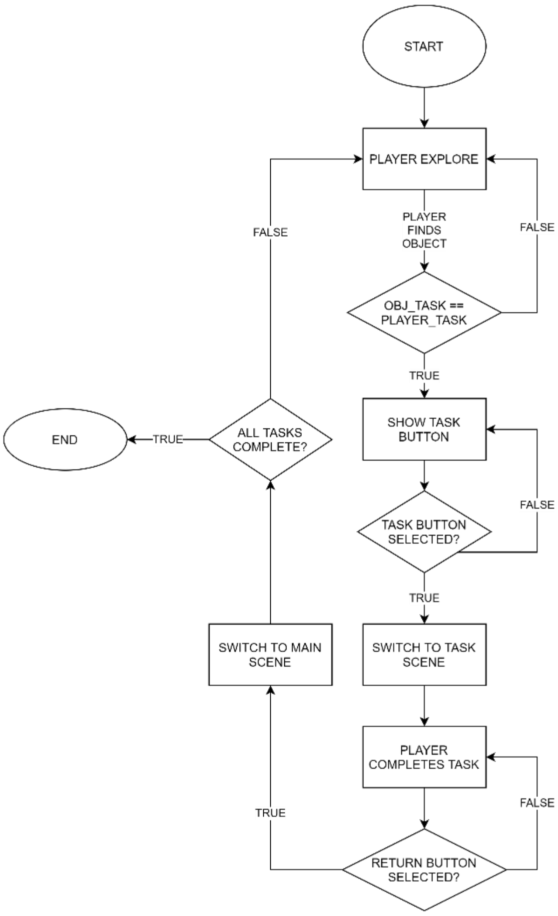


Figure 50: Game Algorithm Flow Chart

The event marked **PLAYER_FINDS_OBJECT** is activated when a player enters the trigger zone of an object. In this case, the condition **OBJ_TASK == PLAYER_TASK** checks to see if any of the IDs of the player's tasks matches the object's task IDs. If so, the object is highlighted, and the player's task button is activated. If not, or if the player moves out of the trigger zone, the object sprite remains inactive.

The user may select the return button at any time. If selected, the player returns to the main map.

The **ALL_TASKS_COMPLETE** condition makes reference to all the tasks of all the players on a single team. When this condition is met, the end screen is shown and the game finishes. Otherwise, the players continue until the last task of a single team is completed.

3 Implementation

The scope of this functional prototype in its current state englobes the four (4) previously mentioned task scenes, as well as the single player functionality. Since the server at the time of redaction is not operable and without API, the multiplayer operation of the game is not possible, and neither is the additional question bank feature. However, to be able to test in single player mode, a test question bank has been created locally for the purpose of parse testing and statistics tracking.

To go into greater detail, the game features bi-dimensional player movement throughout the main map. Using the arrow keys and the 'AWSD' keys, the player is able to explore the parts of the map that are not blocked by colliders. Continuing with the colliders, aside from the wall colliders, all of the objects on the map have their corresponding two colliders (the first to block passage, and the second as a trigger for the scene swap to the corresponding task). To interact with these objects, a button appears on the user HUD that allows the user to attempt the task that corresponds with that particular object if the ID of that particular task is also in the list of player task IDs.

The following subsections detail each of the different task types.

Calculator Scene

For this scene, the player inserts zero to five digits that correspond to the answer to the question on the right-side of the calculator. This answer is an **integer** value of one to five digits. This question can be in the form of an image or a text. If the player submits the correct answer, the digits of the calculator turn green, and the player is shown a congratulations screen that takes them back to the main map scene. If the player fails, the digits turn red and the player is given the opportunity to try again, with each additional try being added up to the statistics of the player.

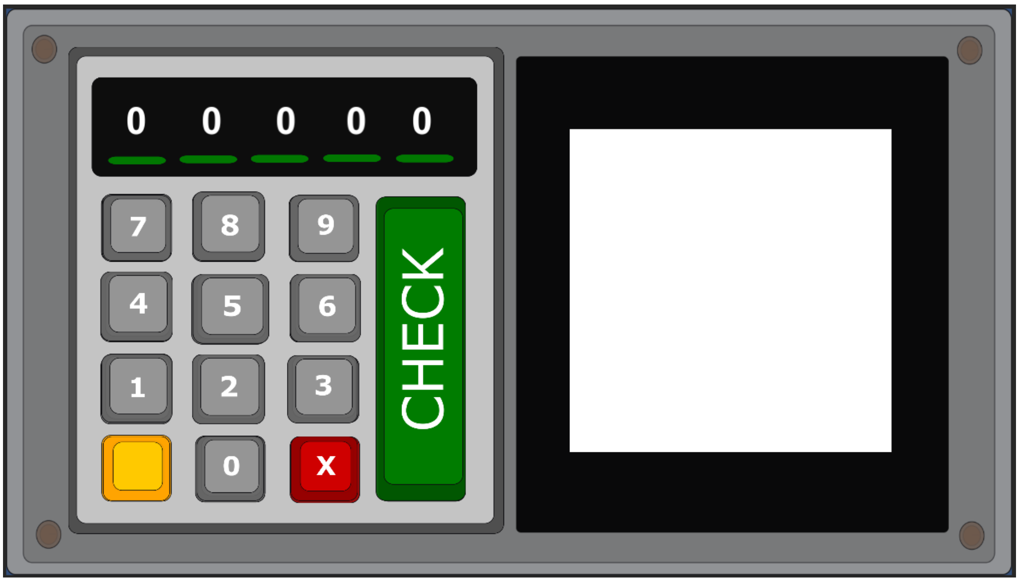
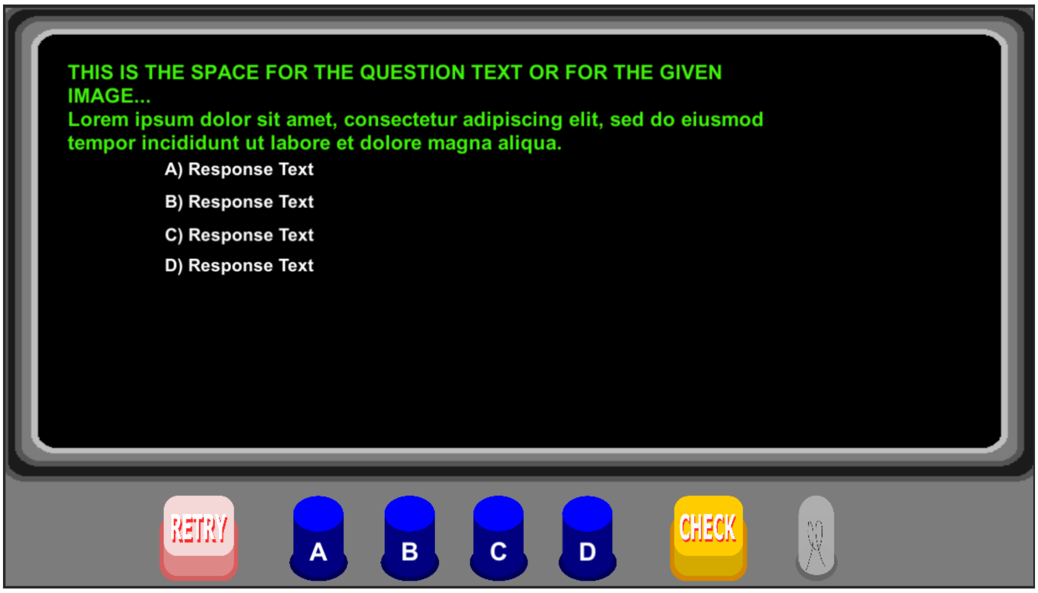


Figure 51: Calculator Scene with Image Slot

Multiple Choice Scene

For this scene, the player selects any number of buttons that correspond to the answers displayed on the screen. The player then selects the check button, and if the light glows green, the player is shown the success screen and taken back to the main map. If the player fails, the light glows red and the player can select the retry button to reset the scene, with each reset adding to the total number of attempts for the question.



THIS IS THE SPACE FOR THE QUESTION TEXT OR FOR THE GIVEN IMAGE...
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

A) Response Text
B) Response Text
C) Response Text
D) Response Text

Figure 52: Multiple Choice Scene

Category Scene

For this scene, the player drags the different tiles into one of the two categories. These can be any arbitrary split between two categories as defined by the professor in the question box (i.e. True/False, Static/Dynamic, Fruit/Veg, etc.). Each of these tiles can be either an image or text. When the player has placed all the tiles, they select the check button, which compares the position of the tile to an array of Boolean values that determines the category. If the light glows green, the player is shown the success screen and taken back to the main map. If the player fails, the light glows red and the player can select the retry button to reset the scene, with each reset adding to the total number of attempts for the question.

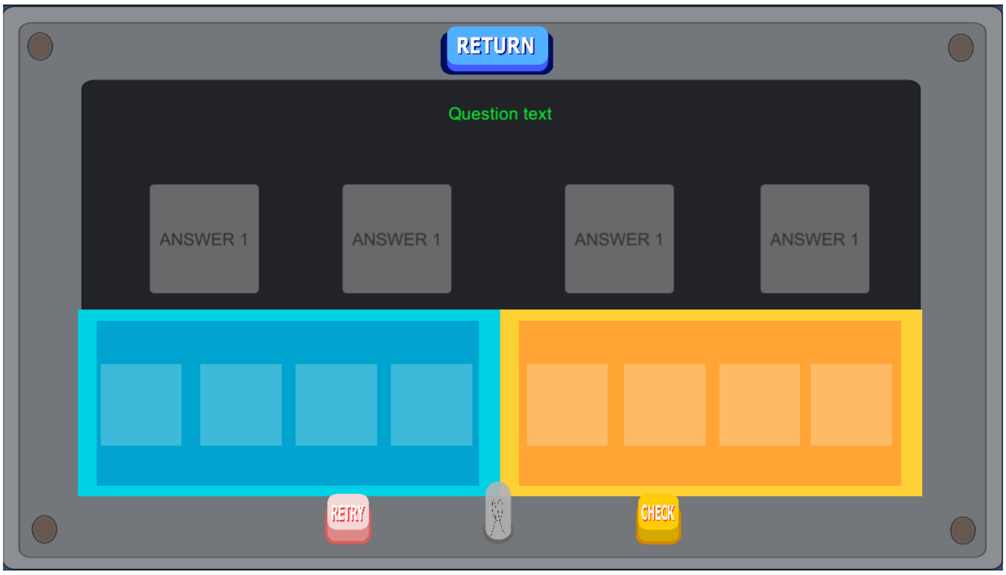


Figure 53: Category Scene

Order Scene

For this scene, the player drags the different tiles into the ordered slots. Each of these tiles can be either an image or text. When the player has placed all the tiles, they select the check button, which compares the position of the tile to an array of Integer values that determine its order. If the tiles are correctly ordered, when the player selects the check button, the slot backgrounds glow green, and the player is shown the success screen. If the player fails, the slot backgrounds glow red and the player can select the retry button to reset the scene, with each reset adding to the total number of attempts for the question.

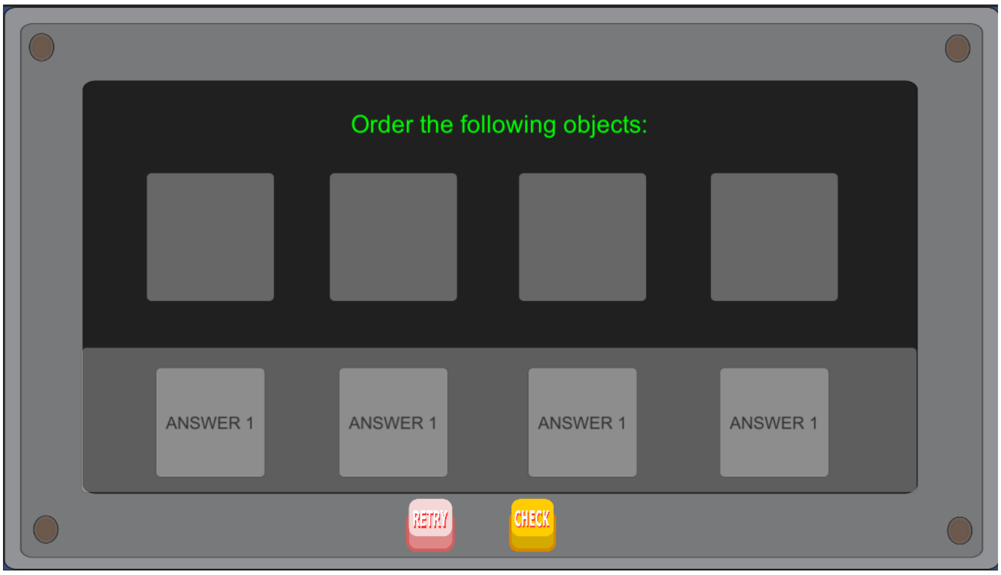


Figure 54: Order Scene

4 Tests

The tests were designed as acceptance tests for certain aspects of program functionality. The tests were performed at **runtime** (**RT**) and fall under the following categories: **Audio** (**AA**), **Animation** (**AN**), **Camera** (**CM**), **Collisions** (**CO**), **Launch** (**LC**), **Management** (**MM**), **Object** (**OB**), **Player** (**PL**), **Task** (**TK**), and **Team** (**TM**).

Test Identifier	**Description**	**Classification**
RT-AA01	Audio played during menu scene.	**PENDING**
RT-AA02	Background audio played during gameplay.	PENDING
RT-AN01	Player walks up, right, left, down, with left and right animation.	ACHIEVED
RT-AN02	Task button changes to pressed button sprite when touched.	ACHIEVED
RT-AN03	Objects change to highlighted sprite when it comes in contact with a player with the same task ID in their task list.	ACHIEVED
RT-CM01	Camera follows player who initiated game instance.	ACHIEVED
RT-CM02	Camera captures question screens fully.	ACHIEVED
RT-CO01	Object colliders trigger ‘Start Task’ button.	ACHIEVED
RT-CO02	Player collides with walls and objects.	ACHIEVED
RT-CO03	Player collides with other players.	ACHIEVED
RT-CO04	Player sprite maintains visual perspective when it collides with objects.	ACHIEVED

RT-CO05	Object collider disabled once the task has been completed.	ACHIEVED
RT-LC01	Object tasks are assigned on game launch.	ACHIEVED
RT-LC02	Players are assigned tasks on game launch.	ACHIEVED
RT-LC03	Object task dictionary is created on game launch to map objects to tasks.	ACHIEVED
RT-LC04	Player tasks appear on task board.	ACHIEVED
RT-MM01	Task completion is tracked globally.	ACHIEVED
RT-MM02	Completed tasks are displayed in green on the player’s task board.	ACHIEVED
RT-MM03	Incomplete tasks are displayed in red on the player’s task board.	ACHIEVED
RT-MM04	Players positions are tracked globally.	PENDING
RT-MM05	Scene switcher changes the scene based on the type of question.	ACHIEVED
RT-MM06	Scene switcher returns player to main map.	ACHIEVED
RT-MM07	Objects are assigned their original tasks after the scene switch.	ACHIEVED
RT-MM08	Game finishes when a team completes all the required tasks.	ACHIEVED
RT-MM09	Game is timed.	PENDING
RT-PL01	Player moves using **AWSD** keys.	ACHIEVED
RT-PL02	Player task board opens using **t** key.	ACHIEVED
RT-PL03	Player tasks are maintained through scene switch.	ACHIEVED
RT-PL04	Player task tries are recorded.	ACHIEVED
RT-PL05	Player task completion is recorded.	ACHIEVED
RT-PL06	Player task board shows required number of tasks (4).	ACHIEVED
RT-PL07	Player ID is not modified through scene switch.	ACHIEVED
RT-TK01	Multiple choice scene maintains selection pressed.	ACHIEVED
RT-TK02	Multiple choice scene allows for multiple selection.	ACHIEVED
RT-TK03	Multiple choice scene compares user selection with response correctly.	ACHIEVED
RT-TK04	Multiple choice scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK05	Multiple choice scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK06	Multiple choice scene allows users to try again when they answer incorrectly.	ACHIEVED
RT-TK07	Multiple choice scene contains a return button for users to abandon the incomplete task.	ACHIEVED
RT-TK08	Multiple choice scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK09	Multiple choice scene shows question and multiple answers from task details.	ACHIEVED
RT-TK10	Categories scene pieces drag and drop correctly.	ACHIEVED

RT-TK11	Categories scene pieces snap into place in category box.	ACHIEVED
RT-TK12	Categories scene compares user selection with response correctly.	ACHIEVED
RT-TK13	Categories scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK14	Categories scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK15	Categories scene allows users to try again when they answer incorrectly.	ACHIEVED
RT-TK16	Categories scene contains a return button for users to abandon the incomplete task.	ACHIEVED
RT-TK17	Categories scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK18	Categories scene shows question and multiple answers from task details.	ACHIEVED
RT-TK19	Categories scene pieces appear as picture or text.	PENDING
RT-TK20	Drag N Drop scene pieces drag and drop correctly.	ACHIEVED
RT-TK21	Drag N Drop scene pieces snap into place in category box.	ACHIEVED
RT-TK22	Drag N Drop scene compares user selection with response correctly.	ACHIEVED
RT-TK23	Drag N Drop scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK24	Drag N Drop scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK25	Drag N Drop scene allows users to try again when they answer incorrectly.	ACHIEVED
RT-TK26	Drag N Drop scene contains a return button for users to abandon the incomplete task.	ACHIEVED
RT-TK27	Drag N Drop scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK28	Drag N Drop scene shows question and multiple answers from task details.	ACHIEVED
RT-TK29	Drag N Drop scene pieces appear as picture or text.	PENDING
RT-TK30	Calculator scene buttons are pressable.	ACHIEVED
RT-TK31	Calculator scene prohibits more than 5 characters in answer.	ACHIEVED
RT-TK32	Calculator scene compares user selection with response correctly.	ACHIEVED
RT-TK33	Calculator scene distinguishes between correct and incorrect answers when the user presses the check button.	ACHIEVED
RT-TK34	Calculator scene shows success panel when the user answers correctly.	ACHIEVED
RT-TK35	Calculator scene allows users to try again when they answer incorrectly.	ACHIEVED

RT-TK36	Calculator scene contains a return button for users to abandon the incomplete task.	ACHIEVED
RT-TK37	Calculator scene switches to the main map scene while maintaining player task statistics.	ACHIEVED
RT-TK38	Calculator scene shows question from task details.	ACHIEVED
RT-TK39	Calculator scene question appears as picture or text.	PENDING
RT-TM01	Teams are automatically assigned with even numbers of players on each team.	PENDING
RT-TM02	Team progress is tracked in progress bar.	ACHIEVED
RT-TM03	Opposition progress is tracked in progress bar.	ACHIEVED

The following table details all the bugs that were found during the testing phase and the level of severity of the bug.

FUNCTIONALITY	**DESCRIPTION**	**CLASSIFICATION**
Animation	The player does not stop moving despite the input being null.	**BUG – Fixed 06.05.21** The player animator uses additional variables to monitor movement.
Animation / Collisions	Objects don’t highlight when the player enters the object trigger range.	**BUG – Fixed 28.04.21** The collider was updated to track player ID and task ID.
Collisions	The object colliders do not trigger the ‘Start Task’ button.	**ERROR – Fixed 28.04.21** A new controller was created to register the collisions.
Collisions	The player sprite goes over the objects, but the perspective is incorrect.	**BUG – Fixed 26.04.21** The player collider was expanded to avoid perspective errors.
Player	The player needs to open personal task board by selecting **t** key, but key does not register on PlayerTaskController.	**ERROR – Fixed 28.04.21** The game controller was modified to register player input on keyboard (**t**).
Tasks	The calculator scene does not register more than four digits and the number that is calculated is not in the correct order.	**ERROR – Fixed 28.04.21** The loops for the text boxes were not properly instantiated.
Tasks	The image swap mechanism allows for images to be used in place of both questions and answers.	**ERROR – Fixed 11.05.21** The images folder was created, and all images must be placed in it to be able to be accessed. (NOTE: This will be tracked by server).
Tasks	Tasks do not survive transition from one scene to another.	**ERROR – Fixed 06.05.21** The task manager was

		included in the game manager, which does not get destroyed when the scenes are switched.
Tasks	Player tasks not assigned upon game initiation.	**ERROR – Fixed 06.05.21** The game controller instantiates tasks and assigns them to the task controller singleton object.
Tasks	Player tasks are reassigned on switching scenes.	**ERROR – Fixed 09.05.21** The player tasks were written to local files and read from them once assigned. (NOTE: Server should monitor this).
Tasks	Player tasks do not register as completed on task board.	**BUG – Fixed 09.05.21** The task completion is now tracked by the task controller through IDs rather than the scene switcher.
Tasks	Images do not appear in image swap containers.	**ERROR – Unsolved**
Tasks	DragNDrop images grow if return is selected multiple times.	**BUG – Fixed 05.06.21** Return button disabled before selection is checked.
Teams	Player task completion not registered on team task completion bar.	**ERROR – Fixed 15.05.21** The team uses player tasks and player task completion marker.
Teams	Player not assigned to teams.	**ERROR – Fixed 15.05.21** Game controller assigns players upon creation to team.
Scene	Endgame scene does not register scene change.	**ERROR – Fixed 10.06.21** Removed unregistrable variables.
Scene	End panel does not appear when team completes tasks.	**ERROR – Fixed 10.06.21** Added team name to the sceneswitch module to register team name upon completion.

C User Manual

SpaceRaceEdu

An Educational Odyssey

28/04/2021

User Manual

Escuela Politécnica Superior,
Universidad Autónoma de Madrid
May 2021

1 Introduction

This document describes the use cases and application for the different types of users: **students** and **teachers**. The sections described in this document are organized as follows:

Sections

- Installation
- Objective
- Controls
- Question Input

2 Installation

The requirements for the software can be found in the table below.

Table 7: Software Requirements

Operating System	Windows	macOS	Linux
OS Version	Windows 7 (SP1+) y Windows 10	Sierra 10.12+	Ubuntu 16.04, 18.04, 20.04
CPU	x86, x64 architecture with SSE2 instruction support.	X64 architecture with SSE2.	X64 architecture with SSE2 set instruction support.
Graphics API	DX10, DX11, DX12 capable .	Metal capable Intel and AMD GPUs	OpenGL 3.2+, Vulkan capable.

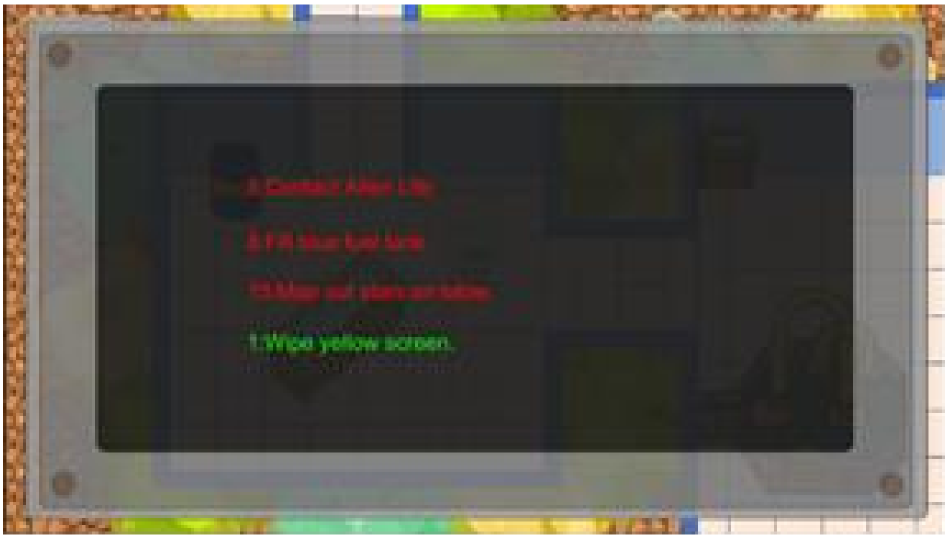
The Beta testing game software is available free, online at the following link:

https://drive.google.com/drive/folders/1D6xgVdOwgiu51CXZPsD0WLTBfD4wfwYb?usp=sharing

The download should be unzipped and directly installed on the computer. Once there, the teacher will be able to access the question configuration files in the AppData folder.

3 Objective

The objective of the game from the student's perspective is to complete the tasks assigned to their player. These tasks can be found by selecting the **t** button on the keyboard once in the game (see Fig. 39). These tasks can be in the form of multiple-choice questions, order questions, categorization questions, and numerical questions. Once these tasks are completed, the player must wait for the rest of their team to complete their tasks. When an entire team has completed their tasks, the game finishes.



2.Contact Alien Life

3.Fill blue fuel tank

15.Map out stars on table

1.Wipe yellow screen.

Figure 55: Example Task Board

4 Controls

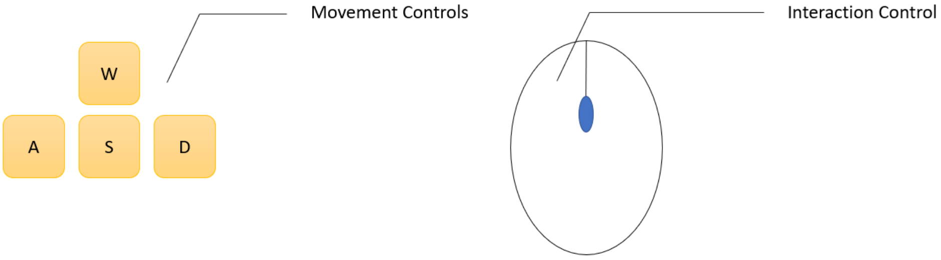


Figure 56: Game Controls

The controls are shown in the diagram above. The user can move with either the arrow keys or the AWSD keys. To interact with the objects, the user selects the task button with the cursor. Each task scene has a different mode of interaction described below.

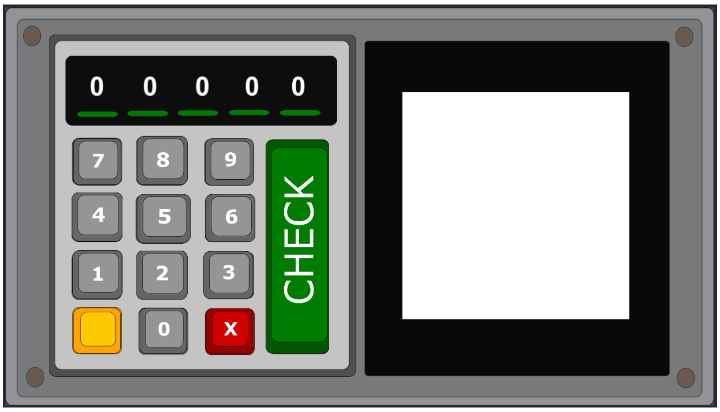
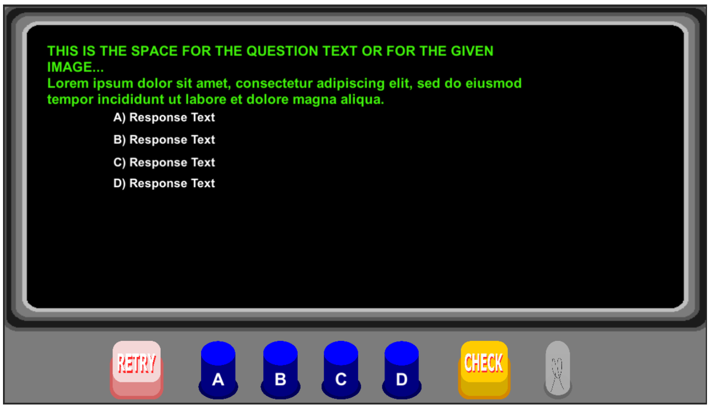


Figure 57: Calculator Task

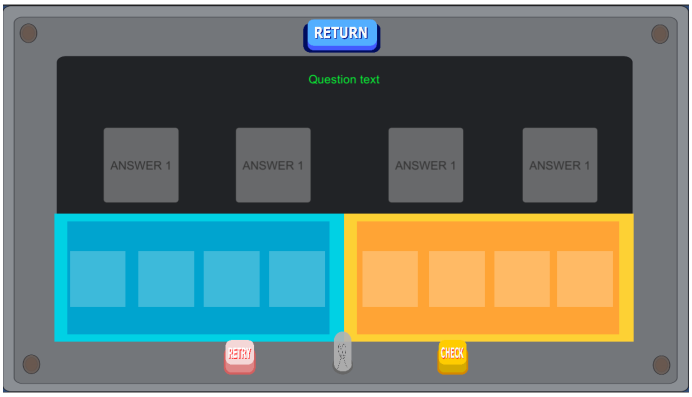
To interact with the calculator scene (Fig. 51), with the mouse the user selects the buttons on the calculator that comprise the final answer. Selecting the digits from left to right. i.e. if the answer is 735, then the user would select buttons 7, 3, and 5 in that order, and then they would hit the **check** button. If the answer is correct, the digits are written in green and the user can select the **return** button to return to the main map. If the answer is incorrect, the digits are written in red and the user must select the yellow button (**retry**) to attempt the question again. The red **X** button clears the input in the event that the user made an error they wish to correct. The calculator will not accept more than 5 digits.

To interact with the multiple choice scene (Fig. 52), with the mouse, the user selects the blue letter buttons that correspond with the correct answer(s). Finally they select the **check** button to grade their attempt. If the answer is correct, the light on the right side of the screen glows green and the user may select the **return** button that appears to return to the main map scene. If the answer is incorrect, the light on the right side of the screen glows red and the user must press the **retry** button to make a new attempt.



THIS IS THE SPACE FOR THE QUESTION TEXT OR FOR THE GIVEN IMAGE...
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
A) Response Text
B) Response Text
C) Response Text
D) Response Text

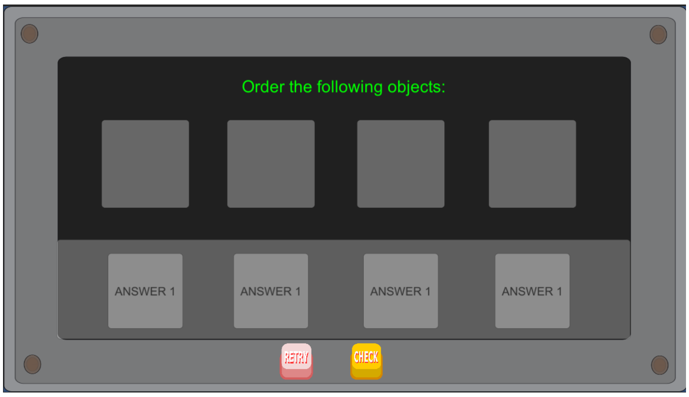
Figure 58: Multiple Choice Task



Question text

Figure 59: Categories Task

To interact with the categories scene (Fig. 53), with the mouse the user selects the answer from the top half of the screen and drags it to the appropriate category box, into one of the slots available. When finished, the user selects the **check** button to grade their attempt. If the answer is correct, the light on the bottom of the screen glows green and the user may select the **return** button that appears to return to the main map scene. If the answer is incorrect, the light on the bottom of the screen glows red and the user must press the **retry** button to make a new attempt.



Order the following objects:

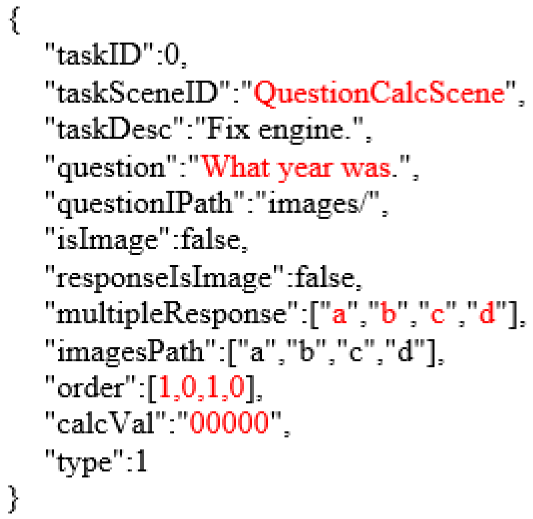
Figure 60: Ordering Scene

To interact with the ordering scene (Fig. 54), with the mouse the user selects and drags the bottom squares into the top slots. When finished, the user selects the **check** button to grade their attempt. If the answer is correct, the squares on the top of the screen glows green and the user may select the **return** button that appears to return to the main map scene. If the answer is incorrect, the squares on the top of the screen glows red and the user must press the **retry** button to make a new attempt.

5 Question Input

ALPHA VERSION

To input the questions into the question bank in the Alpha version of the game, the teacher must access the Pl80ium folder in C://Users/<user>/AppData/LocalLow/. Here, the teacher can fill the JSON files with the questions and answers they require. The modifiable variables are listed in red below:



```
{
    "taskID":0,
    "taskSceneID":"QuestionCalcScene",
    "taskDesc":"Fix engine.",
    "question":"What year was.",
    "questionIPath":"images/",
    "isImage":false,
    "responseIsImage":false,
    "multipleResponse":["a","b","c","d"],
    "imagesPath":["a","b","c","d"],
    "order":[1,0,1,0],
    "calcVal":"00000",
    "type":1
}
```

The options for the **taskSceneID** variable are:

- **QuestionCalcScene**
- **QuestionButtonScene**
- **QuestionCategoriesScene**
- **QuestionDragNDropScene**

The order variable represents **true** or **false** when referring to the **QuestoinButtonScene** and the **QuestionCategoriesScene**, and takes only integer values of [0,1]. When referring to the **QuestionDragNDropScene**, this variable takes values from [0,3].

Note: Basic knowledge of JSON is required.

BETA VERSION

To input the questions into the question bank in the Beta version of the game, the teacher selects the **Create Question Bank** button on the beginning screen. If there are no questions available, the **Join Game** button is automatically disabled. The teacher is then prompted to input questions and values for each answer, as seen below in Figure 45.

Once the question input is finished, the teacher selects the **Return** button at the bottom of the screen, which takes them back to the initial screen. From there, the **Join Game** button is enabled and can be selected to begin the game.

D Questionnaire Results

The following table presents the results compiled as of 15/06/2021 with regards to the student opinion questionnaire.

Table 8: Student Opinion Questionnaire Results

How does playing the game compare to taking an actual test?	Do you feel like you would get a better grade on an exam styled this way rather than on a traditional exam?	How would you rate the gameplay?	How intuitive is the game?	How does being able to play with teammates help/hinder the learning experience?	How does being able to compete against classmates help/hinder the learning experience?	How comfortable would you be playing this game in the classroom?	How would you like to see this game used in the classroom?	If given the chance, would you use this software to study at home?
Easier	Maybe	4	4	Your friends can help you with difficult questions	If you are competitive, this game is great	5	For revisions	Maybe
Easier	Yes	3	4	Makes it more fun and exciting	Competitive	4	For maths	Yes
The same	Yes	4	4	Help	Help	4	Interactive teaching	Yes
The same	Yes	4	4	Help	Help	4	Interactive teaching	Yes
Easier	Yes	3	5	Playing with teammates always helps, because we can "teach" and try to exchange ideas about the issues.	It always helps because we have a healthy competitiveness	4	Yes !	Maybe
Easier	Yes	4	5	They can help each other in certain steps of the game, this way promoting the team work.	Usually when we play video games we always want to be better than the others that we are competing with. In this case, classmates would compete with each other, this way encouraging them to do better and improve their way of thinking.	5	This would be an interesting and probably well received way of learning to the students.	Yes
Easier	Maybe	4	4	Its more funny	Be the best is always the main priority when playing games	5		Yes
Easier	Maybe	4	5	Allows for a stress-free experience while doing the work given	It makes it more competitive	5	During extra point oriented activities	Yes
The same	Yes	4	4	You could divide the different tasks between your teammates, using each other's strengths better if you know the tasks before doing them.	Would give some extra motivation to the students, since it'll be competitive and fun. In a similar way as kahoot tests i would say.	4	Perhaps not for the main tests, but for end of the week tests that will keep the students motivated to forget about the subject until the main exam.	Maybe
The same	No	5	4	Faster or slower students may get discouraged in a group dynamic if the group's pace doesn't match their own.	Competition can motivate some students to do better, especially in a Game-Style setting. Other students may get discouraged at the idea of competition.	5	As practice leading up to tests, warm up before class	Maybe
Easier	Yes	4	4	Helps	Helps	5	Instead of tests	Yes
Easier	Yes	4	5	Its more fun	I like it	5	To study new things	Yes
Easier	Yes	5	5	Helps.	Help.	5	Regularly.	Yes
Easier	Yes	4	5	Playing with teammates	Helps learning as the competitive	5	As study methods, practice quizzes and tests	Yes

				allows me to see things not only through my perspective but my teammates and therefore could teach me new things or teach me new ways to solve problems. I think working together is always beneficial as you are able to pick up traits and learn from each other	feeling at game level Is more exciting, so would make me try harder and work/ study harder			
Easier	Yes	4	4	It allows you to focus more, and be less anxious when taking the test, because games are generally more relaxing, and are used for leisure, while a test is something that's seen as something you have to prepare for and you can't fool around with a test.	It shows that you've been studying, and putting in the time and effort where you're classmates haven't, so it puts you a step up.	4	For exams	Yes
Easier	Yes	1	5	The mad banter from the bois dem would be detrimental to the acquisition of academic knowledge 🤓	Motivated competition = more learning	5	Maybe as a pop quiz replacement or for test revision	No
Easier	Yes	5	4			5		Yes
Easier	Maybe	3	3	Team work always helps	It makes it more competitive	4	In a big screen	Yes
Easier	Maybe	5	5	It enhances the learning experience	Again, It enhances the learning experience	3	The games allows for a more "relaxed" way of teaching/learning, opposed to the more traditional way, might help those with different levels of concentration/assimilation	Yes
Harder	Maybe	3	4		Would probably help the learning experience, due to the competitiveness it would provide.	4	Used on boring subjects, to Spike the interest.	Maybe
Easier	Yes	4	4	helps	it helps to fire up your competitiveness	3	sure	Maybe
Easier	Yes	5	5			5		Yes
Easier	Maybe	4	5	Help as it's more fun	Help as it's more competitive	4	To facilitate review / group study	Maybe
Easier	Yes	4	4	It allows you to focus more, and be less anxious when taking the test, because games are generally more relaxing, and are used for leisure, while a test is something that's seen as something you have to prepare for and you can't fool around with a test.	It shows that you've been studying, and putting in the time and effort where you're classmates haven't, so it puts you a step up.	4	For exams	Yes

The following table presents the results compiled as of 15/06/21 with regards to the teacher opinion questionnaire.

Table 9: Teacher Opinion Questionnaire Results

Do your students have access to computers in class?	Would gamified education be easily applied to your subject?	Regarding the video provided in the description, how applicable is this software to your classes?	How so?	In your opinion, would this encourage teamwork in your class?	How do you see your students reacting to this software?	Would you use this software for exams or reviews?	Would you give your students the option to use this software outside of the classroom?	Do you see this software providing benefits to the learning process?	Please add any thoughts/opinions/comments/criticisms below.
Yes	Maybe	Somewhat	A primera vista encuentro dos formas: 1) como ejercicio de clase para repasar contenidos de cara al examen final, 2) como herramienta de autoestudio para los estudiantes durante el curso.	Yes	Creo que los estudiantes reaccionarían mostrando un mayor interés, distensión y participación que en una clase más convencional.	No	Yes	Yes	Pros: - Me parece una herramienta muy potente para el autoestudio que mejora en todos los sentido el estudio con libro/apuntes y la realización de cuestionarios. - Puede ser un buen ejercicio en grupos de un cierto nivel y tamaño para tener clases más distendidas en las que los estudiantes presten más atención. Contras: - En el caso del autoestudio puede ser complicado que los estudiantes se organicen para hacer una partida con varios equipos de varios jugadores. - La adaptación a ciertas asignaturas de ciertos niveles educativos puede ser complicada e incluso requerir cambios en el propio software.
Yes	Yes	Somewhat		Yes	Excited	Maybe	Yes	Yes	
Yes	Maybe	Somewhat	Through questionnaires related to the topics, working in groups and competing with each other	Yes	Pending evaluation. But I trust that it is well received	No	Yes	Yes	Thanks
Yes	Yes	Very	Vocabulary can be made into multiple choice questions. The calculator doesn't apply.	Yes	Positively	Yes	Yes	Yes	N/a
Yes	Yes	Very	I.T. related questions can be made into the multiple choice format. Ordering of program code can be done in the ordering section. True or False can be used for binary outputs or code outputs. Calculator can be used for program outputs	Yes	Positively. They prefer games to normal lectures.	Yes	Yes	Yes	You should include a short answer section.
Yes	Yes	Somewhat	I see that the software is more suitable to primary school and not as much for university level degrees.	Yes	The dynamic of the game seems too easy for university students. The questions, I guess, could be adapted to university level degrees.	Maybe	Yes	Yes	
Yes	Maybe	Not at all		Maybe		Maybe	Yes	Yes	
Yes	Maybe	Somewhat	Preguntas que conlleven un cálculo matemático serían más fácil de hacer con respecto a preguntas que conlleven	No	No sé. Tendría que ver el juego en más detalle. Si conlleva cierta dificultad y varias funcionalidades puede que les	No	Yes	Maybe	

			la ordenación de respuestas.		llame la atención.				
Yes	Maybe	Very		Maybe	I don't know	Maybe	Maybe	Maybe	
Yes	Yes	Very	Se podría adaptar a cuestiones de morfología o sintaxis e incluso relaciones de movimientos literarios con sus autores.	Yes	La respuesta de los alumnos ante los juegos es siempre positiva. Suelen ser competitivos y es una forma dinámica y divertida. Así que su reacción sería siempre positiva.	No	Yes	Yes	Mi experiencia con la implementación de juegos y actividades interactivas en clase es positiva, así que os animo a seguir con este proyecto.
No	Yes	Somewhat	Se podría usar en el caso en que los alumnos usaran su dispositivo móvil.	Maybe	Seguro que les gustaría, al menos al principio. Después de algunas clases, perdería emoción. Ya les ocurrió con otras herramientas como Kahoot.	Maybe	Yes	Yes	

E Initial Design Sketches

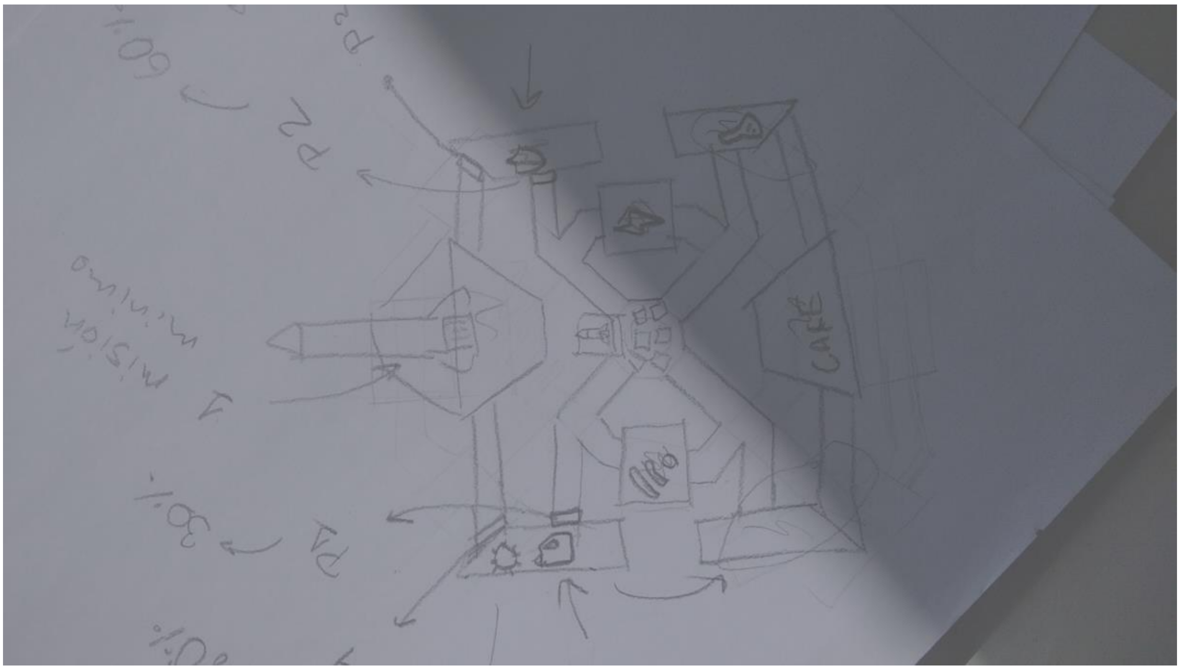


Figure 61: Map Layout

Figure 62: Player

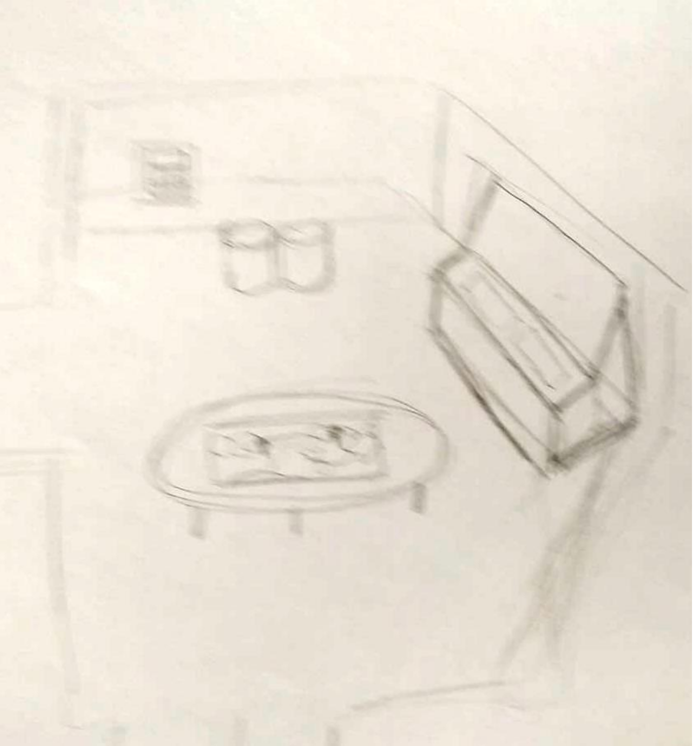


Figure 63: Map Room

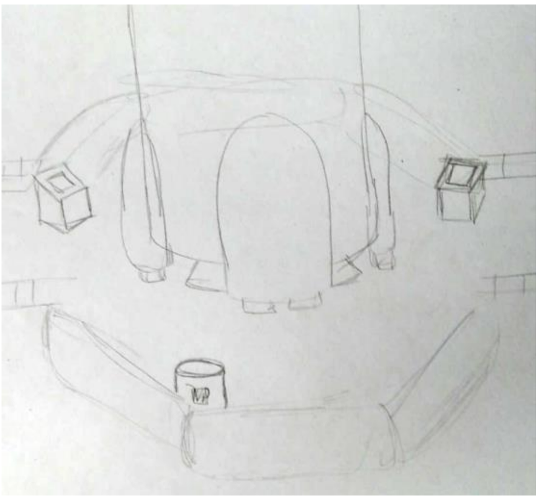


Figure 64: Rocket Launchpad

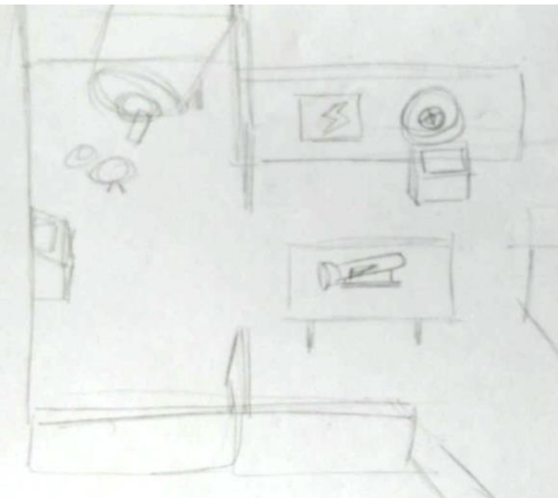


Figure 65: Astronomy Room

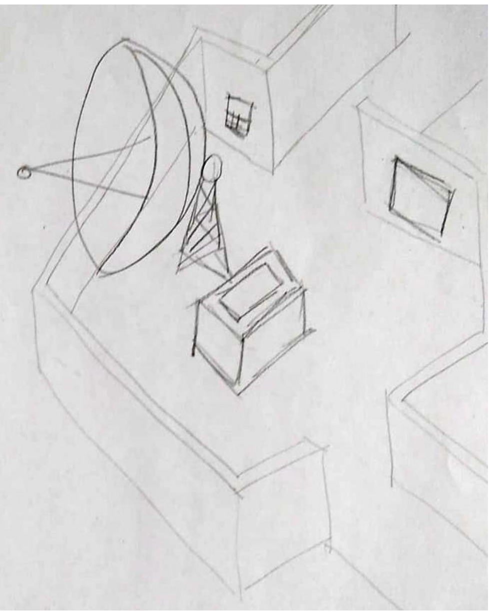


Figure 66: Satellite Room

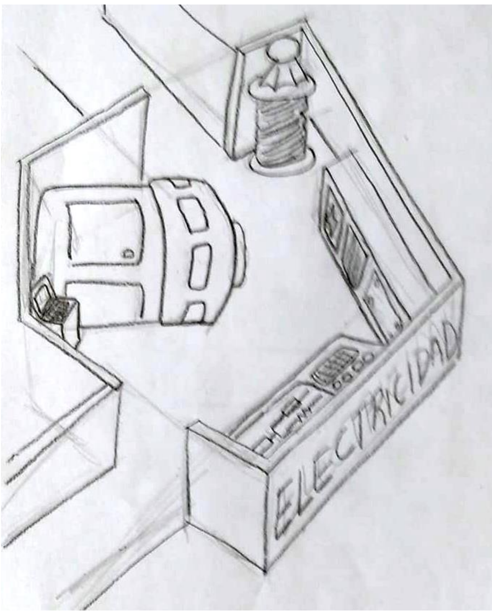


Figure 67: Electric Room

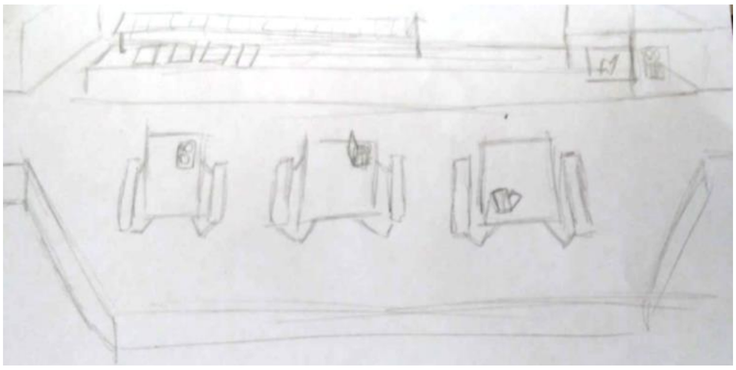


Figure 68: Cafeteria

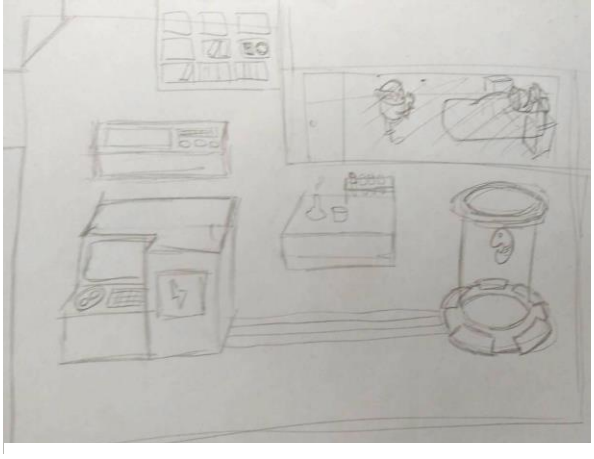


Figure 69: Experiment Room

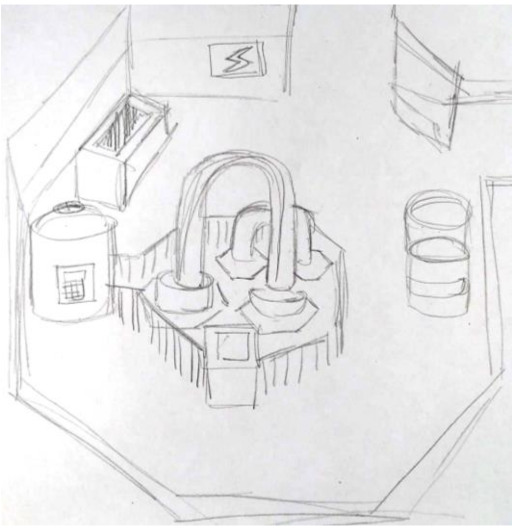


Figure 70: Pump Room