UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Design and implementation of a learning game

Diploma Thesis

Efstathia Koutougera

Supervisor: Hariklia Tsalapata

Volos, 2022

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Design and implementation of a learning game

Diploma Thesis

Efstathia Koutougera

Supervisor: Hariklia Tsalapata

Volos, 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδίαση και υλοποίηση μαθησιακού παιχνιδιού

Διπλωματική εργασία

Ευσταθία Κουτούγερα

Επιβλέπουσα: Χαρίκλεια Τσαλαπάτα

Βόλος, 2022

Approved by the Examination Committee:

Supervisor **Hariklia Tsalapata**

Laboratory Teaching Staff, Department of Electrical and Computer Engineering, University of Thessaly

Member **Aspassia Daskalopulu**

Assistant Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Eleni Tousidou**

Laboratory Teaching Staff, Department of Electrical and Computer Engineering, University of Thessaly

Date of approval: 24-06-2022

Acknowledgements

I would like to thank my supervisor, Dr. Hariklia Tsalapata for her help and valuable assistance in implementing this thesis. Moreover, I would like to thank Dr. Aspassia Daskalopulu and Dr. Eleni Tousidou for their useful comments and for participating in the examination committee. Additionally, I would like to thank Aris, Eleni, Hermes, and Ioli for providing me with their drawings to make the game look stunning.

Last but not least, I would like to thank my family and my friends for always being there for me throughout my studies. Their support played a vital role in composing this thesis.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Efstathia Koutougera

Abstract

Nowadays, technology has penetrated every area of life, so it is essential for education to follow it. In addition, digital games attract almost everyone, especially children. At the same time, various methods are gradually being adopted by teachers and educational systems, which put the student at the center of the educational process. Active learning is becoming more and more popular, and books are no longer the only means of acquiring knowledge. A new method of learning is educational games, where students learn in an engaging way, assimilate the curriculum and have fun at the same time. For this reason, a math learning game for elementary school children was designed and implemented using the Unity platform.

Περίληψη

Στην εποχή μας, η τεχνολογία έχει εισχωρήσει σε κάθε τομέα της ζωής, επομένως είναι αναγκαίο να την ακολουθήσει και η εκπαίδευση. Επιπλέον τα ψηφιακά παιχνίδια ελκύουν σχεδόν όλους τους ανθρώπους, πόσο μάλλον τα παιδιά. Παράλληλα, έχουν αρχίσει να υιοθετούνται διάφορες μέθοδοι, από καθηγητές και εκπαιδευτικά συστήματα, που θέτουν το μαθητή στο επίκεντρο της εκπαιδευτικής διαδικασίας. Η ενεργή μάθηση γίνεται όλο και πιο δημοφιλής και τα βιβλία παύουν να είναι το μοναδικό μέσο απόκτησης γνώσεων. Ένας νέος τρόπος εκμάθησης είναι τα εκπαιδευτικά παιχνίδια, όπου οι μαθητές με ελκυστικό τρόπο μαθαίνουν, αφομοιώνουν τη διδακτέα ύλη και διασκεδάζουν ταυτόχρονα. Για τον λόγο αυτό, σχεδιάστηκε και υλοποιήθηκε ένα μαθησιακό παιχνίδι μαθηματικών για παιδιά του δημοτικού χρησιμοποιώντας την πλατφόρμα Unity.

Contents

List of Figures

List of Tables

”Play is our brain’s favorite way of learning things.”

-Diane Ackerman, Deep Play

Chapter 1

Introduction

In the 21st century, people are into technology more than ever in the past, and technology is playing a more vital role in society than ever before. It affects people's everyday life as it improves every aspect rapidly. Everyone nowadays takes technology for granted, and children do not know how life is without the internet, smartphones, tablets, etc.

Internet and smartphones are necessary goods these days. To confirm this fact, as presented in the Statista survey [1], in 2011, only 31,8% of the global population had access to the internet, with the developed countries possessing a percentage of 67,6%. On the contrary, one decade after, in 2021, it is estimated that 63% of the worldwide population access the web. This percentage reaches up to 90% in developed countries. As far as smartphone subscriptions are concerned [2], in 2016, there were almost 3,5 billion. After a five-year period, in 2021, these subscriptions reached almost 6,26% billion globally, thus holding a percentage nearly 80% of the international population. So, it is obvious that the requirement for technological goods has changed dramatically, and these goods are important and vital for current communities.

As information and technology are a part of modern society, education should follow as well. The kids are familiar with the digital world as they are born within it. At home, they watch videos on the internet, play games, communicate through video calls, and use social media apps from an early age. The time spent using these devices differs between each child and family, but undoubtedly they use them sometimes during the week. On the other hand, the school program has not embraced technology enough, and the gap between school and home should be bridged, and schools must start using it more dynamically. Generally, the educational system should be redesigned to meet the current altered needs. According to a survey [3], in 2019, around 57% of the students in the United States use digital learning tools on a daily basis. Nowadays, books are not the only means of learning, but videos, games, podcasts, et cetera, can support education. Moreover, the collaboration of teachers, parents, and students can be more instant and efficient.

People, and especially children, love playing video games. Today, a video game can be played on multiple machines, from smartphones to consoles specially designed for playing video games. It is worth mentioning that, the global market size of Video Games is around US$155.5 billion in 2021, with a forecast of reaching US$ 240 billion in 2026 [4]. Hence, it is obvious that the game industry is becoming larger over the years, as more and more people want to play digital games, and at the same time, there is a demand for game variety as the audience is getting bigger and having different needs and desires.

The combination of technology, the digital era, education, and video games leads to the development of novel digital games and their use in the educational process.

1.1 Aim of the thesis

The purpose of this thesis is to design and build a digital learning game. This game combines the excitement of playing with the education process, particularly with algebra learning. Its target group is elementary school students, emphasizing to the first grades of primary school. The used technology for the construction of the game, provides its appliance on multiple platforms and devices.

1.2 Contribution

This thesis's contribution is the development of a new learning mathematical digital game for primary school students, intergrading multiple elements used in games and gamification. Specifically, achievements, avatars, chance, content unlocking, leaderboards, levels, points and quests are used to keep the pupils engaged and enjoy both the game and the learning process.

1.3 Structure of the thesis

The thesis is divided into six chapters. The first chapter is the introduction, describing the aim of the thesis. The second chapter refers to the learning methodologies and especially to active learning. The third chapter presents the learning games, their definitions, and the elements a digital game may have and illustrates other learning games that are currently used. The fourth chapter introduces Unity, the platform that was used for the development of the game. The fifth chapter presents the game and its development. Finally, the last chapter is the conclusion, summarizing the thesis and presenting the future work.

Chapter 2

Learning Methodologies and Active Learning

”Learning is defined as a process that brings together personal and environmental experiences and influences for acquiring, enriching or modifying one’s knowledge, skills, values, attitudes, behaviour and world views.” [5]

2.1 Teaching Methods

Teaching theories can be classified into four categories based on two main parameters: a teacher-centered approach versus the use of a student-centered approach and high-tech material use versus low-tech material use.[6]

Teacher-Centered Approach to Learning

Instructors are the primary authority figure in the teacher-centered instruction model. Pupils and students passively get information from their teachers through lectures and direct instruction, with a final goal of approving results from testing and assessment. Teaching and evaluation are seen as two isolated entities; student learning is estimated through objectively scored tests and examinations.

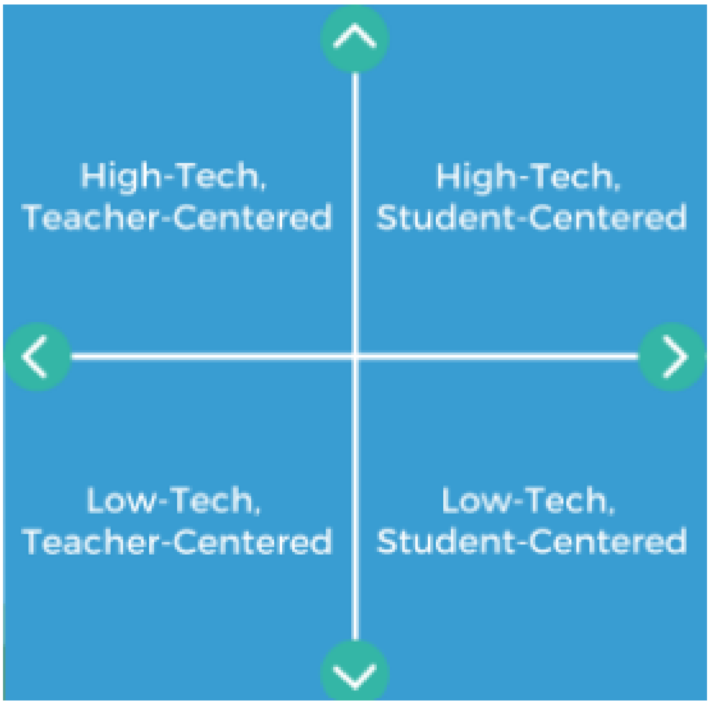


Figure 2.1: *Teacher-Student Centered and High-Low Tech Centered Teaching [6]*

Student-Centered Approach to Learning

Even though teachers are still an authority figure in a student-centered model, instructors and students play an equally active role within the learning process. The teacher works as a guide while the students play a more active role. The teacher's fundamental part is to coach and encourage student learning and the general comprehension of material. In addition, his role as pedagogue is to measure pupil learning through formal and informal evaluation forms, like group projects, student portfolios, and class participation. Education and evaluation are linked within the student-centered classroom, as student learning is constantly measured during instruction by the professor.

High Tech Approach to Learning

Progress in technology has impacted the education sector in the last few decades. The high-tech approach to learning uses different technology to support students' in-class learning. Many teachers use computers and tablets in the classroom, and some may use the internet to assign homework. The internet is additionally helpful in a classroom environment because it gives unlimited resources and information. Educators may also use the web to connect their pupils to individuals from around the globe.

Low Tech Approach to Learning

Despite the fact that innovation has transformed teaching, numerous teachers prefer to use a more conventional, low-tech approach to learning. Some learning styles demand a physical presence and interaction between the instructor and the student. Furthermore research has revealed that low-tech classrooms may boost learning. For instance, pupils who take handwritten notes have a better memory than students who take typed notes.

Teaching methodologies

The teachers can adopt one or multiple learning methodologies depending on whether they follow a student-centered or teacher-centered approach and to which extend technology contributes to the learning procedure, Figure 2.2.

Direct instruction is the general term that refers to the conventional educating technique that depends on straightforward teaching through lectures and teacher-led demonstrations. It uses passive learning, or the concept that ”the audience” can learn what they need to through listening and observing very detailed instructions. Teachers act as the exclusive provider of information, and beneath the explicit instruction model, instructors regularly use systematic, scripted lesson plans. Direct instruction programs contain precisely what the educator plans to present and the exercises students should complete for each phase of the lesson.

Flipped classroom represents the education system with students watching pre-recorded lessons at home and completing assignments at school, instead of attending lectures in the classroom and doing assignments at home. This approach allows the use of class time to explain contents in greater depth through activities oriented to understanding skills for real-life situations. Instructors who follow the flipped classroom model frequently record their instructional videos, but others use pre-made videos from online sources. [7]

Rather than attending lectures, students commit to hands-on physical exercises in the *kinesthetic learning* model. Kinesthetic learning, which values movement and

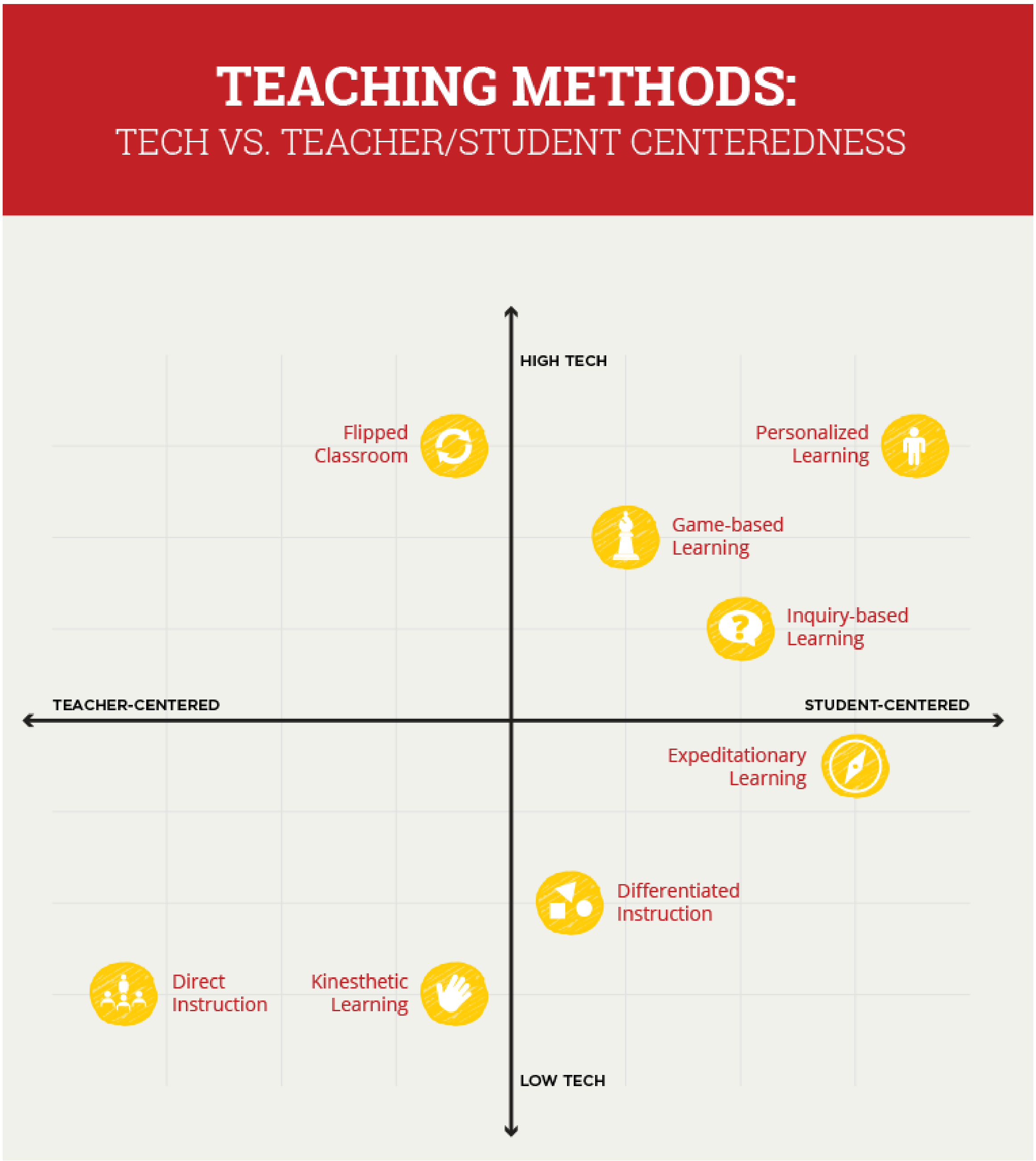


Figure 2.2: *Teaching Methods [6]*

imagination over technological skills, is most commonly used to increase conventional instruction types—the approach requiring students to do, make or construct something, which exercises various learning muscles. [8]

Differentiating instruction is the teacher's education practice to observe and comprehend the differences and similarities among the learners and use this knowledge to design the teaching. Depending on the students, the educator can teach in small groups or one-to-one sessions if needed.[9]

Inquiry-based learning is an educating process that casts an instructor as a supportive figure who gives direction and support for students throughout their learning procedure, instead of a sole authority figure. Teachers encourage learners to question and consider what they need to know about the world around them. Students at that point investigate their questions, find information and sources that illustrate key concepts, and solve problems they may experience along the way.

Expeditionary learning includes ”learning by doing” and taking part in a hands-on experience. Learners may participate in fieldwork, learning expeditions, projects, or case studies to apply knowledge learned in the classroom to the real world instead of learning through the virtual world.

In *personalized learning*, instructors urge pupils to pursue personalized, self-directed learning plans inspired by their particular interests and talents. Since evaluation is also custom-made to the person, students can progress at their own pace, moving forward or spending additional time as required. Educators offer a few conventional instructions and online material, whereas also constantly reviewing student progress and meeting with students to make any required changes to their learning objectives.

Everyone enjoys playing games, and the students, even more, so impressive improvement has been made in the domain of *game-based learning*, which requires scholars to be problem solvers as they work on quests to achieve a particular goal. For learners, this approach combines targeted learning goals with the fun of earning points or badges or moving up on leaderboards, much like they would in a video game. Game-based learning and mainly digital learning games will be extensively analyzed in Chapter 3.[6]

2.2 Active Learning

During the 1960s, Edgar Dale theorized that learners memorize more information by what they ”do” as opposed to what is ”heard,” ”read,” or ”observed.” Dale’s Cone of Experience is a model that combines several theories related to instructional design and learning procedures.[10] That is, active learning is more efficient than conventional. His study led to the development of the Cone of Experience.

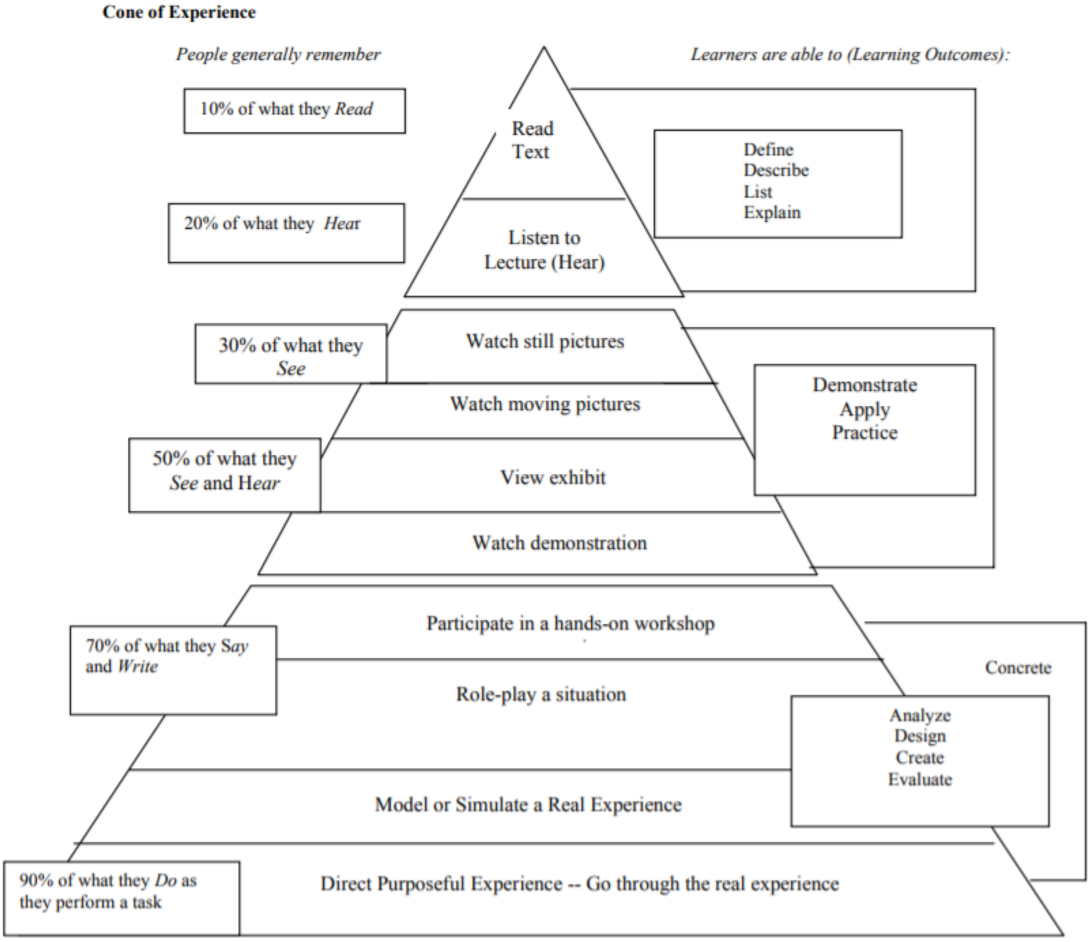

Cone of Experience
People generally remember
Learners are able to (Learning Outcomes):
10% of what they Read
20% of what they Hear
30% of what they See
50% of what they See and Hear
70% of what they Say and Write
90% of what they Do as they perform a task
Read Text
Listen to Lecture (Hear)
Watch still pictures
Watch moving pictures
View exhibit
Watch demonstration
Participate in a hands-on workshop
Role-play a situation
Model or Simulate a Real Experience
Direct Purposeful Experience -- Go through the real experience
Define
Describe
List
Explain
Demonstrate
Apply
Practice
Concrete
Analyze
Design
Create
Evaluate

Figure 2.3: *Cone of Experience [10]*

As shown in Figure 2.3, people remember only 10% of what they read, 20% of what they hear, and 50% when they are both listening and watching. There are better results when becoming more active. Thus, people remember 70% of what they read and write and 90% of what they do.

So, student-centered learning, known as active learning, is proven much more effective.

Active learning is a teaching procedure that includes students engaging actively with the course material through dialogues, problem-solving, case studies, role plays, and other approaches. Active learning methods put a more prominent degree of commitment on the student than passive approaches such as lectures, but teachers' guidance is still essential in the active learning classroom. Active learning exercises may range in length from a couple of minutes to full lesson sessions or over further class sessions. [11]

2.2.1 Active learning benefits

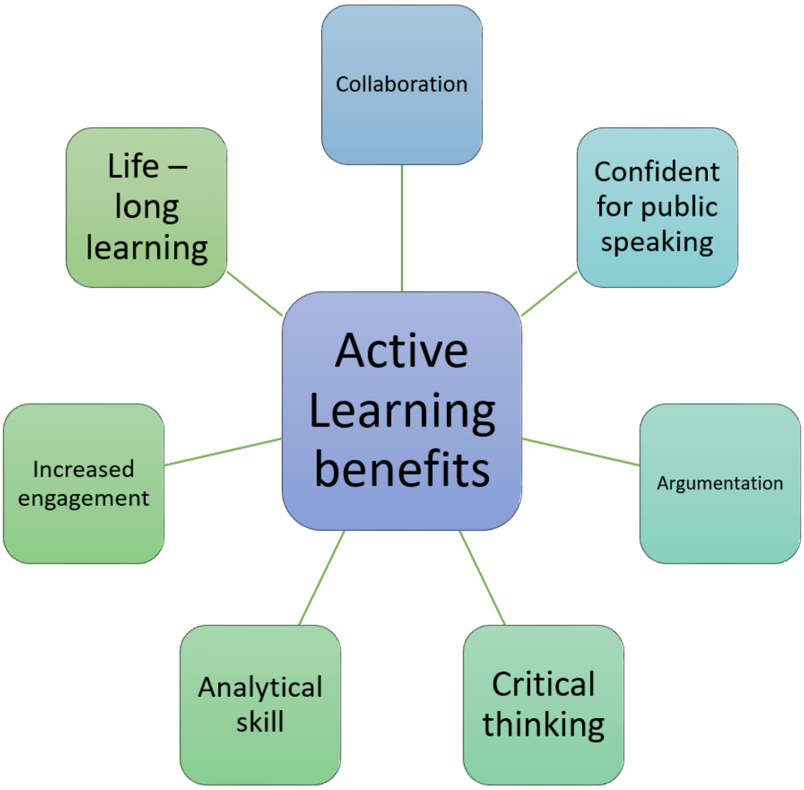


Figure 2.4: *Active Learning Benefits*

Active learning has many benefits for the learner, and as they are at the center of training, they develop multidimensionally. The student does not just learn the curriculum of the lesson but also develops particular elements of his personality.

Collaboration is one of the main elements of active learning. Students often work in small groups as part of the learning procedure. They become more responsible, as a whole team depends on them, but at the same time, they depend on the other team members as well. Moreover, they exchange opinions to complete the given task, so they evolve their communication skills and become more confident speaking in public. Furthermore, critical thinking and argumentation skills are developed when discussing and interacting with others. Analytical skills also help learners be better at problem-solving and applying their learning. [12]

In an active learning method, learning is not exclusively about the content but also about the process. Active learning forms students' independence and their ability to learn. Active learning gives learners more great involvement and control over their learning. This implies that students can better resume learning once they have finished school and university. [13]

2.2.2 Active learning effectiveness

The perception of learning is different for each person, and sometimes people learn without even realizing it. The following study [14] conducted at the Chemistry and Chemical Biology Department at Harvard University in Cambridge for the Physics course shows how effective active learning is without the learners realizing it.

The class is separated into two groups randomly. For the static equilibrium lecture, the first group instructor followed active learning methods, while the second group professor taught using passive lectures for one lecture. For the fluids lecture, the professors exchanged their teaching methods. The first teacher uses passive lectures, while the second uses active learning. Both groups are taught the same material. So, both groups are instructed with the active and passive lectures but for a different topic. By the end of the class period, the students answered both a *learning test* that is not taken into consideration for their final grade and a *feeling of learning survey* where they grade how much they think they have learned for each lecture. - which implies with which method they think they learned more.

As shown in the two graphs Figures 2.6 and 2.5, the objective learning evaluation, the learning quiz, shows that the students participating in the active lecture have better learning outcomes than those in the passive lecture. At the same time, there is a contradiction between what the students thought they had learned and what they actually learned. The survey showed that students believe that they learned more when following the passive lecture than active lectures.

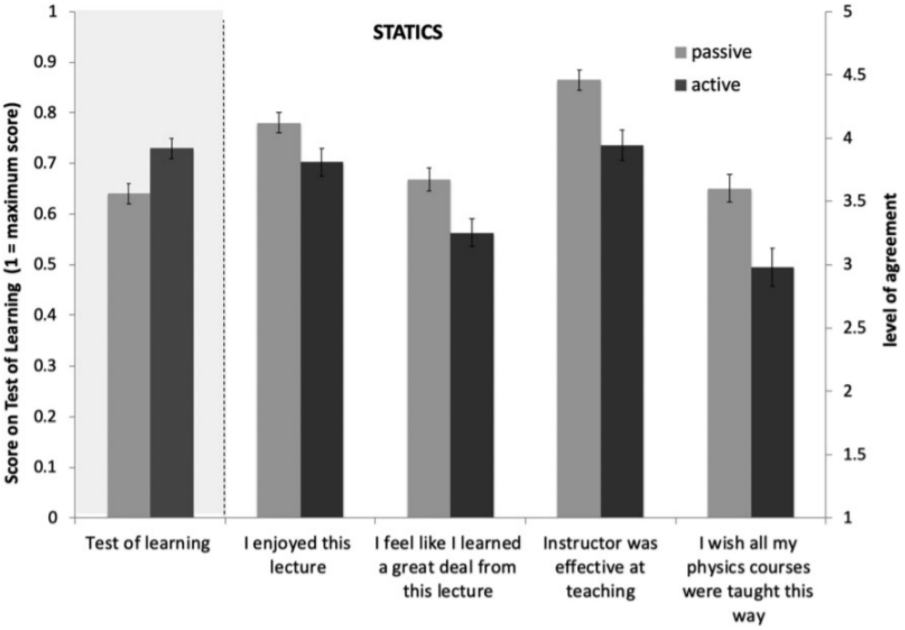


Figure 2.5: *A comparison of performance on the test of learning and the feeling of learning responses between students taught with a traditional lecture (passive) and students taught actively for the statics class. Error bars show 1 SE. [14]*

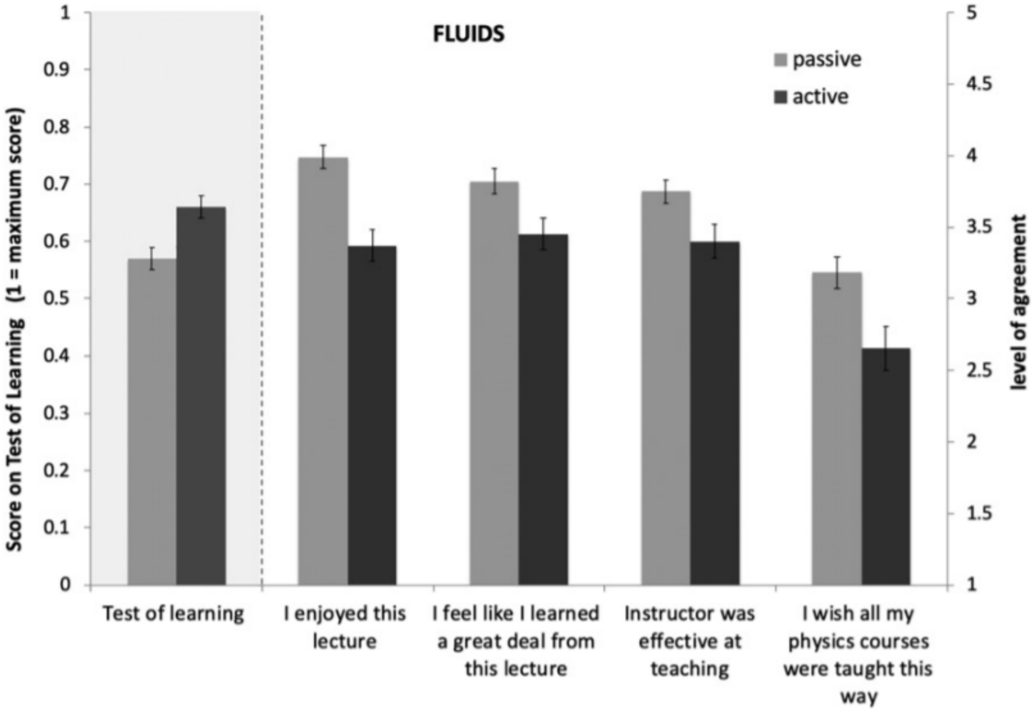


Figure 2.6: *A comparison of performance on the test of learning and the feeling of learning responses between students taught with a traditional lecture (passive) and students taught actively for the fluids class. Error bars show 1 SE. [14]*

Chapter 3

Learning Games

The children's attitude towards games, especially video and computer games, contrasts with their attitude toward school. Their attitude towards games is what educators would like their students to have in the learning process; curious, competitive, collaborative, results-oriented, actively seeking knowledge and solutions. Moreover, like all human beings, kids love to learn when it is not pushed on them. Therefore, modern computer and video games provide not forcing learning opportunities every moment.The combination of learning and digital games might achieve the pleasant and motivating lessons the teachers have tried all these years. "A sine qua non of successful learning is motivation: a motivated learner can not be stopped." [15]

3.1 Introduction and Definitions

Games are known since the ancient times. The first known board game was played in ancient Egypt around 3100 BC, and it is called Senet [16]. Many games were invented in ancient times but are still popular nowadays, like Go [17], backgammon [18], checkers[19]. In the course of time, more and more games are created utilizing resources and are constructed according to by the progress of each century.

The technological process and the development of the first computers in the late 30s and at the beginning of the 40s paved the way for creating the first digital games. OXO, known as tic-tac-toe nowadays, is one of the first video games, and it was released in 1952 at Cambridge University. The computer, EDSAC (Electronic Delay Storage Automatic Calculator), was used for developing the game; it was built in

1949. So, it is clear that games follow the pace of technological development. Some other games were later developed as well, but not for commercial purposes. These games were in the research and development stage. The first commercial game was published in 1972, and it was called ”Computer Space” and it could be played in the first game console known as ”The Brown Box”. As a result, a new category of games appeared, named as digital games. [20]

Games that use some computational machinery are called digital games. This computational machinery could be personal computers, smartphones, or a video game console. [21]

Some digital games are created for not only entertainment purposes but also for achieving one additional goal; these games are called serious games[21]. These additional goals more often relate to learning and education, training, health, social awareness, and marketing [22].The digital games that exclusively entertain the player are called *entertainment games*.[21]

Dörner et al. differentiate educational games from learning games. They argue that educational games are for formal education while learning games are primarily used for informal education [21]. Formal education takes place in educational institutions where the educational process has some curriculum that must be taught and the students must be evaluated. Informal education takes place outside of any educational environment by participating in activities without having a specific learning goal and occurs unintentionally in people’s daily lives.Non-formal learning happens beyond formal learning environments within some sort of organizational framework with a learning intention, but without following a curriculum or the existence of an assessment [23].

Often in the bibliography, these differences are not taken into account, so learning and educational games are used for describing both serious game categories.

Another term that has emerged in the past decades is *Gamification*. The term needs to be defined and distinguished from entertainment and serious games. According to Deterding et al. [24], *“Gamification” is the use of game design elements in non-game contexts.* Consequently, gamification’s result is not necessarily a ”full” game but adopting and using game methods and elements for non-game applications and systems enhancing the user’s experience and engagement. Despite the fact that gamification is commonly used in digital environments, the term should not be bounded to digital technology.

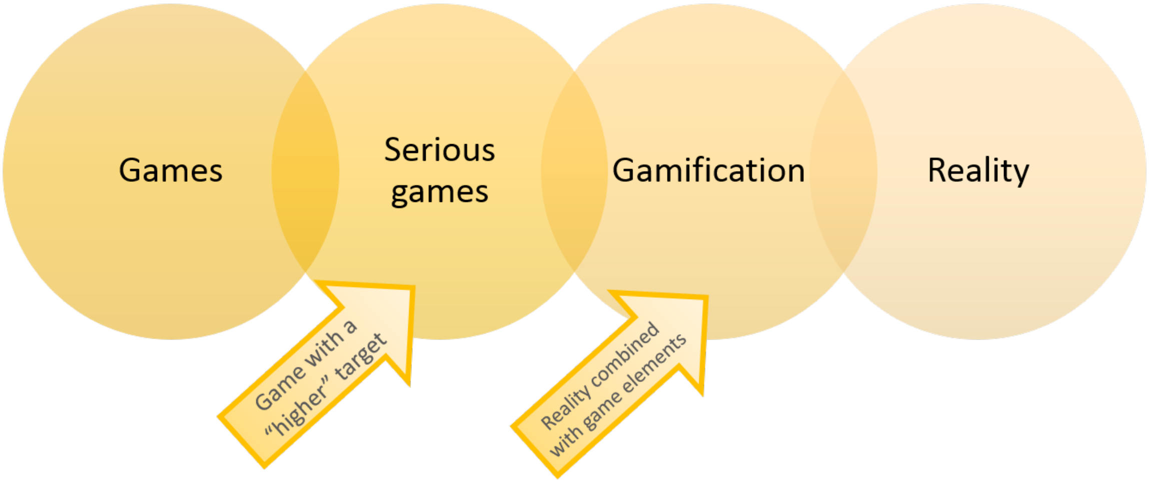

Games
Serious games
Gamification
Reality
Game with a "higher" target
Reality combined with game elements

Figure 3.1: *Serious games VS Gamification [25]*

3.2 Game elements

Werbach and Hunter [26] have developed a framework for the elements that are commonly used in gamification. Its structure is a three-level pyramid, Figure 3.2.

The base of the pyramid consists of the components, the middle layer incluedes the mechanics, and on the top is the dynamics. The pyramid is used as a guide in order to create the best possible experience for the player. The game elements should be used effectively in order a game to be successful and it is not critical to use the best or all the elements to achieve this.

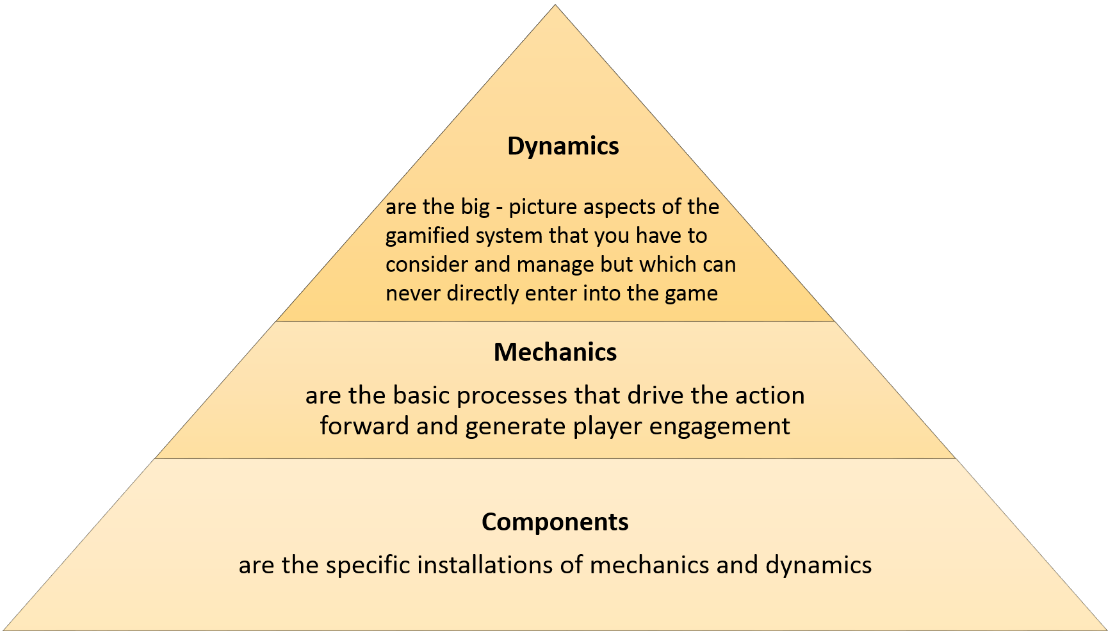

Dynamics
are the big - picture aspects of the gamified system that you have to consider and manage but which can never directly enter into the game
Mechanics
are the basic processes that drive the action forward and generate player engagement
Components
are the specific installations of mechanics and dynamics

Figure 3.2: *Game Elements Hierarchy [26]*

Dynamics

The most abstract level is the dynamics. The most important are the constraints, the emotions, the narrative, the progression, and the relationships.

Each game has some constraints that are limiting the freedom of the players. The games have some rules that must be followed and there are boundaries in place that have been specified within the configuration process. The games create significant options and intriguing problems for the learner to select from and eventually may change the path or process of the simulation. [27] [28]

The games evoke emotions. The full spectrum of emotion can lie from pure joy to sadness, or from feeling a sense of accomplishment to frustration when making a bad decision. This variety of emotions can make the experience richer and drives people to want to play even more.

The narrative of the game is another dynamic element. Each piece of the game should be put together creating a storyline with a flow. Each game's piece should exist for a specific reason and should be correlated to the other pieces in order to maintain coherence. Otherwise, there will be an abstract bunch of items without serving any purpose.

The progression, thus the player's growth and development is essential as well. The player should feel that there is a prospect for improvement or at least move forward from where they initiated without constantly repeating the same action and remain stable.

Finally, in a game there are relationships. There is the social interaction among the players, as they can be friends, teammates, or opponents. This interaction can generate feelings of camaraderie, altruism, and status.

Mechanics

The middle layer of the pyramid refers to the mechanics. Each mechanic contributes to completing one or more of the dynamics. The following ten are the most noteworthy [28] [26]:

- Challenges, the games set some objectives, puzzles, and other missions that should be accomplished requiring some effort.

- Chance, there is some luck involvement that makes the result slightly random and it is not depending completely on the player.

- Competition, a single player or a group of them are competing against each other and there are winners and losers at the end.

- Cooperation, is the opposite of competition, thus the players should cooperate with each other to accomplish a shared goal.

- Feedback, the players can see their performance in real-time.

- Resource Acquisition, games provide you with useful things or give you the chance to get collectible items in order to move the game forward.

- Rewards, are some benefits that the player gets for action or achievement in the game.

- Transactions, thus purchasing and selling or trading something with others or with the system either directly or through intermediates.

- Turns, each player waits to participate in the activity in their turn following a sequential order.

- Win states, a condition within a game or level which represents the successful completion of a predetermined objective or mission.

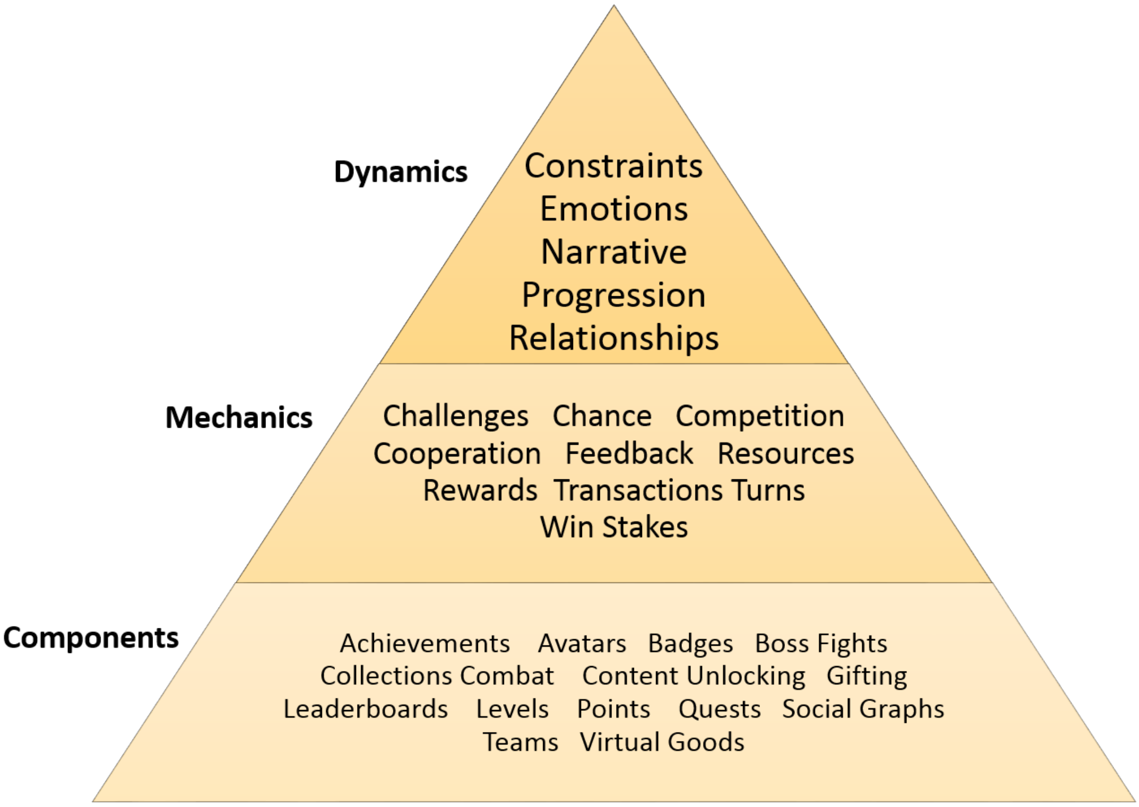

Dynamics
Constraints
Emotions
Narrative
Progression
Relationships
Mechanics
Challenges Chance Competition
Cooperation Feedback Resources
Rewards Transactions Turns
Win Stakes
Components
Achievements Avatars Badges Boss Fights
Collections Combat Content Unlocking Gifting
Leaderboards Levels Points Quests Social Graphs
Teams Virtual Goods

Figure 3.3: *Game Elements Hierarchy [26]*

Components

The pyramid base consists of the components that are used in a game. Each component connects to one or more pieces of its higher layer, thus the mechanics. Many components are used in games to create a better game experience and engage them.

First and foremost, it is the use of points that can represent multiple issues. The scoring system of the game can be based on the points the player collects during the game session. The points may represent the players' performance and their level as well. Moreover, points can define a state condition, like the player should collect a certain amount of points to level up or win or unlock another feature or earn a reward.

Furthermore, points provide feedback instantly to the user, showing their progress. In addition, points may be an external display of progress, thus the players may be able to see the points of others linked to the game. Lastly, by providing the data related to points to the game designers - like the number of points, the earning rate, etc - they can analyze, evaluate and redesign their games. [26]

Another fundamental component is the achievements and badges. The achievement is a defined objective whereas the badge is its visual representation. A badge is a special distinction that it can be given to gamers when they meet a goal. [29] There can exist multiple badges for different activities and goals. Their use motivates the player, as they shed light on the targets that they should achieve and they provide guidance on what the game system offers. Noteworthy, the badges can be used as a virtual status symbol, as they show what the participant has achieved. Moreover, the players can be separated into groups based on the badges unlocked. [26] Some badges can be earned randomly to surprise the player.[30]

Leaderboards are the other elemental component. The leaderboard displays the performance of individual players in comparison to other players. [31] Leaderboards are an ordered list of players based on the points they have obtained in a specific game.[32]There can be multiple leaderboards in a game, for instance, there can be seasonal leaderboards, or for each level, and a leaderboard for a specific crowd like the users in a social media account. [33] Leaderboards can motivate the player to set a higher goal in the game by comparing their score with those of the leaders in the game and eventually improve their performance. [31]

Points, Badges, and Leaderboards consist the PBL triad, which is commonly used as an initial point in the gaming industry. However, not all projects demand their use. Some further notable components are useful as well and are presented below: [26]:

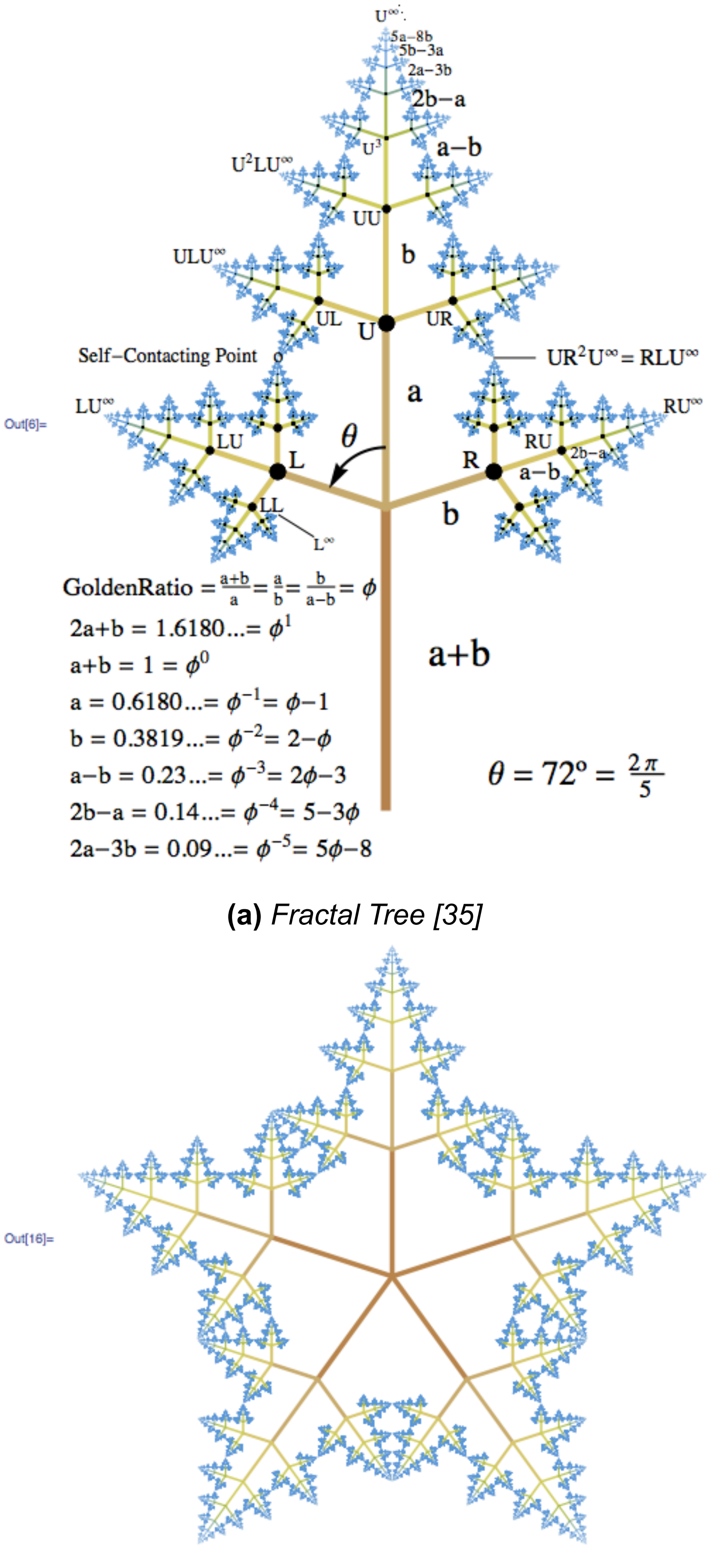
- Avatars, a visual representation of the user's character.
- Boss fights, an extraordinarily hard challenge, a high-level monster to defeat in order to level up.
- Collections, group of various things or badges for collecting.
- Combat, a defined battle.

- Content unlocking, thus it needs to do something, perhaps an achievement, or reach the next stage, in order to gain access to particular fresh content in the game [28].
- Gifting, hence sharing and giving resources to others or getting something.
- Levels
- Quests, predefined challenges with goals and rewards.
- Social Graphs, representation and interaction with user's social network within the game.
- Teams, group of players who collaborate to achieve a common goal.
- Virtual goods, things that players can purchase with virtual or real currency.

3.3 Digital Learning Games for Mathematics

Mathematics is considered disconnected from everyday life and is just a school subject. However, it is remarkably useful, and it applies to almost every aspect of daily life without anyone realizing it most of the time. Apart from its obvious applications, like engineering, physics, computers, science, etc., mathematics can be found in art, music, medicine, geography, public transportation, skate part design, and almost every field. [34] Mathematics could be characterized and used as a universal language as its ”vocabulary” and ”syntax” are the same in every country in the world. Its symbols are comprehensive, and its structure is determined. A mathematical expression or formula has an identical meaning, regardless of the text's language. Moreover, the numbers contribute to human communication, as the numbers are used to measure things, read the clock, in transactions, etc. As a result, it is evident why the importance of mathematics has been appreciated since ancient times.

Mathematics is considered a demanding and difficult subject, and many students face difficulties with it. Mathematics requires time, effort, and energy to comprehend and absorb the new concepts and curriculum. However, each pupil needs a different amount of time to get used to the new learning. Math's unique characteristic is that the student can not be introduced to new material if they have not completely understood, what it is already taught because each new concept demands some specific knowledge. However, many times, there is a plan for the school curriculum, and the teachers should move on and cover the predefined curriculum. The result is that some students can not follow the lessons as they have some knowledge discontinuity.



(a) *Fractal Tree [35]*

(b) *Fractal Star [35]*

Figure 3.4: *Mathematical Fractals*

Mathematics know-how is cumulative. It means that mathematics is like building a wall, and each mathematical concept is a brick in the wall. It requires gaining the understanding of each brick to move and construct a new brick on the top of the previous one, otherwise the ”wall” will collapse. The foundation of the ”wall” is constructed in the primary school, introducing the students to the rules of addition, multiplication, basic equations, and more. In every grade, students learn more and more concepts and formulas with which they should get acquainted. So, it is important to understand that a bad grade does not only show the current student’s performance but also that they lack essential knowledge for succeeding academically in the future. [36]

Nowadays, learning games are developed to support education and keep the students more engaged, but it is observed that they are used rarely despite their growth. As far as mathematics is concerned, the majority of them concentrate on numbers and operations, algebra, geometry, probabilities, and functions [37]. Thus, contemporary instructors have new digital tools to use and support their lessons and improve their students’ performance.

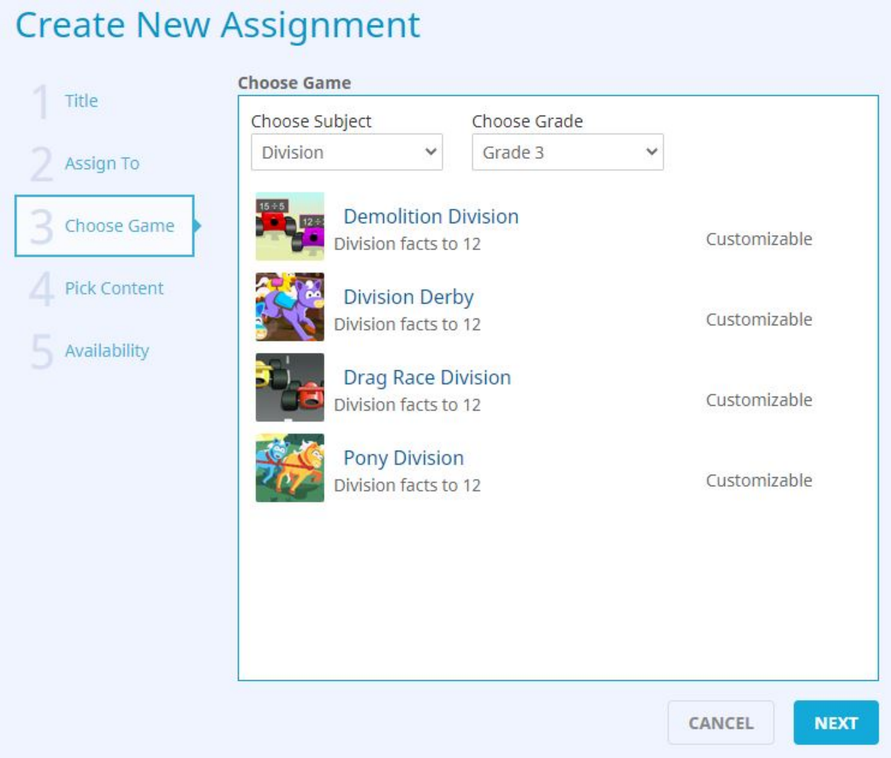
The following three paragraphs present games or platforms with educational games. These games focus on maths, but also they may offer other games for different fields as well. The first one is Arcademics, an online video game hub for learning games in primary school. The second one is Minecraft, a well-known video game, but it is presented from an educational perspective. Finally, the last game is Bunny Times, which is proposed by the National Council of Teachers of Mathematics, for learning addition and multiplication.

3.3.1 Arcademics

Figure 3.5: *Arcademics Logo [38]*

Arcademics is an online educational video game hub for kindergarten and elementary students, that approaches dynamically the learning of basic mathematics, language arts, vocabulary, geography, and thinking skills. The instructors or parents are allowed to access data tracking reports regarding their children’s performance. [39]

The games can be classified based on the grade addressed or the school subject and curriculum for better navigation and selection of the suitable game. Some games



Create New Assignment

1 Title

2 Assign To

3 Choose Game

4 Pick Content

5 Availability

Choose Game

Choose Subject: Division

Choose Grade: Grade 3

Demolition Division	Division facts to 12	Customizable
Division Derby	Division facts to 12	Customizable
Drag Race Division	Division facts to 12	Customizable
Pony Division	Division facts to 12	Customizable

Figure 3.6: *Create a new assignment [38]*

are customizable, and the teacher can adapt them to the needs of each separate pupil or group.

The teacher can create assignments, thus games, and assign them to the students, or groups of them. The instructor entitles the task and assigns it to suitable students. Then they select the game and configure its customization, if feasible and needed. Finally, there are options regarding the assignment's availability, thus the time period in which it is active. (Figure 3.6)

Arcademics provides reports and insights about the engagement and performance of each scholar in the premium subscription.

As learners play, they are able to earn distinction accomplishment badges and fun power-ups for using them in the games.(Figure 3.7)

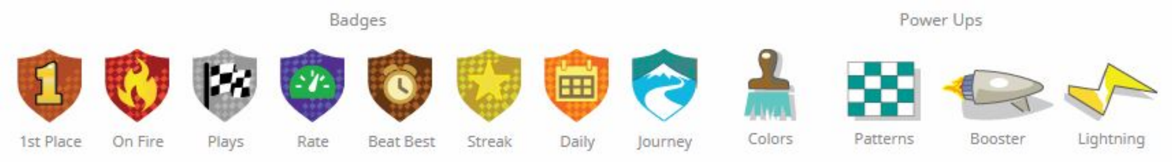


Figure 3.7: *Badges and Power Ups*

Arcademics suggest multiple ways to use their environment and keep the students' engagement and excitement at high levels.[40]

- Class Championship

  Assign a game to the student group for a limited period of time. The target is for the student to climb to the top of the leaderboard and be rewarded with small prizes. The leaderboard can host the pupils who scored the most points, either the fastest time achieved, or the student with the most progress.

- Beat the Teacher

  The instructor plays with the students, and they aim to beat their teacher.

- Grade Tournament

  Each grade consists of some classes. At first, there is a mini tournament in the class, the top two winners from each game qualify and move to the next round until a final four is created. Each class's final four are combined together and continue competing until there is a final four for the overall grade. Finally, the top four finishers will be awarded.

- Class Cup

  Each class can have a weekly championship.In particular, the students will be able to compete and aim to gain the championship trophy until next week's challenge.

- Class VS Class

  Each class forms a team, and the classes compete with one another. The points of each student are accumulated to calculate each team's total points. The team/class with the highest total score is proclaimed as the winner.

- King of the board

  Have a game played on a whiteboard, and the winner stays on it. In your lab time, have one of the games shown on the whiteboard, with the winner staying on it to play new challenges. [40]

An example of a random game in Arcademics is Jet Ski Addition, Figure 3.8. There are four jet skis and a math equation with four possible answers. The player should choose the correct answer to move forward and reach first the finishing line.

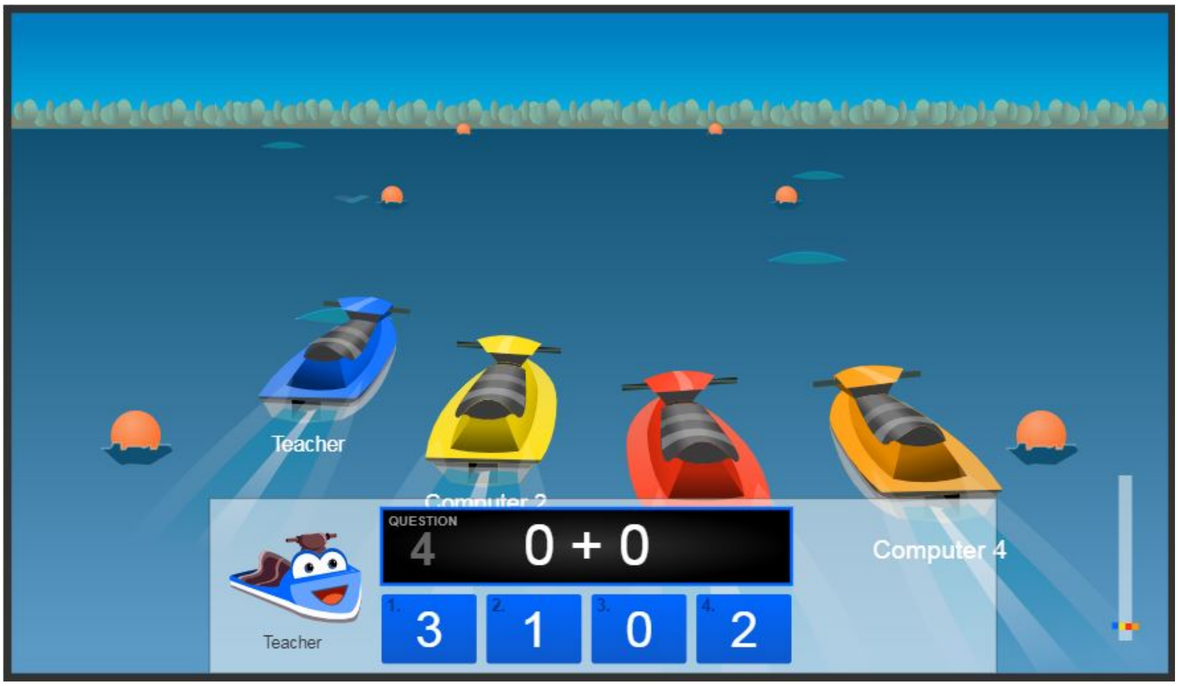


Figure 3.8: *Jet Ski Addition [38]*

3.3.2 Minecraft Education Edition

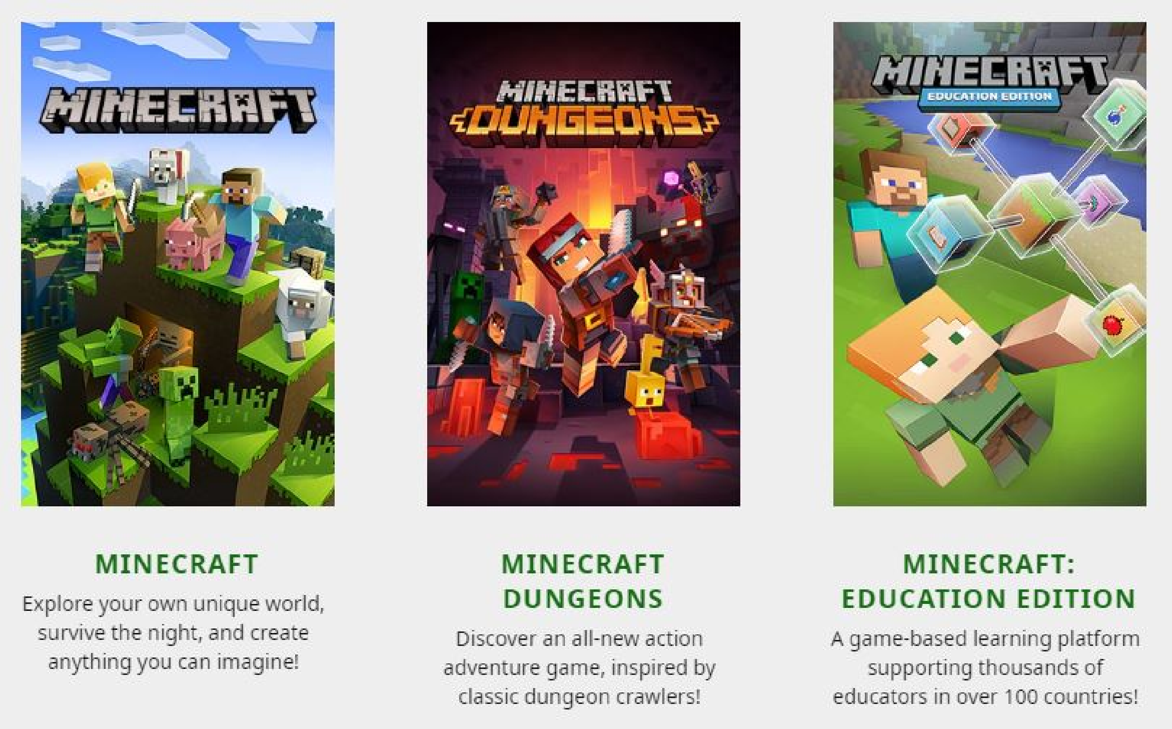
Minecraft is a renowned sandbox [1] construction video game in which the players build their own world by placing three-dimensional blocks. The game has two basic modes, creative and survival. In creative, the players have un-

Figure 3.9: *Minecraft Logo [41]*

[1] A sandbox is a type of game in which minimal character constraints are placed on the gamer, allowing the gamer to wander and modify a virtual world at will. Sandbox games can contain structured elements – such as mini-games, tasks, submissions, and storylines – that the players may ignore. It is also known as an open-world or free-roaming game. [42]

limited supplies to build their world. In survival mode, they must find their own supplies and food and defend themselves from bad creatures like zombies in order to stay alive. A designated but totally optional game's goal is to travel to a dimension called the End and conquer the Ender dragon.[43] Generally, the purpose of the game is to be creative and build, explore, gather resources and, if necessary, survive.[44] This game contradicts the game elements that are presented in chapter 3.2, as there are no points, leaderboards, quests, or badges. It focuses more on playing and not on gaming[45]

The first Minecraft release was in 2009 and gained a large audience quickly. As a result, five years later, in 2014 Microsoft bought the game. In 2016, the Minecraft Education Edition was released with some features made for using the game in the classrooms to create a playful and pleasant educational setting. In May 2020, Minecraft Dungeons was launched. The main difference is that the new game offers a linear, narrative-driven experience, and it is not an open-ended sandbox as the original game. [46]



MINECRAFT

Explore your own unique world, survive the night, and create anything you can imagine!

MINECRAFT DUNGEONS

Discover an all-new action adventure game, inspired by classic dungeon crawlers!

MINECRAFT: EDUCATION EDITION

A game-based learning platform supporting thousands of educators in over 100 countries!

Figure 3.10: *Minecraft Games [41]*

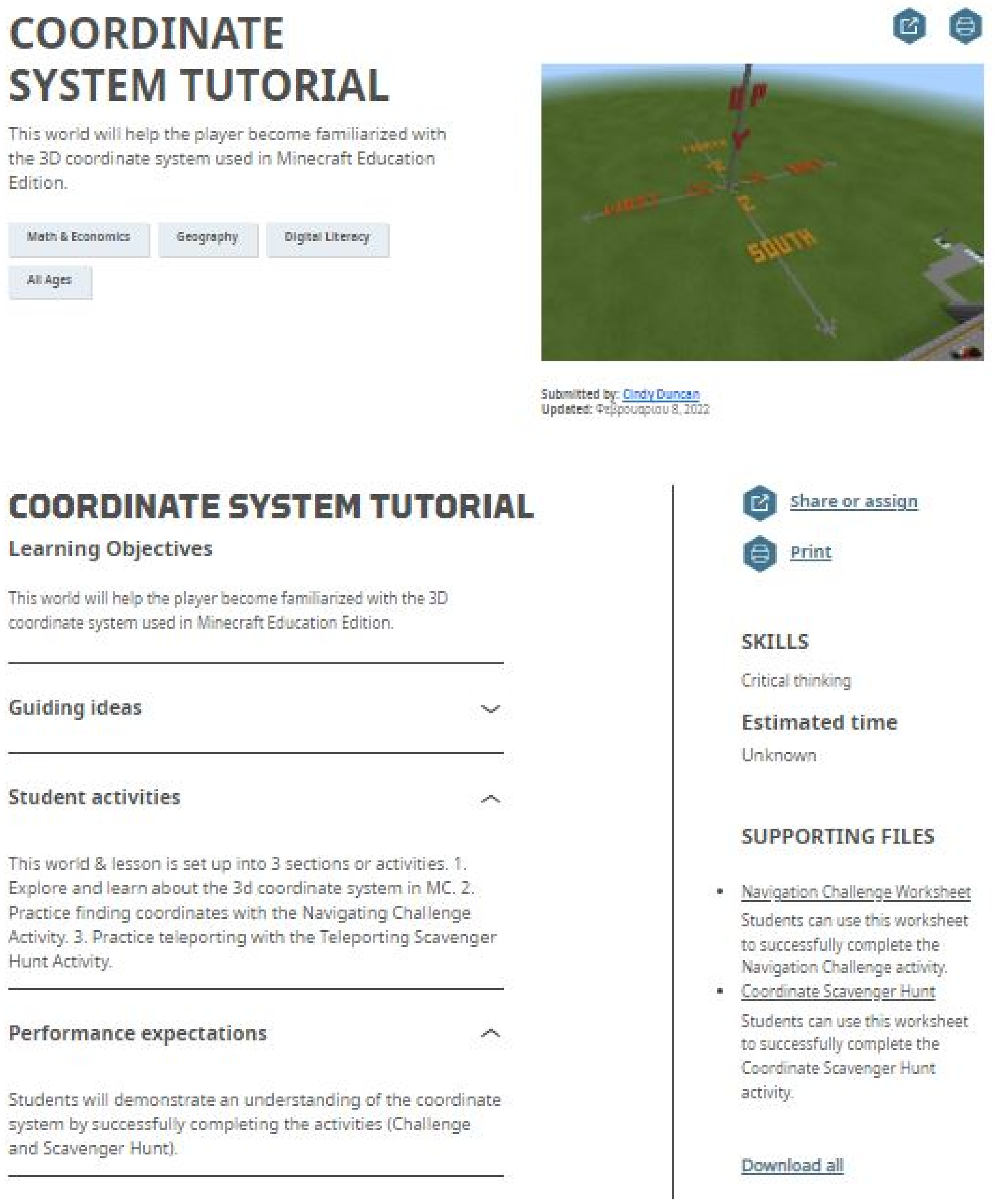
This section focuses on Minecraft Education Edition. This version offers numerous pedagogical tools, lesson plans, and challenges regarding science, math, computer sciences, language arts, history and culture, art and design, digital citizenship, social-emotional, equity and inclusion, and climate and sustainability. Many lessons have a supplementary Minecraft world to support their learning and apply the knowledge in the game. Each course may address specific age groups and refer not necessarily to only one field but provide interdisciplinary development. Each one has a learning objective and expectations about the students' performance. These lessons may provide the educators with guiding ideas, student activities, supporting files, and working sheets with appealing tasks. Furthermore, Minecraft is not a competitive game, without a clear definition of success or failure, allowing students to be free and safe to make decisions, take actions, and make mistakes. As a result, these games contribute to the student's engagement and help their assessment as well, without feeling the exam pressure as they feel playing. Generally, these game lessons equip the students with essential skills that the modern era demands, like critical thinking, problem-solving, creativity, collaboration, etc.[45]

To be more specific, an example game, the Coordinate System Tutorial [47], will help analyze how Minecraft is used for education purposes. This lesson has a specific world and two additional activities. For the first activity, the student should move to specific coordinates, dig, find and document the hidden items. The second activity is about teleporting.

The guiding ideas describe what the teacher should explain to the students before starting playing the game, thus creating the student's background. In this example, the instructor should explain what the coordinates are, why they are useful and practical, what kind of information they give to the people, and explain some features of the specific game's world.

The student activities inform the educator about how the game and some supplementary files can be used and provide ideas for indoor the classroom activities. Specifically, regarding the example game, it suggests separating the lesson into three sections. Firstly, they explore and learn about the 3d coordinates, then they try to navigate and find some coordinates using the activity provided in the additional files, and finally practice teleporting using the second file.

Performance expectations point out what the learner should have learned by the end of the lesson. Particularly, in the coordinates game, they should successfully finish the activities.



COORDINATE SYSTEM TUTORIAL

This world will help the player become familiarized with the 3D coordinate system used in Minecraft Education Edition.

Math & Economics | Geography | Digital Literacy | All Ages

Submitted by: Cindy Duncan
Updated: Φεβρουαρίου 8, 2022

COORDINATE SYSTEM TUTORIAL

Learning Objectives

This world will help the player become familiarized with the 3D coordinate system used in Minecraft Education Edition.

Guiding ideas

Student activities

This world & lesson is set up into 3 sections or activities. 1. Explore and learn about the 3d coordinate system in MC. 2. Practice finding coordinates with the Navigating Challenge Activity. 3. Practice teleporting with the Teleporting Scavenger Hunt Activity.

Performance expectations

Students will demonstrate an understanding of the coordinate system by successfully completing the activities (Challenge and Scavenger Hunt).

Share or assign

Print

SKILLS

Critical thinking

Estimated time

Unknown

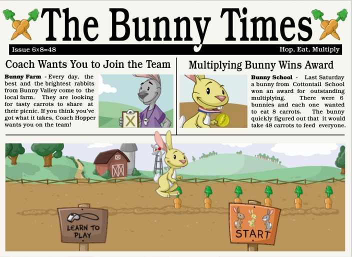
SUPPORTING FILES

- Navigation Challenge Worksheet
  Students can use this worksheet to successfully complete the Navigation Challenge activity.
- Coordinate Scavenger Hunt
  Students can use this worksheet to successfully complete the Coordinate Scavenger Hunt activity.

Download all

Figure 3.11: *Coordinate System Tutorial as presented in Minecraft page [47]*

3.3.3 Bunny Times



The Bunny Times

Issue 6×8=48 Hop, Eat, Multiply

Coach Wants You to Join the Team

Bunny Farm - Every day, the best and the brightest rabbits from Bunny Valley come to the local farm. They are looking for tasty carrots to share at their picnic. If you think you've got what it takes, Coach Hopper wants you on the team!

Multiplying Bunny Wins Award

Bunny School - Last Saturday a bunny from Cottontail School won an award for outstanding multiplying. There were 6 bunnies and each one wanted to eat 8 carrots. The bunny quickly figured out that it would take 48 carrots to feed everyone.

Figure 3.12: *Bunny Times start scene [48]*

Bunny Times [48] is a game that allows the children to learn and understand the basic mathematical operations emphasizing addition and multiplication. It is suitable for kindergarten kids to students in the first grades of elementary school. The fundamental scene is a field with carrots - grid structured - and one or more bunnies or groups of them (gray and brown bunnies). The aim of the game is that the child should find how many carrots are available, and then the bunnies hoppingly collect them. The calculation can be either with counting or with calculating with the use of addition and multiplication. The multiplication facts range from 2x2 through 12x12.

It consists of eleven different levels. Each level familiarizes the students with different factors. The higher the level, the wider the field with the carrots can become, and the more demanding the calculation is. In more advanced ones, fog can appear to hide the carrots and urge the child to find other skip-counting schemes to determine the correct answers (the overall carrots) instead of counting one by one.

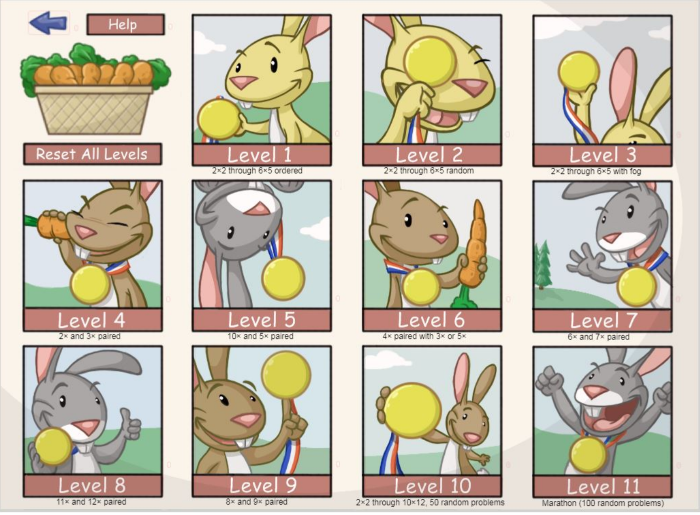
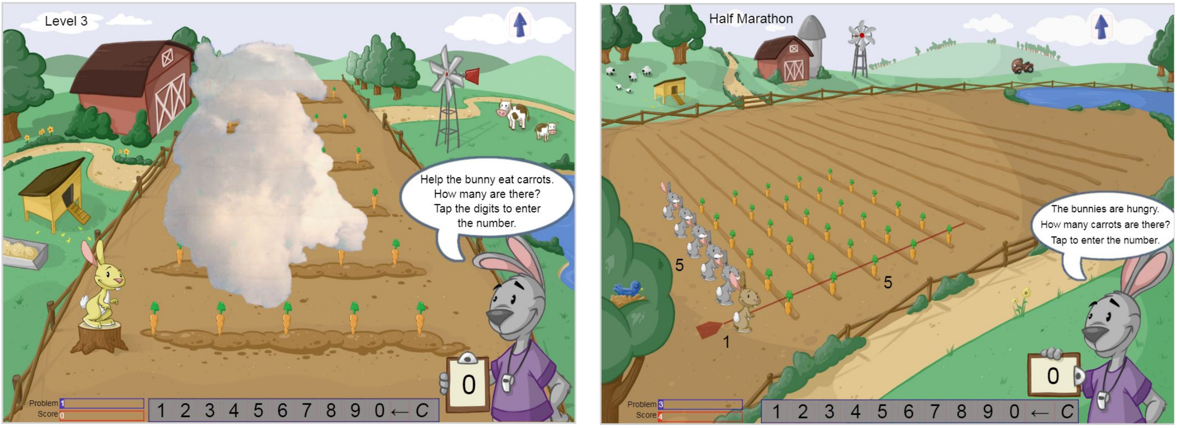


Figure 3.13: *Bunny Times levels [48]*

As the carrot fields grow larger, children move from a single bunny to manipulating entire teams of bunnies. When there are two different teams, the bunnies can be redistributed by sliding the line that separates them. For instance, a field with 5 rows and 7 columns can be split into two smaller grids, like 3x7 and 2x7.

Bunny Times rewards repeated play with opportunities for children to improve



Help the bunny eat carrots.
How many are there?
Tap the digits to enter
the number.

The bunnies are hungry.
How many carrots are there?
Tap to enter the number.

(a) **(b)**

Figure 3.14: *(a) A single bunny and the field in fog (b) A random screenshot of the game with two groups of bunnies [48]*

their scores and performance and develop, at their own pace, new knowledge and methods for multiplication. By advancing successfully through all the eleven levels, juniors will approach multiplication with confidence, accuracy, and learning.

Chapter 4

Unity

4.1 Introduction to Unity Engine

Unity is a development platform for video games, and it is one of the most widespread game engines worldwide. Game engines are special software that game developers use to produce both 2D and 3D video games. In 2005, Unity was launched with the objective of giving a game engine that anybody could learn, use and contribute to. It has shaped a community of engineers with an enthusiasm for game development.

Figure 4.1: *Unity Logo*

As a game engine, Unity is an incredible platform for designing imaginary worlds, estimating physics for objects in these worlds to move, jump or collide, playing sounds, and more. Moreover, designers adore game engines like Unity due to the fact they give reusable code that each game would need. [49]

Like most game engines, Unity has five components. The primary component is a program that contains the game logic, a rendering engine that generates 3D animated graphics. An audio engine component provides algorithms to create and manipulate sounds. For physical laws, a physics engine handles movement and collisions. For non-player actions and decisions, there is an Artificial Intelligence (AI) component. [49]

Unity supports over twenty-five platforms and technologies, like iOS, Android, Linux, NintendoSwitch, PlayStation (Figure 4.2). So the same game can be deployed

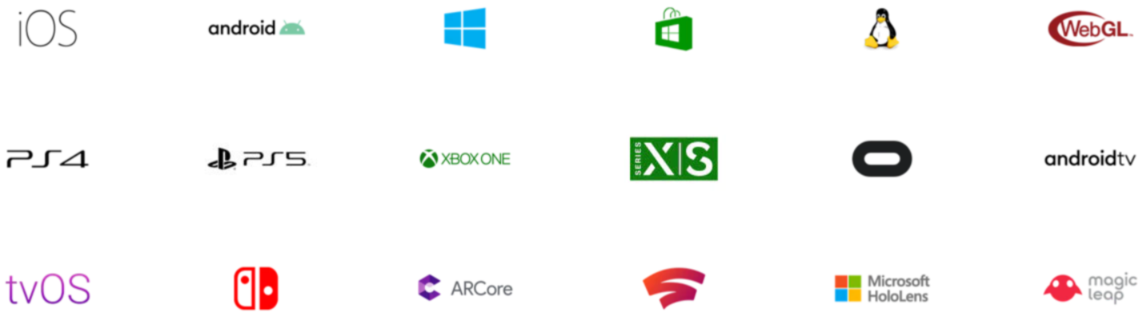


Figure 4.2: *Sample of Unity supported platforms*

to these different platforms and be accessible to the majority of the world. [50]

4.2 Unity interface

Figure 4.3 shows the Unity environment. The most common windows are:

1. Hierarchy window,
2. Inspector window,
3. Scene view,
4. Game view,
5. Project window,
6. Animator window, it is not part of the basics, but it is used in the present thesis, so it is necessary to refer to this.

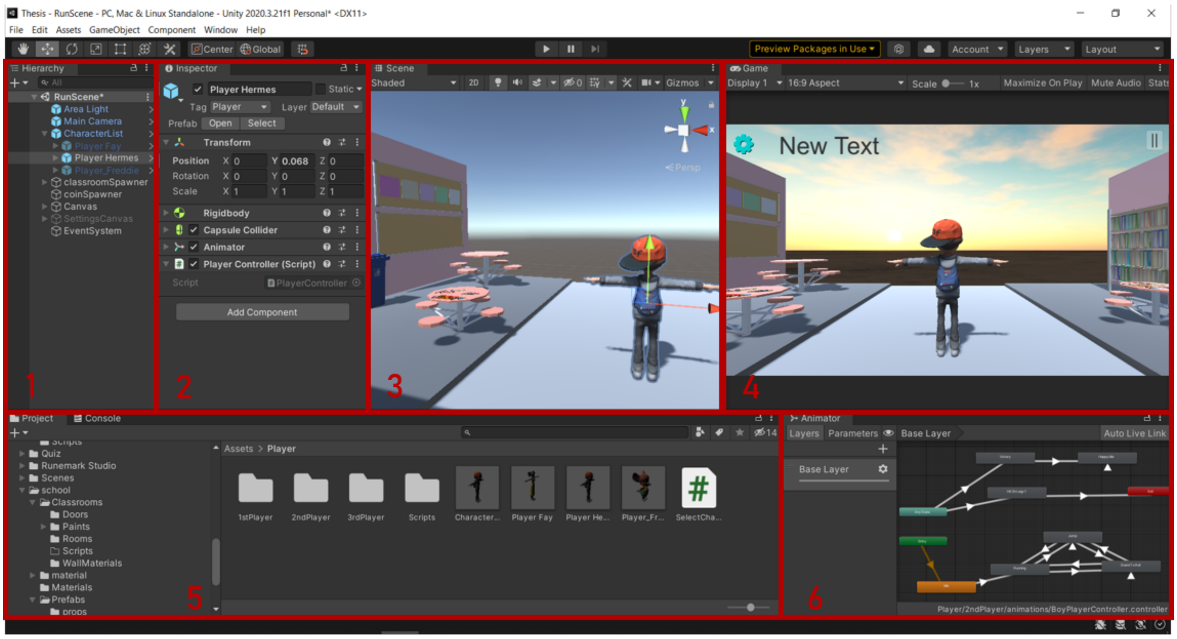

Thesis - RunScene - PC, Mac & Linux Standalone - Unity 2020.3.21f1 Personal* <DX11>
File Edit Assets GameObject Component Window Help
Hierarchy
Inspector
Scene
Game
Project
Console
Animator
New Text
1
2
3
4
5
6

Figure 4.3: *Unity interface*

The Hierarchy window displays every GameObject in the Scene, such as models, Cameras, or Prefabs. A game object can be selected, added, deleted, set to visible or invisible, or get into a group of objects in that window. [51]

The Inspector window shows all the components attached to the selected GameObject. Moreover, every property can be edited or deleted, or else a new property can be added to that GameObject. [52]

The Scene view is an interactive view of the created world. In that view the scenery, characters, Cameras, lights, and all other types of GameObjects can be selected, manipulated, and modified. [53]

The Game view shows what the player sees while playing the game using one or more Cameras. This view is active, when the play button in the upper part of the screen is pressed. In play-mode, the programmer can modify the Scene, and the changes are shown instantly, but simultaneously, they are temporary. When exiting play mode, the changes are not stored. [54]

The Project window displays all of the files related to the Project and is the primary way for navigating and finding Assets as well as other Project files in the application. [55]

The animator window controls the animations and gives motion to GameObjects; it will be explained further in the section 5.4.1 where there is a description of how the player moves.

4.3 Scripting

Scripting is an essential ingredient in all applications that are made in Unity. Most applications need scripts to respond to input from the player and to arrange for events in the gameplay to happen when they should. Beyond that, scripts can be used to create graphical effects, control the physical behaviour of objects or even implement a custom AI system for characters in the game. [56]

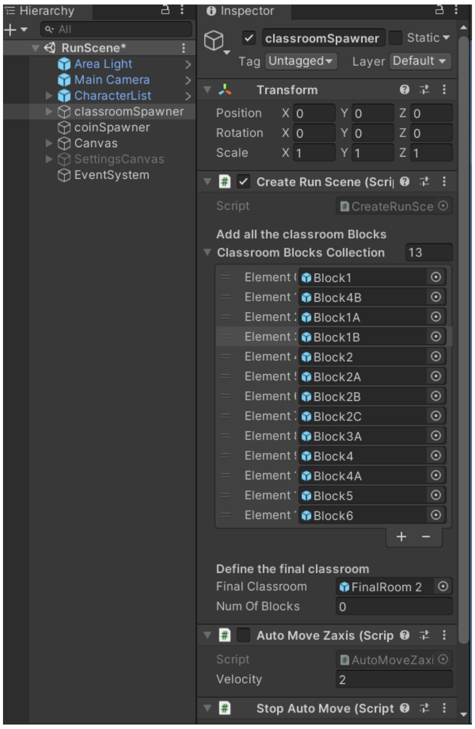
Unity supports the C# programming language natively. Numerous other .NET languages can be used with Unity if they can compile a compatible DLL, however in this diploma thesis, only C# is used.

A script makes its connection with the internal workings of Unity by implementing a class that derives from the built-in class called MonoBehaviour. MonoBehaviour is the base class from which every Unity script derives. It can be considered as a class as a kind of blueprint for creating a new Component type that can be attached to GameObjects. Whenever a script component is attached to a GameObject, it creates a new instance of the object specified by the blueprint. There are two basic functions that are defined inside the class, Start() and Update() functions[57].

MonoBehaviour.Start() The Start() method is used in the majority of the scripts, and it is a MonoBehavior message. Start() is called on the frame when a script is enabled just before any of the Update methods are called the first time.[58]

MonoBehaviour.Update() Update() is called every frame, when the MonoBehaviour is enabled.

As shown in Figure 4.4, the GameObject called classroomSpawner has three different C# scripts. These scripts can make the object move, add more objects in the current Scene, enable or disable scripts and GameObjects and generally modify the game dynamically. Each class/ script inherits the MonoBehaviour for the reasons mentioned above.



☑ classroomSpawner □ Static
Tag Untagged Layer Default

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ Create Run Scene (Scri
Script CreateRunSce

Add all the classroom Blocks
Classroom Blocks Collection 13

Element (Block1
Element ' Block4B
Element : Block1A
Element : Block1B
Element . Block2
Element ! Block2A
Element (Block2B
Element : Block2C
Element { Block3A
Element ! Block4
Element ' Block4A
Element ' Block5
Element ' Block6

Define the final classroom
Final Classroom FinalRoom 2
Num Of Blocks 0

□ Auto Move Zaxis (Scrip
Script AutoMoveZaxi
Velocity 2

Stop Auto Move (Script

Figure 4.4: *Scripts attached to GameObject*

Another widespread class used is ”PlayerPrefs” which stores Player preferences between game sessions. It can store string, float, and integer values into the user’s platform registry. [59]

Chapter 5

The Game

5.1 Game overview

The environment of the game is a school. The game sequence is to answer the question that appears on the whiteboard. After answering the question, the player runs in the school corridors heading to the final classroom. If the answer is correct, green coins appear, and the player should collect as many as possible. If the answer is incorrect, then red coins appear, and the player should avoid collecting them because the score is diminishing. When reaching the end, a new question appears. Answering the questions and then collecting coins repeats until there is no remaining life. A player can select the wrong answer up to three times; only on the fourth false answer, the game is over.

The game's purpose is to answer the given questions correctly in order to be able to score high. The subject that the questions refer to is mathematics, as the game is focused on math questions. Alternatively, there is a potential to add more subjects in the future.

5.2 Game's audience

The game's audience is primary school students and, more specifically, children between six to ten years old.

5.3 Main Menu

The first user interface is the main menu. It has seven basic buttons:

1. select subject,
2. select difficulty,
3. play,
4. settings,
5. stats
6. leader-board, and
7. choose the player character.

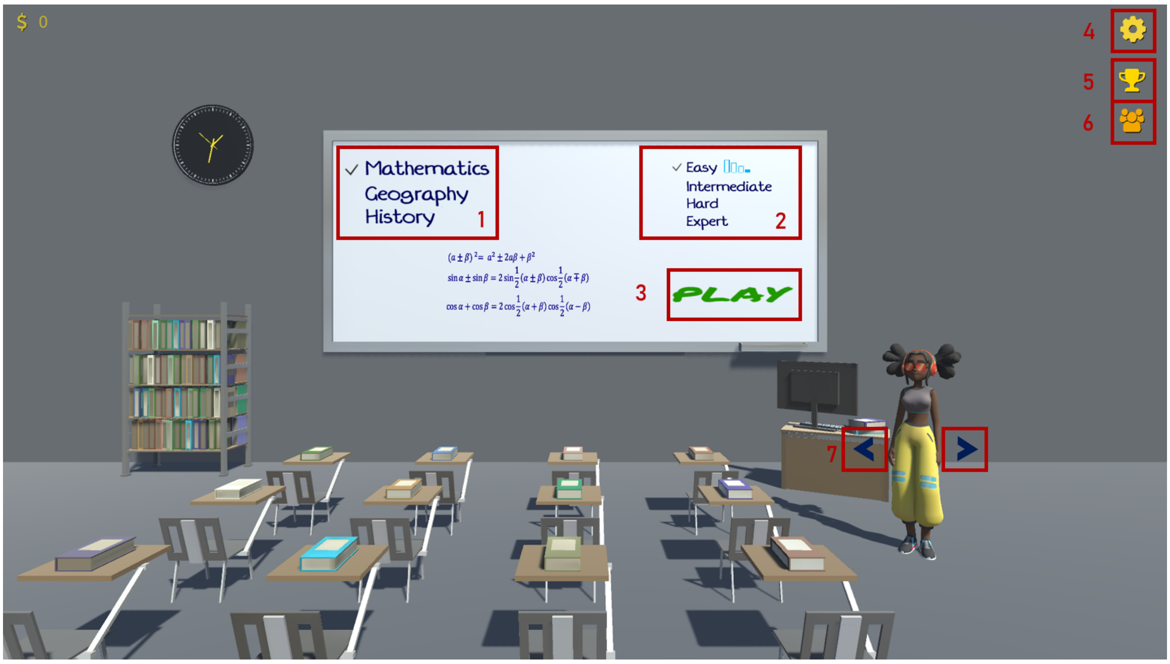

$ 0
✓ Mathematics
Geography
History
1
✓ Easy
Intermediate
Hard
Expert
2
$(\alpha \pm \beta)^2 = a^2 \pm 2a\beta + \beta^2$
$\sin\alpha \pm \sin\beta = 2\sin\frac{1}{2}(\alpha \pm \beta)\cos\frac{1}{2}(\alpha \mp \beta)$
$\cos\alpha + \cos\beta = 2\cos\frac{1}{2}(\alpha + \beta)\cos\frac{1}{2}(\alpha - \beta)$
3
PLAY
4
5
6
7

Figure 5.1: *The main menu and its buttons.*
1. select subject, 2. select difficulty, 3. play , 4. settings, 5. stats, 6.leader - board , 7.choose the player character.

Select subject and select difficulty are toggle groups. Toggle groups allow the user to pick only one component at a time, so pressing one to switch it on automatically switches the others off.[60] Mathematics and the easy level are preselected.

When switching between the subjects, a different image appears. For instance, when selecting Geography, a map appears, whereas when selecting mathematics, equations appear.

The play button starts the game, and the first question comes out on the whiteboard. This game is currently constructed only for Mathematics. Later versions will have other subjects as well. When Mathematics is not selected and the play button is pressed, then a pop-up window appears to inform that this is under construction.

The settings button causes a pop-up window to appear. The user can turn on or turn off the sound in that window.(Figure 5.2)

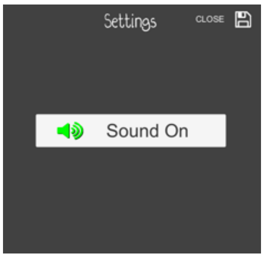
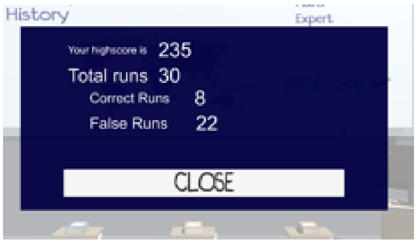


Figure 5.2: *Settings pop up window*

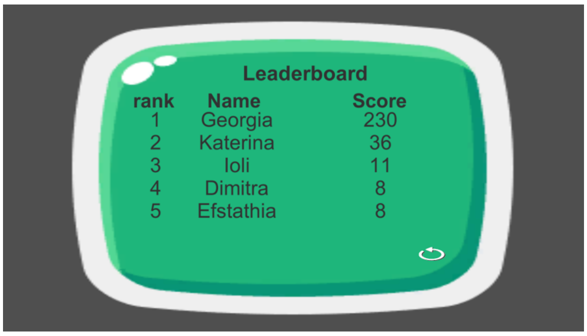


Your highscore is 235
Total runs 30
Correct Runs 8
False Runs 22

Figure 5.3: *Stats pop up window*

The stats button makes a pop-up window emerge as well. In that window, the user can see how many answers, how many correct and false answers have been given, and the high score achieved. (Figure 5.3)

The leaderboard button opens a window that shows the five best scores and their player's name. To submit the player's name, the player should finish the game; otherwise, if the player exits the game earlier, then the current score is registered to the last player's name.



Leaderboard

rank	Name	Score
1	Georgia	230
2	Katerina	36
3	Ioli	11
4	Dimitra	8
5	Efstathia	8

Figure 5.4: *Leaderboard pop up window*

Select character buttons are the arrows on the right and left sides of the character. By pressing these, the character changes, and the player chooses which one is preferred. The boy character, called ”Hermes,” and the girl character, named ”Fay,” are unlocked and free to use. However, the mouse character, called ”Freddie,” is locked, and for unlocking him, the player must spend coins (Figure 5.5). Thus, there is one more button next to the locked character. The player should collect the amount of coins the character costs in the ”wallet” for unlocking. The wallet appears on the left corner of the main menu.

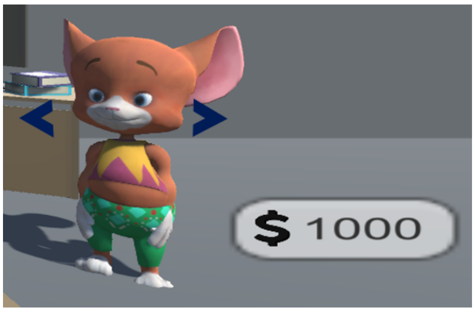


Figure 5.5: *Locked character*

5.4 Game Characters

Hermes, Fay, and Freddie are the characters that the child can use. (Figure 5.6) These characters are downloaded from Mixamo. Mixamo is a free online database that provides 3D characters and a variety of animations for them.[61]

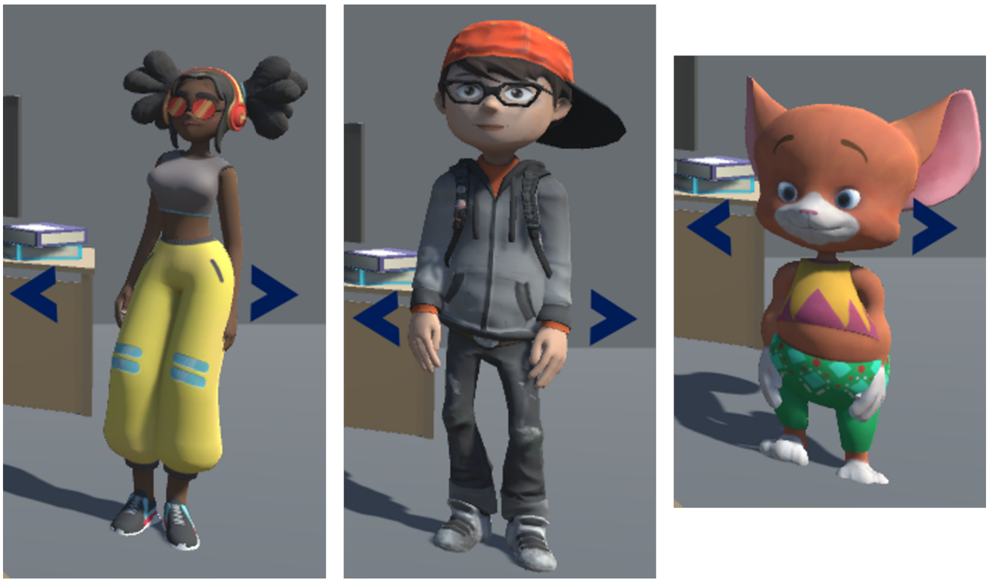


Figure 5.6: *The game characters: Fay, Hermes, Freddie*

5.4.1 Animations and Animator Controller

The characters do not have motion, so numerous animations provide them movement. The animations make the character move, show emotions, and be more realistic.

In order to animate the characters, it is necessary to create an animator controller and attach that to the character in the Unity Editor. The Animator Controller (example in Figure 5.7) has references to the Animation clips used within it and manages the various Animation Clips and the Transitions between them using a ”State Machine,” which could be thought of as a flow-chart of Animation Clips and Transitions, or a simple program written in a visual programming language within Unity. [62]

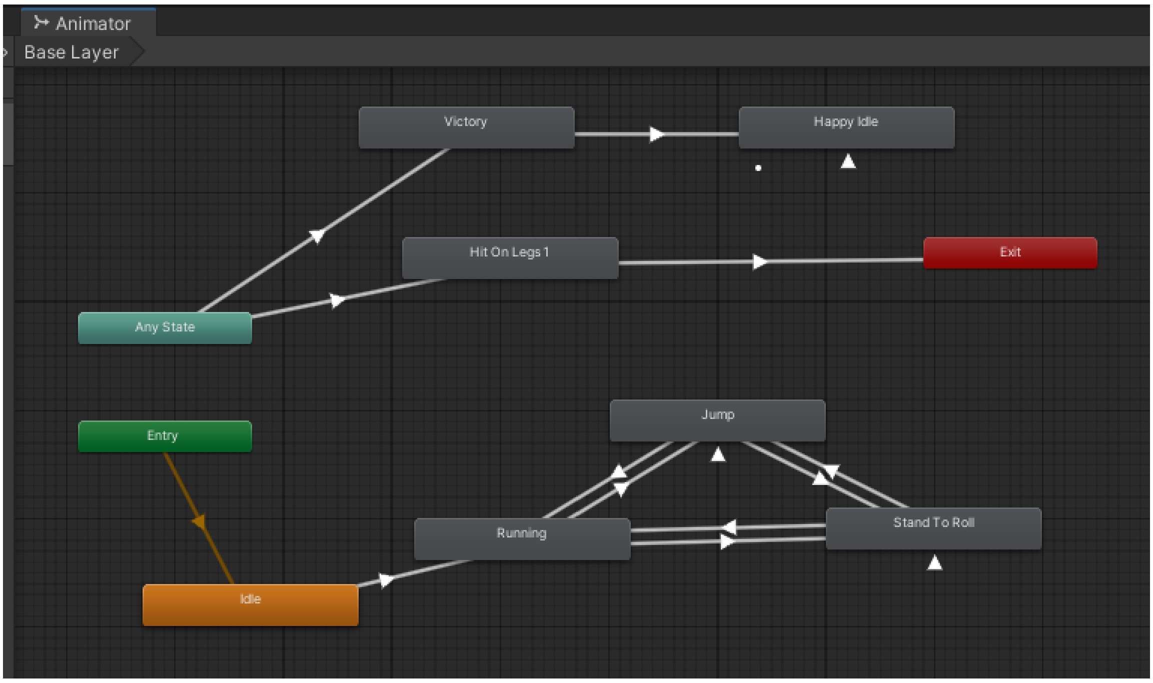
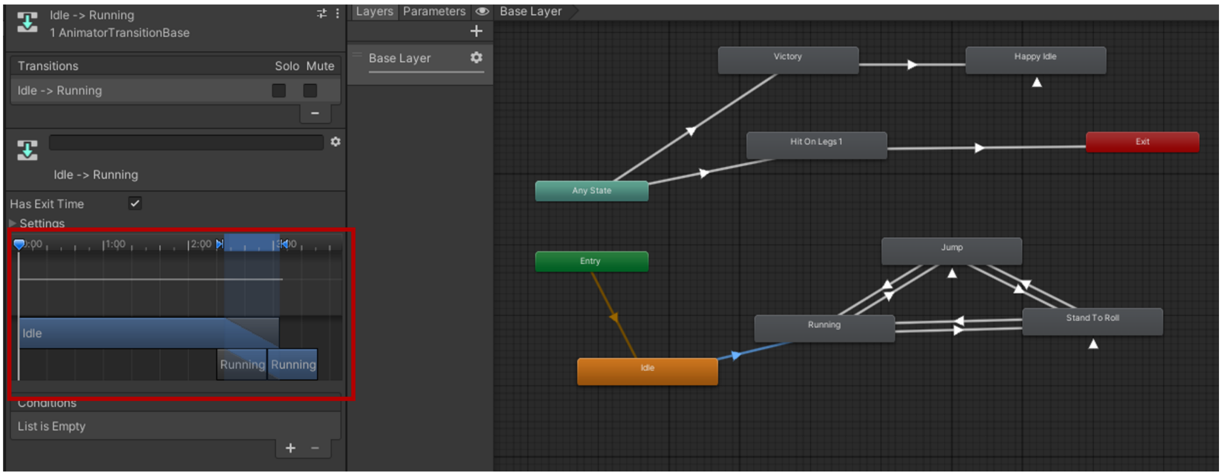


Figure 5.7: *Animator Controller - example of controlling Hermes movement in the game*

Consequently, the first animation that runs is the Idle and then moves to the second, the running. As shown in Figure 5.8, the running animation begins before the idle animation finishes to achieve a smooth transition between these two moves.

Moreover, according to the graph, it is evident that there are transitions between running - jumping, running - rolling, and jumping - rolling. These transitions are triggered in a script. The character runs; if a button to jump or roll is pressed, it proceeds to the corresponding animation and returns to running animation when it finishes.



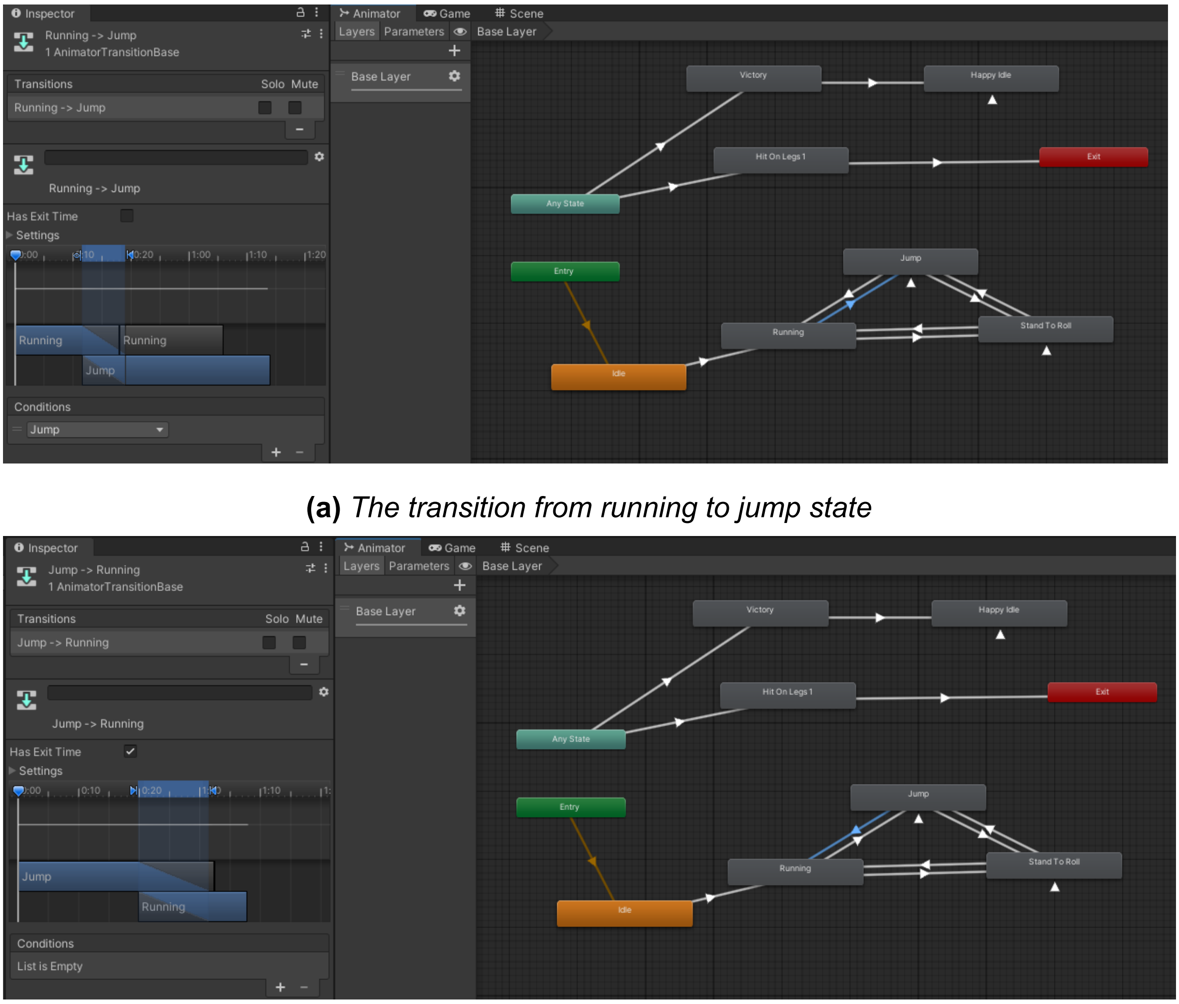
Has Exit Time ☑

List is Empty

Figure 5.8: *Animator Controller example - from Idle state to Running state*

(see Figure 5.9)

Any State is a unique state which is constantly present. It exists for the case where it wants to go to a specific state, regardless of which state it is currently. Returning to the example, when the player reaches the endpoint, a script triggers the victory animation immediately, no matter if the player is running, jumping, or rolling down.



Has Exit Time ☐

(a) *The transition from running to jump state*

Has Exit Time ☑

(b) *The transition from jump to running state*

Figure 5.9: *Animator Controller - Animation transition*

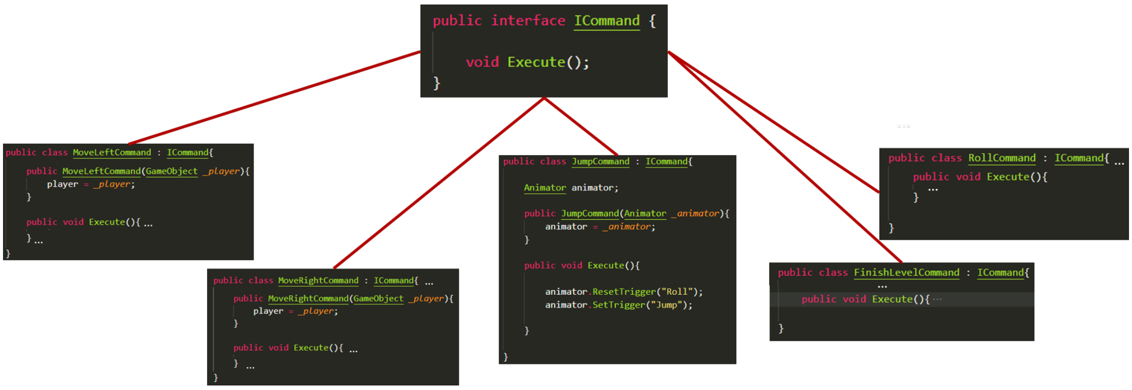
5.4.2 Command Pattern

Design patterns are solutions to general problems that software developers face during software development. Design patterns represent the best approaches used by professional object-oriented software developers. Numerous software developers obtained these solutions by trial-and-error over a rather significant period of time.

In order to control the player's movement and behavior, the command pattern is used. This pattern will be explained by an example. [63]

There is a script attached to the players called ”PlayerController”. There is an interface called ”ICommand” that has an unimplemented function called ”Execute().”

The player makes different movements. Each movement is another class that implements the ”ICommand” interface. Therefore, there is ”JumpCommand”, ”RollCom-



```
public interface ICommand {

    void Execute();
}
```

```
public class MoveLeftCommand : ICommand{

    public MoveLeftCommand(GameObject _player){
        player = _player;
    }

    public void Execute(){ ...
    } ...
}
```

```
public class MoveRightCommand : ICommand{ ...

    public MoveRightCommand(GameObject _player){
        player = _player;
    }

    public void Execute(){ ...
    } ...
}
```

```
public class JumpCommand : ICommand{

    Animator animator;

    public JumpCommand(Animator _animator){
        animator = _animator;
    }

    public void Execute(){

        animator.ResetTrigger("Roll");
        animator.SetTrigger("Jump");

    }

}
```

```
public class RollCommand : ICommand{ ...
    public void Execute(){
        ...
    }

}
```

```
public class FinishLevelCommand : ICommand{
                ...
    public void Execute(){ ···

}
```

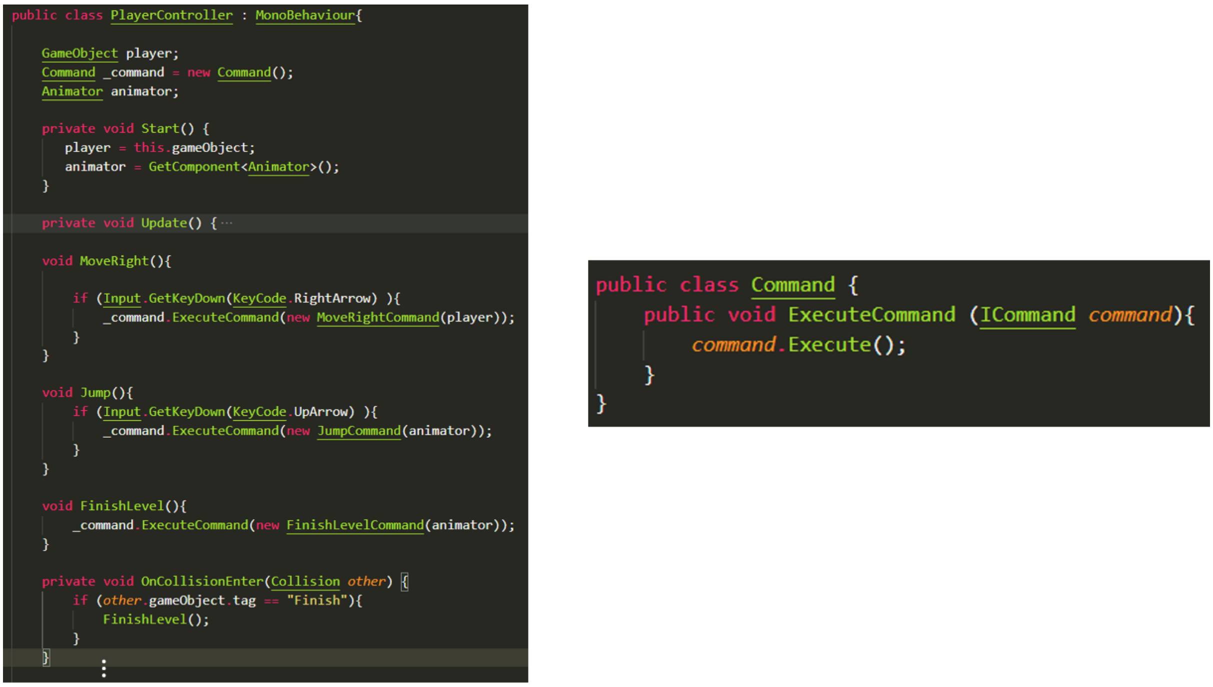
Figure 5.10: *ICommand interface and its implementations*

mand", "MoveRightCommand", "MoveLeftCommand", and "FinishLevelCommand" classes implementing the interface, with a different purpose each time. For example, when the public method "Execute()" is called from "JumpCommand" class, then the Jump animation from the animator file is triggered. On the other hand, when calling the "MoveLeftCommand", in "Execute()" method, the player moves to the left. (see Figure 5.10)

There is a class called Command with a public void function called "ExecuteCommand(ICommand command)" that takes as a parameter the interface "Icommand" and uses the "Execute()" method.

To summarize, the "PlayerController" calls the method from "Command" class, and this method takes as a parameter the interface "ICommand". (see Figure 5.11) Moreover, if the player needs to have a new move, then a new class that implements the interface is created, and then in "PlayerController" it is called using the following command:

_command.ExecuteCommand(new newClassName(parameter));



```
public class PlayerController : MonoBehaviour{

    GameObject player;
    Command _command = new Command();
    Animator animator;

    private void Start() {
        player = this.gameObject;
        animator = GetComponent<Animator>();
    }

    private void Update() {···

    void MoveRight(){

        if (Input.GetKeyDown(KeyCode.RightArrow) ){
            _command.ExecuteCommand(new MoveRightCommand(player));
        }
    }

    void Jump(){
        if (Input.GetKeyDown(KeyCode.UpArrow) ){
            _command.ExecuteCommand(new JumpCommand(animator));
        }
    }

    void FinishLevel(){
        _command.ExecuteCommand(new FinishLevelCommand(animator));
    }

    private void OnCollisionEnter(Collision other) {
        if (other.gameObject.tag == "Finish"){
            FinishLevel();
        }
    }
            ⋮
```

```
public class Command {
    public void ExecuteCommand (ICommand command){
        command.Execute();
    }
}
```

Figure 5.11: *PlayerController class and Command class*

5.5 Game Scenes

The game's purpose is to answer the given questions correctly in order to be able to score high. The subject that the questions refer to is mathematics, but with the potential to add more subjects in the future, we focus on math questions.

The environment of the game is a school. The game sequence is to answer the question that appears on the whiteboard. After answering the question, the player runs in the school corridors until the final classroom or crushing on an obstacle. If the answer is correct, green coins appear, and the player should collect as many as possible. If the answer is incorrect, then red coins appear, and the player should avoid collecting them because the score is reducing. When reaching the end, a new question appears. Answering the questions and then collecting coins repeats until there is no remaining life. A player can select the wrong answer up to three times; the fourth time, the game is over.

There are four levels: easy, intermediate, hard, and expert. The more challenging the level is, the more complex the mathematics equations become, but at the same time, the higher the score will be because the running distance is more.

5.5.1 Quiz scene

The first scene that loads is the Quiz Scene. In this scene, there is a whiteboard with a question - a mathematical expression- and three possible answers. The player should calculate the mathematical expression and choose the correct answer. If it is correct, then a reward bell rings, and the next scene loads with green coins to gather. On the other hand, if the player miscalculates, then a disappointed sound sounds, and the next scene loads with red coins, which the player should avoid.

The questions are of scaled difficulty depending on the selected level. There are four levels: easy, intermediate, hard, and expert. The more challenging the level is, the more complex the mathematics equations become.

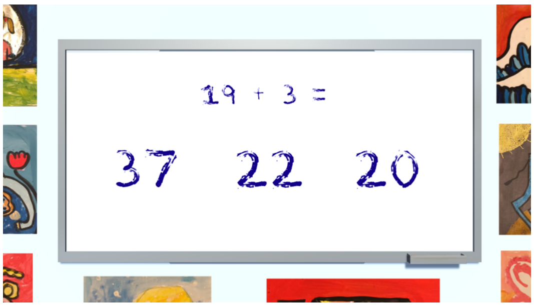


Figure 5.12: *Quiz scene*

The mathematical expression has the form of:

$numberA$ **operator** $numberB \quad =$

The A and B numbers range is directly related to each level. The same is for the mathematical operator; the harder the level, the more difficult operators are possible. Easy level has questions only about addition with A and B number range value from zero to twenty. The intermediate level has questions related to both addition and subtraction. The range is wider than the previous one, and specifically, the student will be familiar with equations between the first one hundred numbers. For the hard level, multiplication is among the available mathematical questions. At this level, the basic calculations, namely addition, and subtraction involve numbers within the zero to one hundred range. As for multiplication, the range of A and B is limited to ten. Ultimately, the last and most challenging level includes division as well. The basic calculations range between zero and one thousand, whereas multiplication and division only involve the first twelve numbers.

Each level with its mathematical expressions and the range for each is displayed in table 5.1.

	+	-	x	/
easy	20			
intermediate	100	100		
hard	500	500	10	
expert	1000	1000	12	12

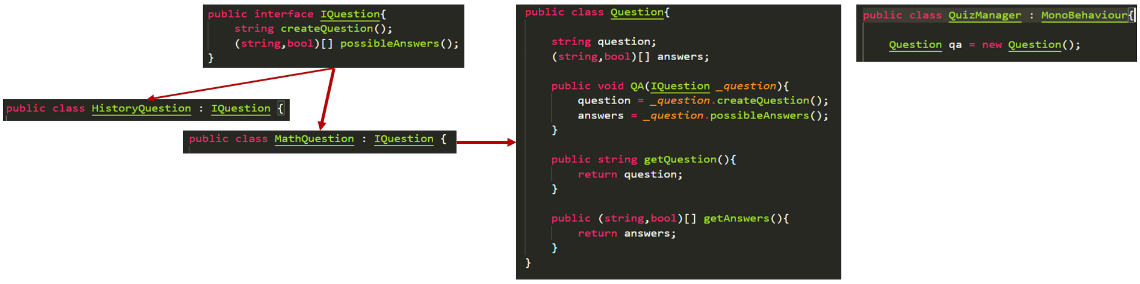
Table 5.1: *Mathematical equations related to levels*

Based on the difficulty, the easy level is appropriate for pre-school or first-grade students in primary school. Intermediate is suitable for first-grade students. The hard level is proper for second-grade students because they learn about multiplication at that age. The expert level fits third-grade pupils because they learn about division and are familiar with more complex equations in that grade.

Command pattern for creating the quiz

The command pattern described in section 5.4.2 is also applied to create the questions.

The IQuestion class is an interface with two fields, one is about the question, and the other is a table with the possible answers. MathQuestion class inherits IQuestion. A class called Question also takes as parameters in the constructor an IQuestion object, and it has two public methods that are used in QuizManager class. QuizManager is responsible for assigning each possible answer to each field using a Random algorithm and informing the next scene if the player answers correctly or incorrectly. Quiz Manager handles the created questions and the possible answers, no matter what type of questions are, thus if it is about maths or geography, etc.



```
public interface IQuestion{
    string createQuestion();
    (string,bool)[] possibleAnswers();
}
```

```
public class HistoryQuestion : IQuestion {
```

```
public class MathQuestion : IQuestion {
```

```
public class Question{

    string question;
    (string,bool)[] answers;

    public void QA(IQuestion _question){
        question = _question.createQuestion();
        answers = _question.possibleAnswers();
    }

    public string getQuestion(){
        return question;
    }

    public (string,bool)[] getAnswers(){
        return answers;
    }
}
```

```
public class QuizManager : MonoBehaviour{

    Question qa = new Question();
```

Figure 5.13: *Command pattern for creating the quiz*

5.5.2 Run scene

The run scene consists of some blocks (Figure 5.14), and the final run scene is made by combing the blocks. The higher levels have more blocks than the lower ones. Specifically, the first level has three blocks, the second has five blocks, the next has ten blocks, and the last has twenty blocks.

The game has twenty different blocks, and they are combined randomly to create the run scene each time. The first and the final block are the same every time.

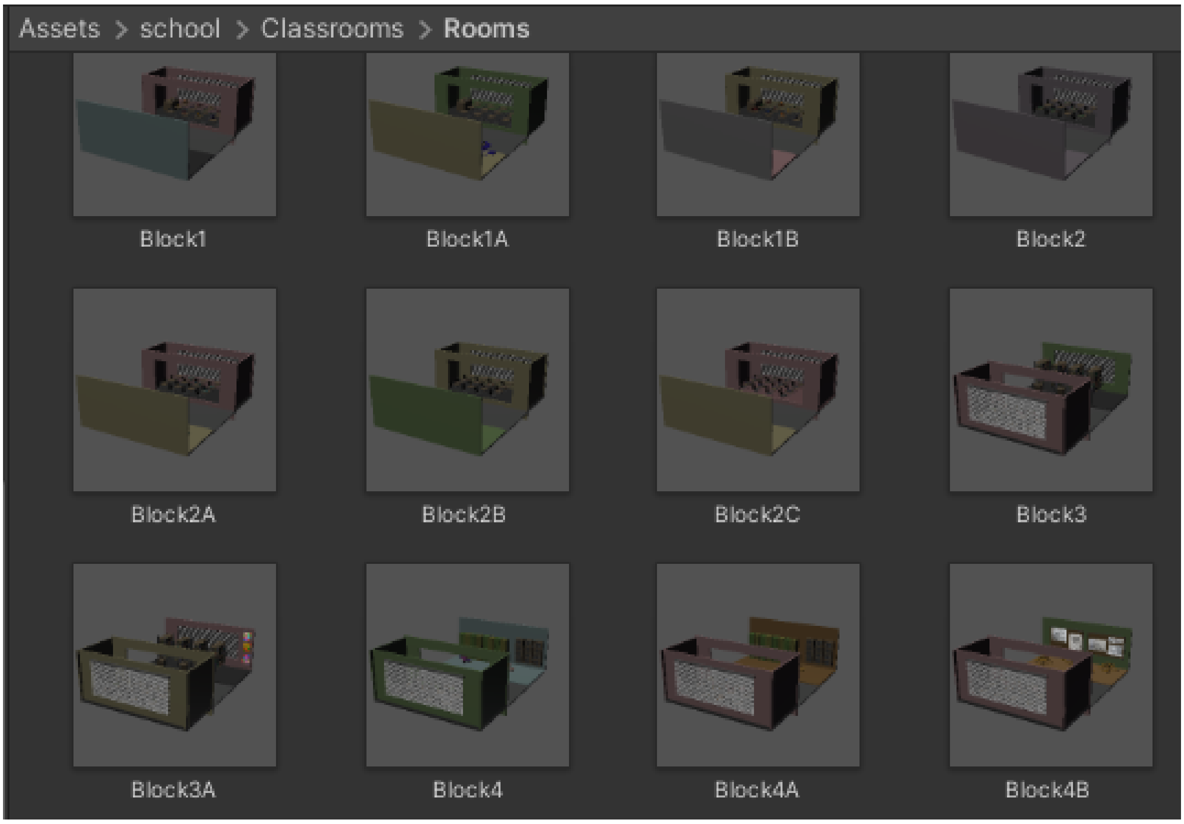


Figure 5.14: *A sample of the classroom blocks*

Each block is designed uniquely using assets from Unity's asset store. The main asset package for the game’s appearance is the ”school assets.” It includes the doors, the boards, the books, the school bus, the chairs, the computers, the lockers, and the tables. [64]. There are some complimentary assets to enhance the game like the globe [65], the stationary (pens, pencils, scissors, etc.) [66], the chess [67], trash bins [68].

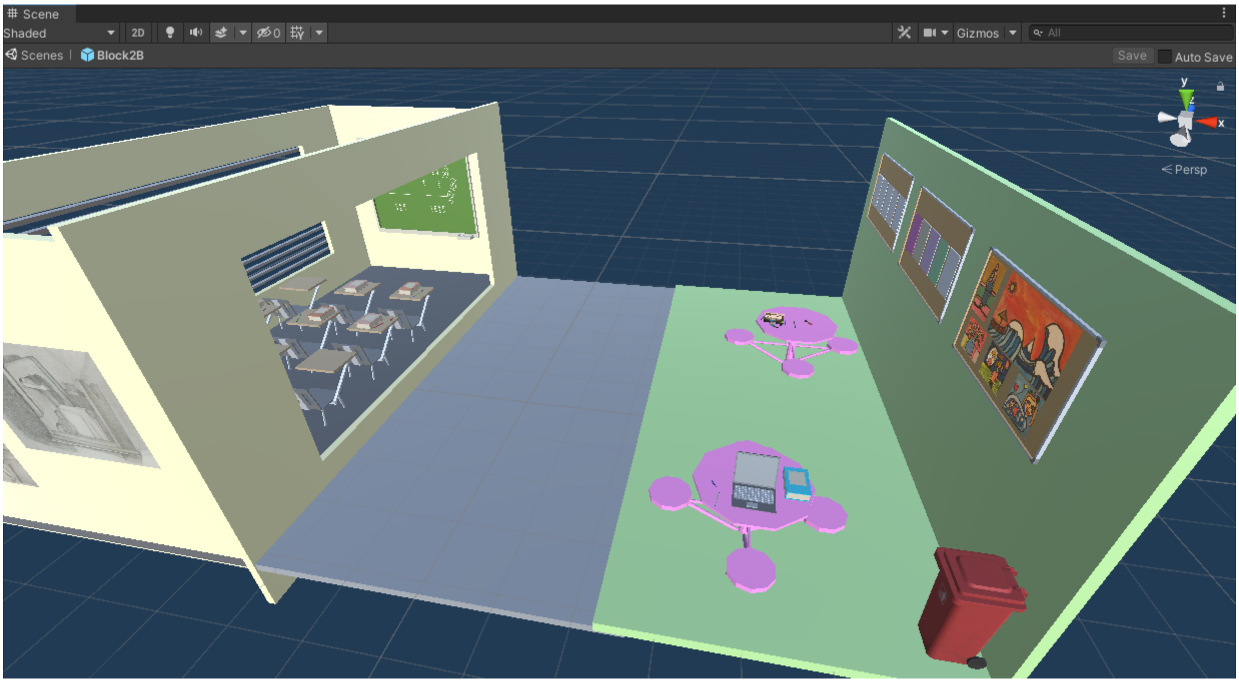


Figure 5.15: *A random classroom*

The coin is custom made in Blender, an open-source 3D creation suite [69]. The student’s answer determines its color—the color changes with the help of a script. The coin is green when the answer is correct; otherwise, it is red. The score rises when collecting green coins and diminishes when collecting red ones. Each level defines the value of each coin; the easier the level, the lesser its coin value is. Therefore, for the first level, the green coin value is one while the red one has no negative value because that level is for little kids, as mentioned before. The correct coin value is two in the intermediate level, while the score is reduced by one for every false coin. In the hard level, the green coin adds five points simultaneously, while the red one removes two points from the current score. The last and most challenging level has the highest reward, ten points for each green coin, but at the same time, it has the highest penalty, three points for each red coin collected. (Table 5.2)

There are also coin blocks, Figure 5.16. Each block contains coins within a pat-

coin	green	red
easy	+1	0
intermediate	+2	-1
hard	+5	-2
expert	+10	-3

Table 5.2: *Coin value per level*

tern, such as a row with coins on the left side. There are seven different coin blocks. The amount of coin blocks used in each run scene is interconnected with levels. In each coin block, except for coins, there are some obstacles the player should avoid, either by changing lines or by jumping above them. If the player stumbles on the obstacle, then the the quiz scene appears, and the child should answer the new question.

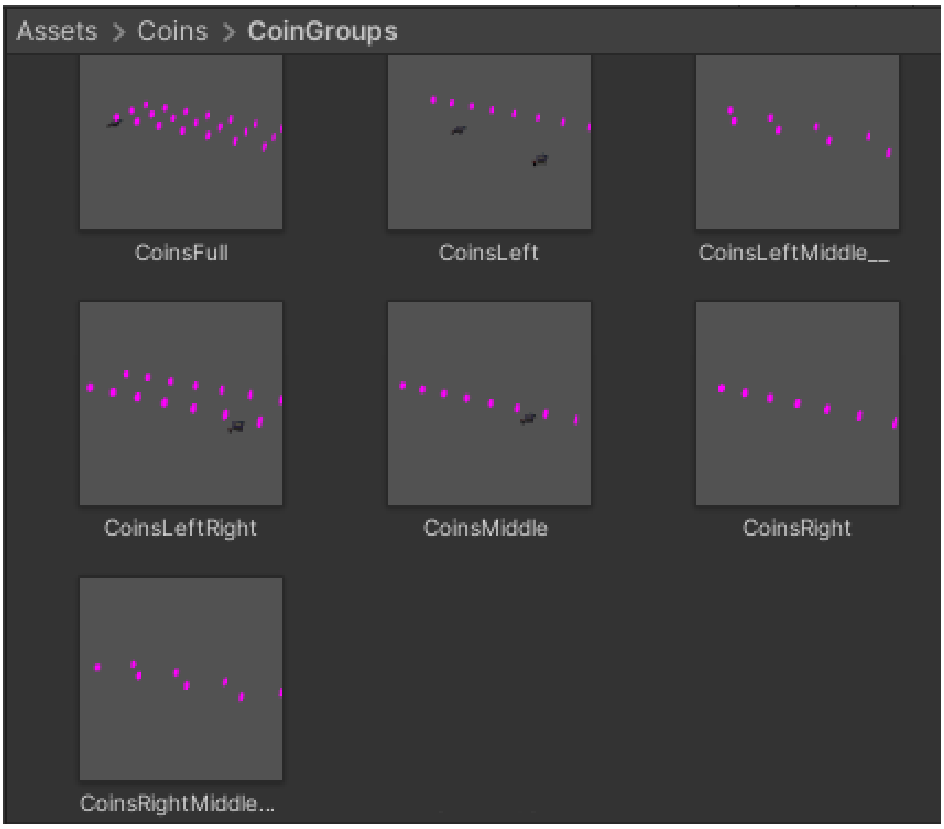


Figure 5.16: *The coin blocks*

To sum up, the chosen level determines how long the run will be, the number of blocks, the number of coin blocks, and the coin value.

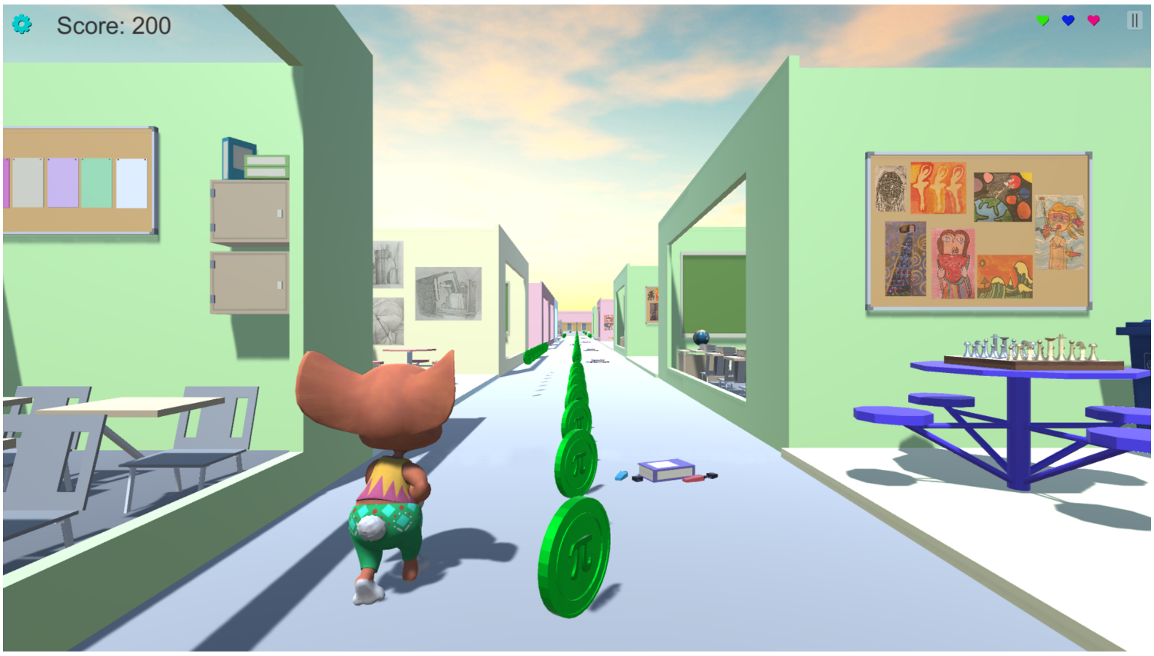


Figure 5.17: *A screenshot of the Run Scene*

In this scene, the player should be alert and active; as a result, it is essential to be able to pause the game. There is a pause button on the right corner, which pauses the game, and the character freezes.

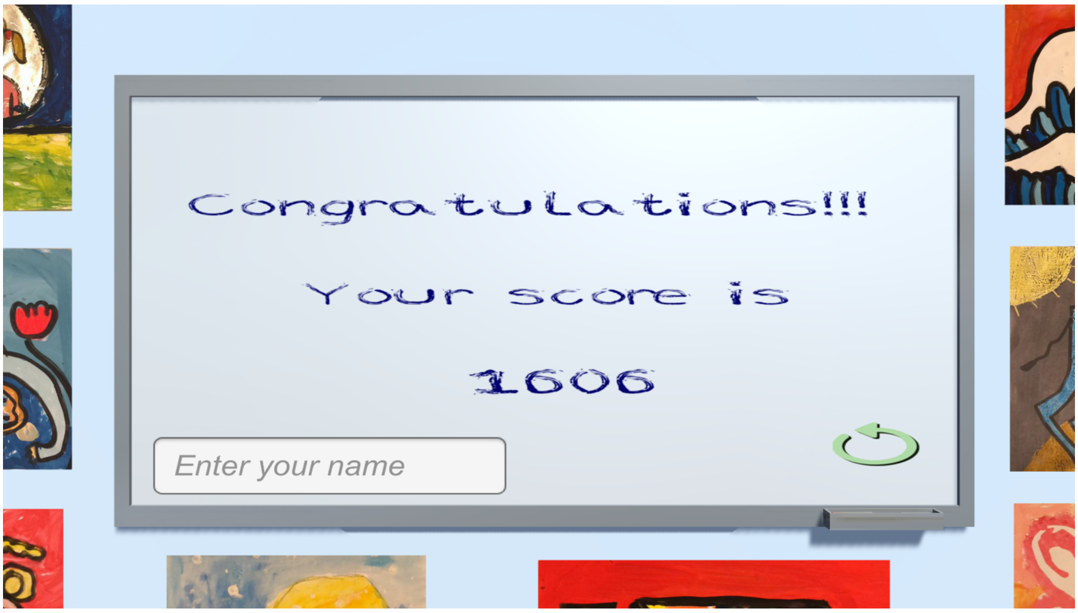
There is the setting button on the left corner. When clicking this, a pop-up window appears. The player can mute or unmute the game in that window or return to the main menu immediately.

The score appears on the left corner, and it is updated every time a coin is collected.

5.5.3 Game's end

The child answers the question and then tries to collect as many points as possible or to lose as few points as possible when answering the given question incorrectly. However, they can not answer wrong multiple times. Every time that there is a wrong answer, they lose a life. There are three lives available; the game is over the fourth time there is a wrong answer.

After that, the score appears on the screen. At the same time, the player should


Congratulations!!!

Your score is

1606

Figure 5.18: *End Game Scene*

register their name in the input field on the bottom side of the screen. The score is attached to the name, and if the score is among the five best scores, it is stored in the leader-board.

On the right bottom corner, there is a button that returns to the main menu.

Chapter 6

Conclusion

6.1 Summary

The educational system, which is the focal lens of the present thesis, beginning from kindergarten to elementary school and high school, is characterised by multiple options to adopt various contemporary approaches in their programs and curriculum. The use exclusively of traditional methods tends to become obsolete, and the teachers start embracing more student-centered methods to be a part of their teaching, as it has been observed to be more effective. Active learning makes the students be the center of the learning procedure and participate actively during the lesson, and the pupils not just are passive recipients of the information.

There are many ways to implement active learning by utilizing, at the same time, the technology and the digital era which characterizes these years. This thesis focuses on how digital games, specifically serious learning games, can contribute to the educational process, keep the students engaged in their learning and help them fulfill their academic goals.

To elaborate on this thesis, a learning game is developed using the Unity game engine. Specifically, it is a math game with an orientation toward the pupils of the first grades of elementary school. This game can be used both during the school program and at home using a smartphone or tablet. It aims to familiarize the students with basic algebra operations and to entertain them simultaneously. It contains numerous game elements to make it more appealing and interesting to the young pupils, like avatars, locked avatars, levels, leaderboards, scores, etc. Moreover, it has a pleasant background and sounds for instant feedback that informs whether they increase or decrease their score. Generally, it is a children-friendly game, that allows them to understand it and play it easily, but meanwhile it educates them.

6.2 Future work

Taking into consideration the fact that there are innumerous games in the market, the present game can be improved and developed even more in order to compete with the other games and learning platforms. The upgrade includes both the aesthetics and the game functionality.

First and foremost, the game may consists of more subjects than mathematics, like geography, history, etc. Generally, any subject can have multiple choice questions and these questions are separated into different levels based on their difficulty. The game is designed in a way that allows the addition of new subjects to be easy by only implementing one more script/class with the new category of questions.

Additionally, the game can allow the players to connect with their classmates and have a shared leaderboard, and aim to climb higher. The teacher might be able to attend to the students' trials, how many correct and false answers they have given, and their overall performance.

Furthermore, the graphics can be enhanced. This current version of the game uses almost free assets that the Unity store and other platforms offer. So, some assets can be altered in order to have a better aesthetic coherence. Moreover, the game character collection can expand, so they can unlock and choose from a wider range of characters and outfits.

One additional feature that the game may offer is to upload a drawing. Specifically, the game's environment is decorated with different paintings, so the child can upload a personal painting and then watch it decorating the game. This helps to explore and develop their artistic side and feel their contribution to the game.

Last but not least, children should test the game. The game is designed for them, and it is important to get feedback straight from them. Children always express themselves and say what they like and what they do not. They have the imagination to provide us with ideas that we have not even pictured yet. Their help will be valuable for the future version of the game application.

Bibliography

[1] Joseph Johnson. *Percentage of global population accessing the internet from 2005 to 2021, by market maturity*. Jan. 2022. URL: https://www.statista.com/statistics/209096/share-of-internet-users-in-the-total-world-population-since-2006/.

[2] S. O'Dea. *Number of smartphone subscriptions worldwide from 2016 to 2027*. Feb. 2022. URL: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[3] Erin Duffin. *Share of K-12 students in the United States who use digital learning tools daily in 2019, by school level*. Jan. 2022. URL: https://www.statista.com/statistics/1076292/share-k-12-students-us-who-use-digital-learning-tools-daily-level/.

[4] *Digital Media Report 2021 – Video Games*. Dec. 2021. URL: https://www.statista.com/study/39310/video-games/.

[5] *Most influential theories of learning*. [Accessed on: 15 Jan 2022]. Feb. 2016. URL: http://www.ibe.unesco.org/en/geqaf/annexes/technical-notes/most-influential-theories-learning.

[6] *Teaching Methods*. [Accessed on: 15 Feb 2022]. URL: https://teach.com/what/teachers-know/teaching-methods/#teachercentered.

[7] Mingorance Estrada et al. "Flipped Classroom to Improve University Student Centered Learning and Academic Performance." In: *Social Sciences* 8.11 (Nov. 2019), p. 315. DOI: 10.3390/socsci8110315. URL: https://doi.org/10.3390/socsci8110315.

[8] *The Complete List of Teaching Methods*. [Accessed on: 15 Feb 2022]. July 2021. URL: https://onlinedegrees.sandiego.edu/complete-list-teaching-methods/.

[9] *What Is Differentiated Instruction? | Scholastic*. [Accessed on: 15 Feb 2022]. URL: https://www.scholastic.com/teachers/articles/teaching-content/what-differentiated-instruction/..

[10] Edgar Dale. *Audiovisual methods in teaching*. eng. 3. ed. New York: Dryden Press, 1969. ISBN: 9780039100544.

[11] *What is Active Learning?* [Accessed on: 15 Feb 2022]. URL: https://www.queensu.ca/teachingandlearning/modules/active/04_what_is_active_learning.html.

[12] *Collaborative Learning*. URL: https://teaching.cornell.edu/teaching-resources/active-collaborative-learning/collaborative-learning.

[13] *Getting started with Active Learning*. [Accessed on: 15 Feb 2022]. URL: https://www.cambridge-community.org.uk/professional-development/gswal/index.html.

[14] Louis Deslauriers et al. “Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom.” In: *Proceedings of the National Academy of Sciences* 116.39 (Sept. 2019), pp. 19251–19257. DOI: 10.1073/pnas.1821936116. URL: https://doi.org/10.1073/pnas.1821936116.

[15] Marc Prensky. “Digital game-based learning.” In: *Computers in Entertainment* (Oct. 2003). DOI: 10.1145/950566.950596. URL: https://doi.org/10.1145/950566.950596.

[16] British Museum. *Top 10 historical board games*. https://blog.britishmuseum.org/top-10-historical-board-games/. (Accessed on 03/01/2022). Feb. 2021.

[17] The Editors of Encyclopaedia Britannica. *go*. https://www.britannica.com/topic/go-game. Mar. 2017.

[18] The Editors of Encyclopaedia Britannica. *backgammon*. https://www.britannica.com/sports/backgammon. Sept. 2011.

[19] The Editors of Encyclopaedia Britannica. *checkers*. Mar. 2018.

[20] HISTORY.COM EDITORS. *Video Game History*. https://www.history.com/topics/inventions/history-of-video-games. Sept. 2017.

[21] Ralf Dörner et al. “Introduction.” In: *Serious Games*. Springer International Publishing, 2016, pp. 1–34. DOI: 10.1007/978-3-319-40612-1_1. URL: https://doi.org/10.1007/978-3-319-40612-1_1.

[22] Stefan Göbel. “Serious Games Application Examples.” In: *Serious Games*. Springer International Publishing, 2016, pp. 319–405. DOI: 10.1007/978-3-319-40612-1_12. URL: https://doi.org/10.1007/978-3-319-40612-1_12.

[23] *Formal, non-formal and informal learning*. (Accessed on 16/03/2022). URL: https://www.coe.int/en/web/lang-migrants/formal-non-formal-and-informal-learning.

[24] Sebastian Deterding et al. “From Game Design Elements to Gamefulness: Defining Gamification.” In: vol. 11. Sept. 2011, pp. 9–15. DOI: 10.1145/2181037.2181040.

[25] Philipp Busch. *Gamification and Serious Games*.

[26] Kevin Werbach and Dan Hunter. *For the win*. Philadelphia, PA: Wharton Digital Press, Oct. 2012.

[27] Bryant Nielson. *Gamification Mechanics vs. Gamification Dynamics*. Apr. 2017. URL: https://www.yourtrainingedge.com/gamification-mechanics-vs-gamification-dynamics/.

[28] Kevin Werbach. *The Pyramid of Elements*. Coursera. 2022. URL: https://www.coursera.org/learn/gamification/lecture/JPQNK/4-2-the-pyramid-of-elements.

[29] *Badges – Special Game Awards*. (Accessed on 01/04/2022). URL: https://developer.roblox.com/en-us/articles/Badges-Special-Game-Awards.

[30] Kevin Werbach. *The PBL Triad*. Coursera. 2022. URL: https://www.coursera.org/learn/gamification/lecture/9Ylm2/4-3-the-pbl-triad.

[31] Fiona Fui-Hoon Nah et al. “Effects of the Use of Points, Leaderboards and Badges on In-Game Purchases of Virtual Goods.” In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 525–531. DOI: 10.1007/978-3-319-20895-4_48. URL: https://doi.org/10.1007/978-3-319-20895-4_48.

[32] Luca Galli, Piero Fraternali, and Alessandro Bozzon. “On the application of game mechanics in information retrieval.” In: *Proceedings of the First International Workshop on Gamification for Information Retrieval - GamifIR ’14*. ACM Press, 2014. DOI: 10.1145/2594776.2594778. URL: https://doi.org/10.1145/2594776.2594778.

[33] *Leaderboards*. (Accessed on 01/03/2022). URL: https://developers.google.com/games/services/common/concepts/leaderboards.

[34] *Applications of Mathematics*. URL: https://mathigon.org/applications.

[35] Benoit B. Mandelbrot. *Adventures into the Mathematical Forest of Fractal Trees*. May 2014. URL: https://blog.wolfram.com/2014/05/22/adventures-into-the-mathematical-forest-of-fractal-trees/.

[36] Grace Fleming. *Why Math Is More Difficult for Some Students*. Aug. 2020. URL: https://www.thoughtco.com/why-math-seems-more-difficult-for-some-students-1857216.

[37] Lara Bertram. “Digital Learning Games for Mathematics and Computer Science Education: The Need for Preregistered RCTs, Standardized Methodology, and Advanced Technology.” In: *Frontiers in Psychology* 11 (Oct. 2020). DOI: 10.3389/fpsyg.2020.02127. URL: https://doi.org/10.3389/fpsyg.2020.02127.

[38] *Arcademics*. URL: https://www.arcademics.com.

[39] Celeste Lebowitz. *Arcademic Skill Builders*. May 2020. URL: https://gdc.unicef.org/resource/arcademic-skill-builders.

[40] *Arcademics Success*. URL: https://www.arcademics.com/success.

[41] *Minecraft official site*. URL: https://www.minecraft.net/en-us.

[42] *Sandbox*. June 2017. URL: https://www.techopedia.com/definition/3952/sandbox-gaming.

[43] *Minecraft*. URL: https://minecraft.fandom.com/wiki/Minecraft.

[44] *What is Minecraft?* The Washington Post. URL: https://www.washingtonpost.com/lifestyle/kidspost/what-is-minecraft/2013/03/14/98c54514-8a57-11e2-a051-6810d606108d_story.html#comments.

[45] Simon Bourdeau, Thibaut Coulon, and Marie-Claude Petit. “Simulation-Based Training via a “Readymade” Virtual World Platform: Teaching and Learning With Minecraft Education.” In: *IT Professional* 23.2 (2021), pp. 33–39. DOI: 10.1109/MITP.2021.3062935.

[46] *Minecraft Dungeons vs. Minecraft – 10 differences*. May 2020. URL: https://www.eneba.com/blog/minecraft-dungeons-vs-minecraft-10-differences/.

[47] *COORDINATE SYSTEM TUTORIAL*. URL: https://education.minecraft.net/en-us/lessons/coordinate-system-tutorial.

[48] *The Bunny Times*. URL: https://www.nctm.org/Classroom-Resources/Illuminations/Interactives/Bunny-Times/.

[49] Udemy Team. *What is the Unity Game Engine?* [Accessed on: 18 Jan 2022]. Sept. 2021. URL: https://blog.udemy.com/unity-game-engine/.

[50] Unity Technologies. *Unity*. [Accessed on: 1 Feb 2022]. URL: https://unity.com/.

[51] *The Hierarchy window*. URL: https://docs.unity3d.com/Manual/Hierarchy.html.

[52] *The Inspector window*. URL: https://docs.unity3d.com/Manual/UsingTheInspector.html.

[53] *The Scene View*. URL: https://docs.unity3d.com/Manual/UsingTheSceneView.html.

[54] *The Game View*. URL: https://docs.unity3d.com/Manual/GameView.html.

[55] *The project window*. URL: https://docs.unity3d.com/Manual/ProjectView.html.

[56] *Scripting*. URL: https://docs.unity3d.com/Manual/ScriptingSection.html.

[57] *Creating and using scripts*. URL: https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html.

[58] *MonoBehaviour.Start()*. URL: https://docs.unity3d.com/ScriptReference/MonoBehaviour.Start.html.

[59] *PlayerPrefs*. URL: https://docs.unity3d.com/ScriptReference/PlayerPrefs.html.

[60] *Toggle Group*. URL: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-ToggleGroup.html.

[61] *Mixamo*. URL: https://www.mixamo.com/#/.

[62] *Animator Controller*. URL: https://docs.unity3d.com/Manual/class-AnimatorController.html.

[63] Robert Nystrom. *Game Programming Patterns*. Genever Benning, 2014, pp. 21–31. ISBN: 0990582906.

[64] A.R.S|T. *School assets*. Version 2. May 2021. URL: https://assetstore.unity.com/packages/3d/environments/school-assets-146253#releases.

[65] Viverna. *School Globe*. Version 1.0. Aug. 2019. URL: https://assetstore.unity.com/packages/3d/props/interior/school-globe-152072#publisher.

[66] Sten Ulfsson. *Office Supplies Low Poly*. Version 1.01. Jan. 2018. URL: https://assetstore.unity.com/packages/3d/props/office-supplies-low-poly-105519.

[67] DevDen. *PBR Chess Board Vol - 1 (Simple)*. Version 1.0. May 2020. URL: https://assetstore.unity.com/packages/3d/props/pbr-chess-board-vol-1-simple-158937.

[68] SpaceZeta. *Plastic Trash Bins*. Version 1.0. Mar. 2021. URL: https://assetstore.unity.com/packages/3d/props/exterior/plastic-trash-bins-160771.

[69] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: https://www.blender.org/about/.