Universidade do Minho
Escola de Engenharia
Departamento de Informática

Francisco Manuel Vital Saraiva

Building A Game With Augmented Reality

For Training Computational Thinking

April 2022

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Francisco Manuel Vital Saraiva

Building A Game With Augmented Reality

For Training Computational Thinking

Master's dissertation
in Informatics Engineering

Dissertation supervised by
Professor Pedro Rangel Henriques
Lázaro Vinícius de Oliveira Lima

April 2022

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Francisco Manuel Vital Saraiva

____

ACKNOWLEDGEMENTS

For the creation of this work, and in all of my academic experience, I would like to thank the following people and entities. I'd like to thank the University of Minho and the head of the master's degree in Informatics Engineering and it's professors for the time of learning and experience. I'd like to thank my professor Pedro Rangel Henriques, and Lázaro Vinícius de Oliveira Lima for the guidance and help given over the supervision of this project, always present and guiding with advice and care. I'd like to thank my fellow colleagues who've I've worked over the course of my masters namely: Rui Santos, André Coutinho, César Magalhães, Francisco Freitas, Pedro Freitas and Inês Alves, thank you for your help, experience, your sympathy and most importantly friendship. I'd like to deeply thank my family for always worrying about me, and most importantly my parents Joaquim and Rosa, who always encourage me to never give up and to fight for a better future, I am eternally thankful. I'd like to thank my girlfriend Carolina, who has helped me pull through many times with her love and care, I love you. I'd like to thank my grandparents Noémia, Lídia and Francisco, who are no longer here, but forever are in spirit and to my grandfather José, thank you for your love. And lastly and no less important to António and Celeste, who are also no longer present, but their spirit is deep within me, thank you dearly for being like my family for many years.

ABSTRACT

Computational Thinking is the problem solving method based on picking a complex problem, understanding what the problem is and developing solutions in a way that a computer or human could solve it. To aid in this task Augmented Reality provides a more visual and interesting way of presenting problems to subjects, that will solve puzzles presented as computational inputs. This work is composed of a case study of Augmented Reality and their uses and practices, Computational Thinking definitions and importance in society, as well as games and puzzles used as a training and skill developing tools among students. Based on the gathered data a proof of concept tool was developed to test and create an engaging game to retain the subjects interest to train Computational Thinking. The project developed tackled various technologies that together provided a fun and interactive game that developed Computational Thinking skills with positive feedback on the use of the Augmented Reality.

Keywords: Augmented Reality, Computational Thinking, Learning Resources, Video Game, Game Based Learning

RESUMO

Pensamento Computacional é o método de resolução de problemas baseado em escolher um problema complexo, entender qual o problema e desenvolver soluções de uma maneira que um computador ou humano podem resolver. Para ajudar nesta tarefa a Realidade Aumentada oferece uma mais visual e interessante maneira de apresentar problemas a indivíduos que irão resolver puzzles apresentados como inputs computacionais. Este trabalho é composto de um caso de estudo sobre Realidade Aumentada e os seus usos e práticas, Pensamento Computacional, as suas definições e a importância na sociedade, como também jogos e puzzles usados como ferramentas de desenvolvimento e treino de aptidão entre estudantes. Com base nos dados recolhidos uma ferramenta de prova de conceito foi desenvolvida para testar e criar uma experiência atraente para reter o interesse de indivíduos para treinar o Pensamento Computacional. O projeto foi desenvolvido enfrentado várias tecnologias que juntas ofereceram um jogo divertido e interativo para desenvolver o Pensamento Computacional com um feedback positivo sobre o uso da Realidade Aumentada.

Palavras-Chave: Realidade Aumentada, Pensamento Computacional, Recursos de Aprendizagem, Jogo De Vídeo, Aprendizagem Baseada em Jogos

CONTENTS

LIST OF FIGURES

1

INTRODUCTION

Computational Thinking, simply put, is the action of "thinking like a computer" (Wing, 2014), but in detail refers to the process of solving problems taking knowledge and computational practices such as systematization, representation, analysis, solving and interpreting problems. It is the activity that computers, be it humans or machines, process thought to formulate problems and find solutions. The focus in Computational Thinking is not about what to think, but yes how to think when tackling problems. Taking into account every day activities, with certain characteristics coming into greater play as Abstraction, is a great and widely used process in computer science and software development to focus on important details of objects or systems and one of the basis of Computational Thinking.

Augmented Reality (Berryman, 2012; Azuma, 1997), is a variation of Virtual Reality in which it allows the user to see the real world but with virtual objects imposed onto the real world as to integrate the digital into the real, in other words, augmenting the reality we see as in example of Figure 1.

Augmented Reality in study by Azuma (1997) had seen numerous potential in it's uses in technology be it medical, manufacturing, repairs, entertainment, planning and military which today, years later, does see much more use in our society (Bell, 2014; McFadden, 2020) such as in DIY (Do It Yourself) car repairs, cooking aids, GPS navigation, Video Games, medical training, car mechanics, customer service, military airplanes, television broadcasts and even house interior modeling.

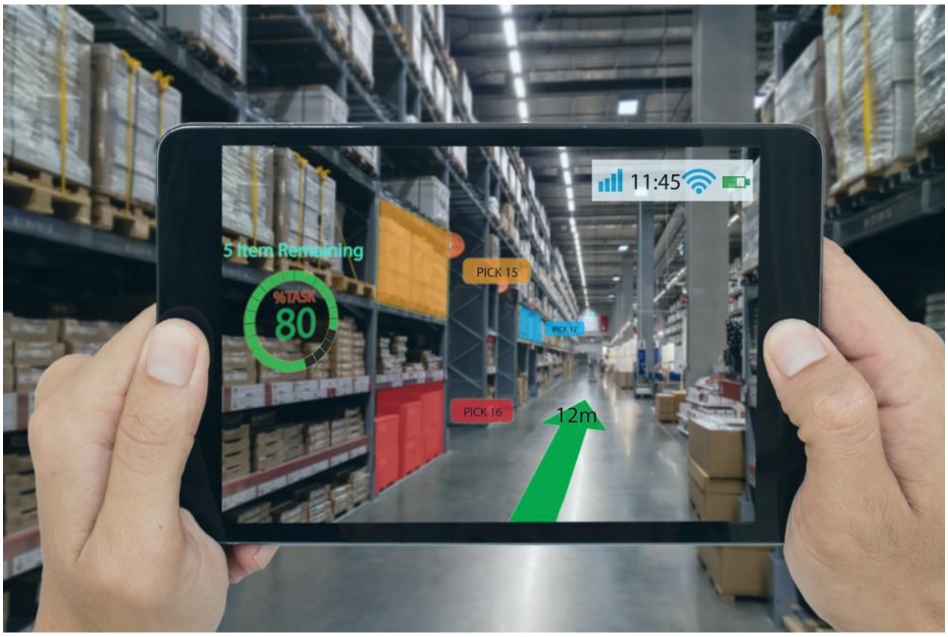


Figure 1: An example of Augmented Reality in a tablet. (Retrieved from forbes.com)

1.1 MOTIVATION

Computational Thinking is an important set of methods to solve problems in various settings and fields, but it has been a topic of study and research on the benefits of training and teaching these skills to students in their education. On a review on the importance of Computational Thinking in K-12 by Grata and Roche (2020), in the subjects of mathematics and science, Computational Thinking aids students to learn to understand concepts and processes of solving problems. Also in the subjects of language and history benefits are seen as students need to focus on the conceptualization and abstraction of data and creativity. These skills being trained from early will give added benefits later in their professional activities and life.

To gauge popularity in the research field over the topic of Computational Thinking a search on Scopus was made, which is a website housing numerous academic journals and research papers over a multitude of topics. This research was done on the 27th of December 2020. Querying the term Computational Thinking over the last three decades between 1990 and 2020 we get a total of **2805** documents. Analyzing the data we see brief occurrences in 1994 and 1995 on documents but since 2006 and ever increasing amount of research has been done on the topic showing it's growing importance.

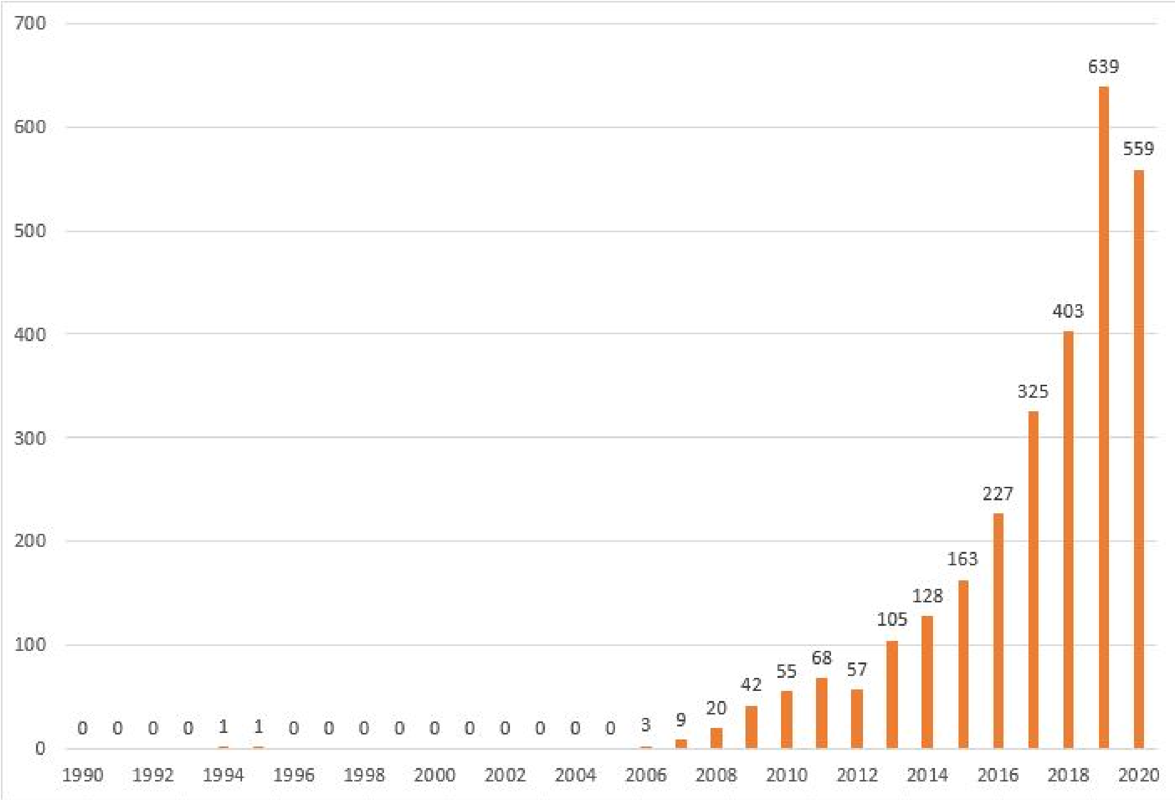


Figure 2: The number of documents found in Scopus made on Computational Thinking over the last 30 years.

Augmented Reality is a very interesting technology to work with and has been growing in recent years. With the advent of mobile technology such as smartphones and their applications as people look for ways to enhance their perception and interaction of the real world (Azuma, 1997). This technology being very attractive to consumers has various advantages and applications ranging from various sectors such as training and education, medicine, marketing and even tourism (Grzegorczyk et al., 2019).

Also using Scopus on the date of the 27th of December, interest was gauged on the topic of Augmented Reality as well. For the number of existing documents made on the topic of Augmented Reality in the last three decades (between 1990 and 2020) we get a total of **29706** documents. Analyzing the data we can see an ever increasing amount of research and work on the topic of Augmented Reality over the years showing the growing interest and popularity of the technology.

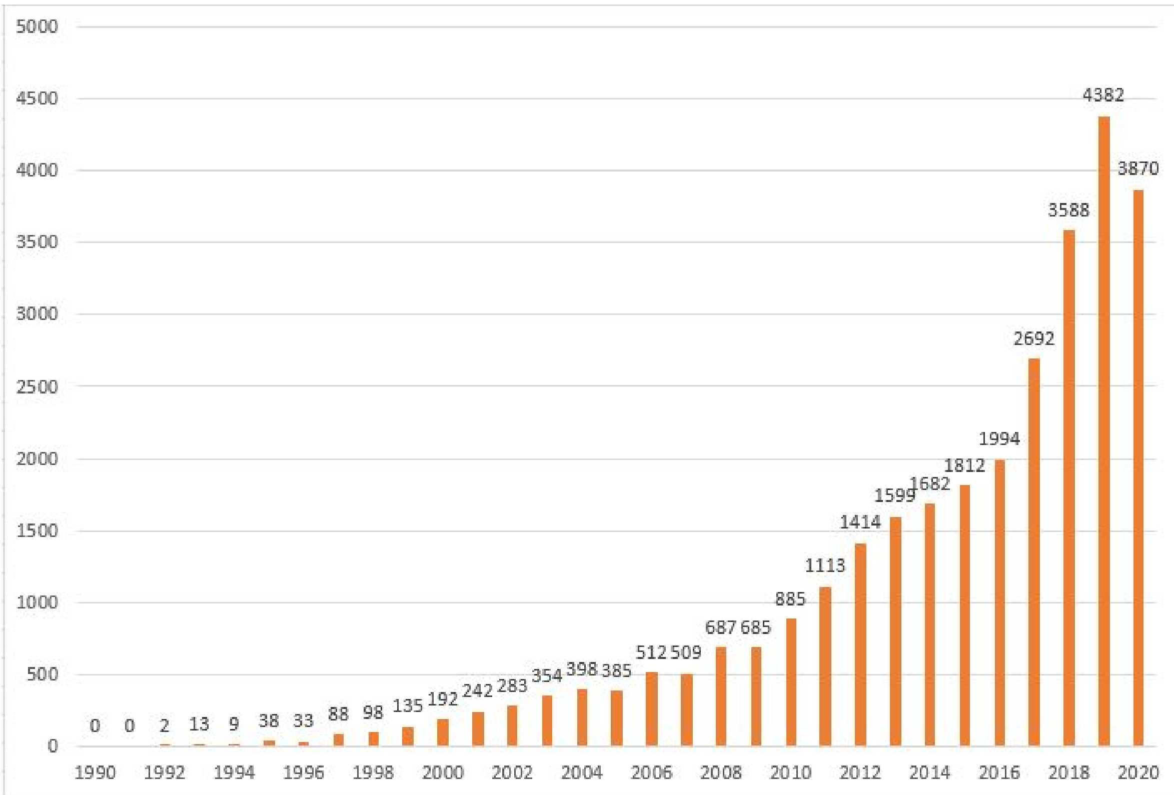


Figure 3: The number of documents found in Scopus made on Augmented Reality over the last 30 years.

After analysing and researching the topics of Computational Thinking and Augmented Reality the goal of the project is to create a tool to aid training these skills in young students. Taking the concepts of Computational Thinking and incorporating them into interactions with Augmented Reality is what is believed to be key to a fun and engaging experience in younger audiences. As a study by Bursali and Yilmaz (2019) on the effects of Augmented Reality in secondary student's has shown, students gained higher learning permanency and higher reading comprehension when using Augmented Reality technology in their school activities, while also being highly interested in using Augmented Reality in more courses, showing the attractiveness of the technology.

1.2 OBJECTIVES

This work is split into a theoretical section about Augmented Reality and Computational Thinking and a practical section involving the proposal of the artefact to help solve the questions about training Computational Thinking.

The objectives of this work are:

- Study Augmented Reality and the impact of building games to train Computational Thinking;
- Develop a game based in Augmented Reality and conduct tests to study the benefits of Computational Thinking.

The final result will be a program that can provide a game that will help develop the four characteristics of Computational Thinking such as Pattern Recognition, Decomposition, Algorithms and Abstraction implemented as game play aspects and hopefully train the subjects in such techniques.

1.3 RESEARCH HYPOTHESIS

It is possible to improve Computational Thinking skills via playing an Augmented Reality puzzle Video Game.

1.4 METHODOLOGY

The methodology used in this work will be based on Design Science Research (Pello, 2019), which is a method of research focused on solving a problem of a new approach of reality instead of explaining an existing reality or problem. The results and evaluation of an artifact provides information and feedback for a better understanding of a reality to improve on it and maybe solve it (Hevner et al., 2004). Therefore the thesis will have the following guidelines as per Hevner et al. (2004) and as per example in Figure 4:

1. **Design as an artefact:** Create a viable artefact to address the problem in question.
2. **Problem relevance:** The objective is to create technology based solutions to important problems.
3. **Design evaluation:** The efficiency, quality and purpose of the artefact must be very well demonstrated via well-executed evaluation methods.
4. **Research contributions:** Benefits of the research must be made clear in the areas of design artifact, design foundations and/or design methodologies.
5. **Research rigor:** DSR applies rigorous methods to in the creation and evaluation of the artefact.
6. **Design as a search process:** Various means must be met to reach the desired results of the artefact.

7. **Communication of research:** The artefact, research and results must be presented to to a technology and management oriented audiences for information spread.

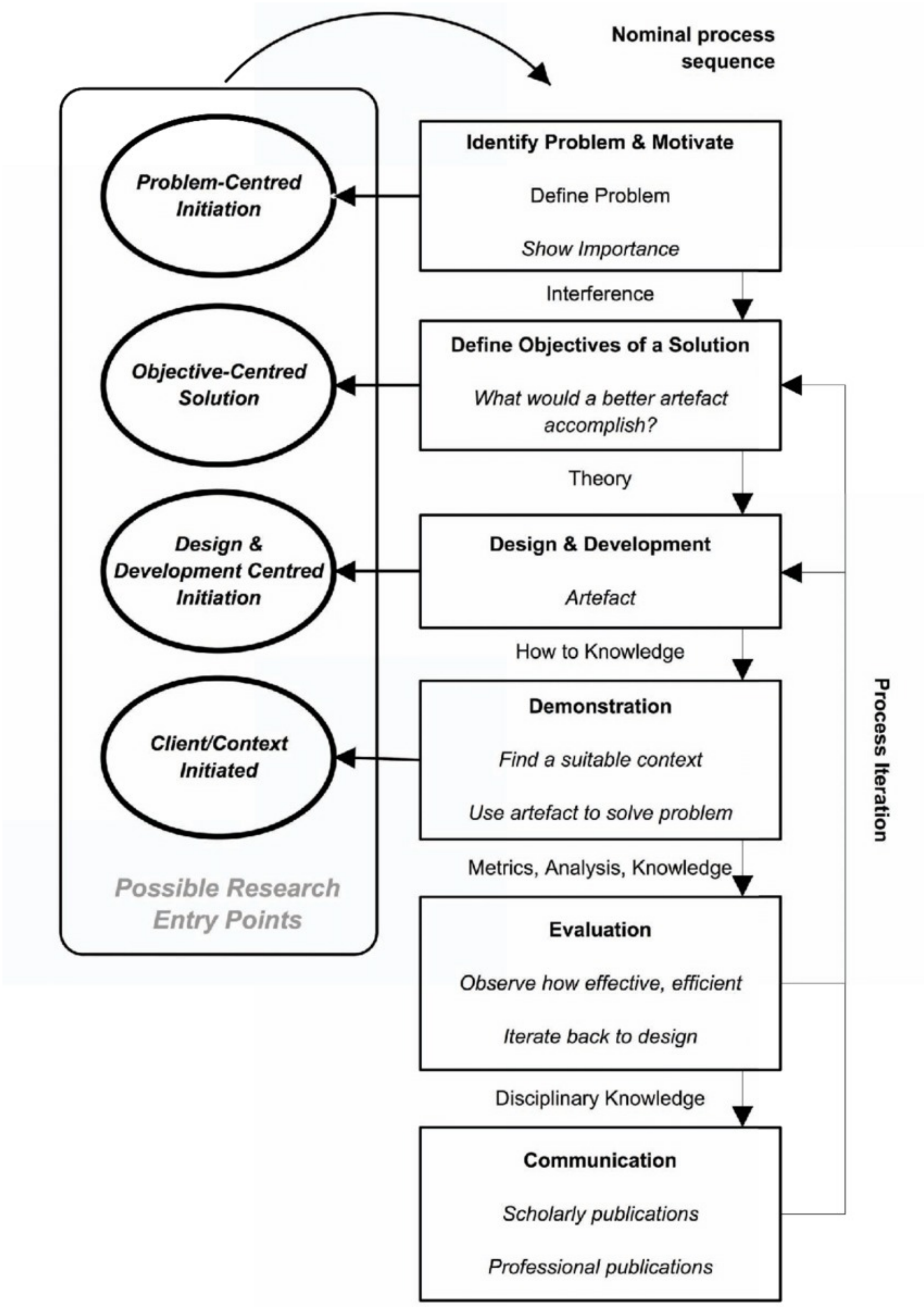


Figure 4: The process and phases of the Design Science Research methodology

For this work to be successful and have a steady and organized development time, it has been split into three major parts taking forth the 7 guidelines proposed for working with Design Science Research. The first part will be compromised of a study on Computational

Thinking and Augmented Reality as tools and their gathered bibliography to analyse and connect the studies together. The second part will be designing the game based on the four principles of Computational Thinking and the tools available for developing with Augmented Reality. Lastly based on the research and the development of the Video Game tests will be made to try and answers the questions of training Computational Thinking via playing the Video Game.

1.5 DOCUMENT ORGANIZATION

This document is organized in the following chapters: At Chapter 2 is an overview on the definition of Computational Thinking and their applications, importance and relation with Video Games. An overview on Augmented Reality, the existing types of Augmented Reality and their uses and popularity over time in Chapter 3. The proposed artifact and its concept, giving examples and connections with Computational Thinking and Augmented Reality in Chapter 4. In Chapter 5 the whole process of creation of the artefact is detailed. Chapter 6 gives a demo and overview of the final artefact developed. In Chapter 7 the study of the hypothesis is discussed and results analyzed. Lastly, Chapter 8 is the conclusion of the document where a review on the state of the project and the future work is given.

2

COMPUTATIONAL THINKING

Computational Thinking's definitions vary, but the most broad and explained definition is that Computational Thinking "is the thought processes involved in formulating problems and their solutions, so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." (Nickerson et al., 2015).

In simpler terms it is the process of thinking "like a computer" where we formulate solutions on various kinds of problems. In Computational Thinking there exist four main characteristics that define it as per Figure 5. These are the four thought processes associated with formulating problems and their solutions:

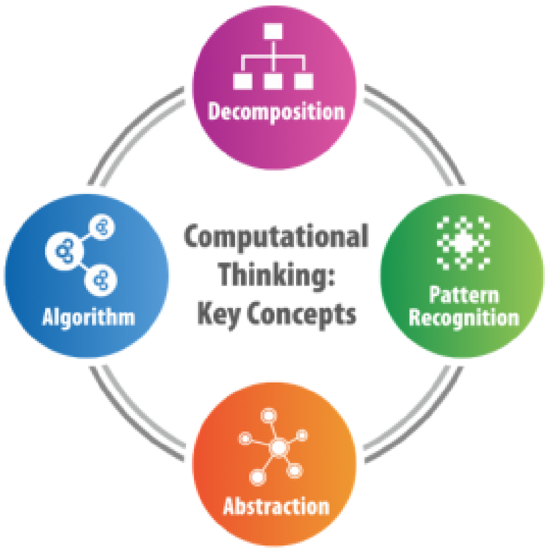


Figure 5: The four concepts of Computational Thinking. (Retrieved from cspathshala.org)

Pattern Recognition is the procedure of recognizing common characteristics or traits in problems and solutions. This leads to finding possible and efficient solutions. **Decomposition** is the process of splitting a bigger problem into smaller ones, which can then be used to solve each individually to solve the bigger problem together. **Algorithms** are sets of instructions calculated for a determined solution of a problem. It is based on executing a set of instructions to lead to an outcome. Lastly **Abstraction** is the ability of separating elements in a problem as a way of organizing the data structures to better understand the complexity and model of the problem.

2.1 Importance and Benefits

The importance of developing Computational Thinking skills as Jeannete M. Wing puts it in her 2014 article Wing (2014), Computational Thinking has found its way into a lot of research agenda of all science and engineering disciplines. Jeannete also goes on to say that Computational Thinking has come into play in a lot of other important professions and areas such as medicine, economics, finance, law, arts and even digital humanities and digital journalism.

On a recent meta analysis of various studies on Computational Thinking and their success in academic studies (Lei et al., 2020), results showed a strong evidence linked with academic achievement in students with better Computational Thinking skills, the relationship being stronger in cultures between west and eastern groups, a strong relationship between elementary students than in later grades and a strong relationship when grading assignments on students. Factors indicating success with Computational Thinking seem to be linked with culture, grades, achievement indicators and gender, but most importantly that learning these skills in fact yields better results academically and at a younger age even better.

Much of today's data mining techniques are also heavily based on the Computational Thinking process and it's four principles. It is for that reason and the forever growing abundance of data to be processed in today's age (Borkovich and Noah, 2013), and the accessibility of computers and internet connections, that developing these skills is important for future graduates and students alike to tackle activities and future jobs and professions that will present them with problems that can be solved via aid of these Computational Thinking skills.

2.2 Computational Thinking and Video Games

A Video Game is an electronic game that involves interaction between an interface and an input device. Video Games offer very fun and interactive experiences to users and over the years have seen more and more popularity in its use as tools of learning to captivate younger generations in the ever more digital world we live in. Learning and Video Games is a topic James Paul Gee has tackled immensely in research and books.

What captivates people to play and learn long and complex Video Games? As James Paul Gee explains in Gee (2005), it's the design of how these games are made that influence the interest and appeal of people to play them, which he denotes as **good principles of learning** that are built into the design of these games.

He has laid 13 principles of learning that are divided into 3 categories: **Empowered Learners**, **Problem Solving** and **Understanding**. Aspects like co-design, customization, identity, manipulation and distributed knowledge, well ordered problems, pleasantly frustrating gameplay and information on demand are some of the aspects that make Video Games very enticing and appealing to play and learn, so taking those aspects and using Video Games to teach or develop some results are a strong possibility.

In the research of results by using Video Games as tools, one study by Oei and Patterson (2014) on if playing puzzle Video Games could improve executive functions, results showed that playing a complex puzzle game that demands strategy, problem interpretation and analysis improved several aspects of executive function. It is assumed that due to the variations of difficulty, strategies and problem solving methods may have led to a greater learning and generalization of the tasks used to test changes in the cognitive and control networks in the brain, responsible for high problem solving and executive functioning.

Another study, this time more directed at fostering Computational Thinking skills in middle school students by Zhao and Shute (2019), it was discovered that by playing Penguin Go for less than two hours a day, improved students Computational Thinking skills significantly, yet results on the impact of learning were inconclusive, but it might support the study on academic success and Computational Thinking by Lei et al. (2020) as being a factor of culture and the differences in the grading system.

Constructionist Video Games are a genre of games where the objective is to challenge the player in their skills to build a system by understanding the world around it, Constructionism being the learning theory of the same definition, but applied in Video Games. Games like Minecraft, Eco, Sim City and Dwarf Fortress are games tied to a simulating system where the gameplay is tied to the player's ability to construct, interpret and analyze common problems in their environments.

Dwarf Fortress in Figure 6 is heavily notorious for being a hard to learn game with its complex systems and UI (User Interface), in which the objective of the game is the simple premise of managing an outpost of dwarves and their lives, but as the player progresses and interacts with the game, it soon realizes the task at hand is much more complex and requires deep thought of management and strategy to keep the outpost going for longer, leading to the player to develop cognitive skills to deal with the various problems presented in the game.

Computational Thinking in Constructionist Video Games was the topic of a study Weintrop et al. (2016) where it was analyzed if playing this genre of games stimulated Computational Thinking skills in their players by guiding three principles they defined: the construction of personally meaningful computational artifacts, , the centrality of powerful ideas and the opportunity for learner-directed exploration, all principles with strong ties to the 13 principles coined by James Paul Gee in his studies in Gee (2005).

The results showed that overall in all three Video Games played, most if not all students were able to show signs of acquiring skills tied to Computational Thinking such as abstraction of data, pattern recognition of properties of the games, algorithms to solve the levels and decomposition of problems to solve the bigger scope. With these studies and research it is to be believed a Video Game can indeed be used to train Computational Thinking skills in people.

Figure 6: Dwarf Fortress is a complex fantasy simulation game, in which the objective is to manage a settlement of dwarves. (Retrieved from morganpawprint.com)

In the following Chapter an analysis and discussion about the technology intended to be joined with these concepts of Computational Thinking will be given. This technology is Augmented Reality.

3

AUGMENTED REALITY

As the plan is to use Augmented Reality in conjecture with Computational Thinking, we will go over a few topics pertaining to Augmented Reality in this Chapter as to contextualize the project and work more. It is important to note that while Augmented Reality originated from Virtual Reality, both today are very different concepts. As the purpose of Augmented Reality is to super impose the world around us with more information, Virtual Reality simply simulates reality in it's own nested environment. Azuma in his survey Azuma (1997) defined Augmented Reality by three characteristics:

- Combines real and virtual
- Interactive in real time
- Registered in 3-D

For a brief history of Augmented Reality and key points over the years, the first head mounted Augmented Reality display was created in 1968 by Ivan Sutherland, called "The Sword of Damocles", that super imposed shapes and lines over the device's lens. Following in 1990 the term Augmented Reality was believed to first be coined by Tom Caudell, who came up with the term for an Augmented Reality system to be used that would help make more efficient work out of switching between workers and the different sets of instructions for plane's wirings (Candy, 2013).

For years Augmented Reality would find it's place only in laboratories and mostly scientists unlike the average consumer of today. In 2000 that changed when Hirokazu Kato developed an open-source software library named ARToolKit aiding in developing Augmented Reality software programs. This began a transition point for the accessibility of the development of Augmented Reality and the exposure to the general audience and population. In more recent times with 2009 we've had the transition of Augmented Reality technology to the Web with also the aid of Hirokazu Kato's ARToolKit and recently in 2017 ARKit and AR Core are launched by Apple and Google respectively to streamline and help the development of Augmented Reality software in their systems.

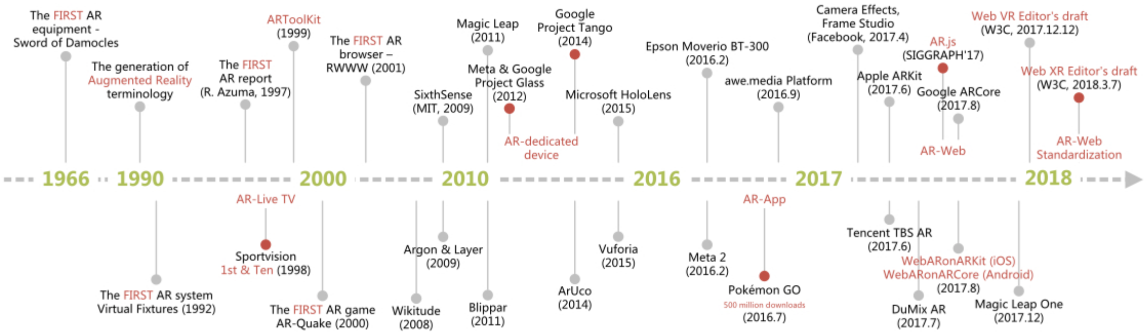


Figure 7: The evolution of Augmented Reality over the years. (Retrieved from Qiao et al. (2019))

3.1 TYPES OF AUGMENTED REALITY

The types of Augmented Reality are classified based on the display used to see the virtual elements (Azuma et al., 2001), therefore they can be classified as: Head-Worn Displays (HWD), Handheld and Projection.

Head-Worn Displays are devices directly attached to the users head, providing the eye directly with feed of the augmentation and virtual objects. They come in two types: optical see through and video see through. The difference, seen in Figure 8, are that the display types directly send feed to a transparent display giving an Augmented Reality overlay, while the video uses a camera as background for the Augmented Reality overlay, which is then transmitted to an opaque display.

Figure 8: Difference between Optical (left) and Video (right) types of Head-Worn Displays. (Retrieved from Azuma et al. (2001))

Handheld, which are the most common of today, give the consumer easy access to Augmented Reality via the camera in their smartphone or tablet, example in Figure 9. This type of Augmented Reality has the advantage of being portable and easy to access with decent computing power tied to the device as compared to the more lightweight Head-Worn Display types.

Figure 9: Example of a handheld type Augmented Reality. (Retrieved from blog.koerich.com.br)

Projection, is the type that is directly projected into an environment without the need of a special lens or head wear to see the information, these usually require a projector to send the imaging to a stable surface, example in Figure 10. There is also a different approach to this type, combining also Head-Worn Display, but instead of projecting the Augmented Reality to a lens to the user to see, it projects the Augmented Reality to everyone around.

Figure 10: Example of a projection type Augmented Reality projecting a piece of shoe wear. (Retrieved from youtube.com by augmented.org

3.2 POPULARITY OVER TIME

Since it's inception Augmented Reality showed promise in their use, but has not seen much popularity or at least attention to it's existence to the every day person. Even by the very early uses back in 1997 by Azuma's paper Azuma (1997), over the years Augmented Reality has seen mild use or a small but steady increase in it's popularity and uses.

Maybe the most famous case of a boom in popularity on Augmented Reality mobile applications has been Pokémon Go (Molina, 2016), a Video Game utilizing Augmented Reality in its core feature where the objective of the game is to go outside into the real world and engage in capturing digital monsters by interacting with them through the lens of the smartphone. This game threw the notion of Augmented Reality into the wider general audience and spiked an interest and boom of other Augmented Reality type Video Games. In the article by Romanov (2020), it's debated how Augmented Reality is very present in much of our every day lives but people do not notice it or give much attention to it's existence.

With the evolution of technology and modern smartphones and readily available web applications on the go Augmented Reality has a more present presence in the form of mobile apps, as opposed to the old independent equipment or screens needed to view it, yet, Augmented Reality has not seen much use in these devices even though the plethora of existing applications that use it. Perhaps it is the need to having to install the application, maybe not seeing any real potential use in it or the fact of the poor marketing or use of Augmented Reality and it's incorporation in the application. WebAR (Kovtun, 2020) and SocialAR might be the bet to push forward people experimenting more and using Augmented Reality more frequently as with these new frameworks accessing Augmented

Reality is faster and simpler as there is no need for the installation of an app, only needed an internet connection and a browser to see it in action. In future prospects according to Statista (Tankovska, 2020), the global market for immersive technology will be of around 6.3 billion dollars with Augmented Reality accounting for 3.7 billion of that value which is more than half, showing indeed there is a market and there is serious investment and growth over the market for these technologies.

As Augmented Reality has had an increase in popularity and the use of this technology becomes more prominent in our lives, the question of its use in helping with education also comes into play. We know of its use in training and educating medical professionals but other areas or varying degrees and levels of education can see the potential and use of the technology. On an article by Matt Bower, Cathie Howe, Nerida McCredie, Austin Robinson & David Grover in Bower et al. (2014), it's discussed how Augmented Reality is used in learning and pedagogical approaches and exercises with lower thinking order as for example remembering, understanding and applying while high thinking orders of analysing, evaluating and creating have not been so thoroughly explored. Such activities include constructive learning, situated learning, games based learning and enquiry-based learning.

3.3 AUGMENTED REALITY GAMES

To better understand how our Video Game will function and use Augmented Reality as the focus point in the gameplay, research on popular and existing Augmented Reality games was made and a comparison between them on its use of the technology was done. Compiling the research done the following list of Video Games of note to analyse and evaluate the use of Augmented Reality and what categorizes as good and bad use of Augmented Reality. Firstly, to define what constitutes "bad" use of Augmented Reality, it is the redundancy or lack of interaction in the gameplay of a Video Game in its Augmented Reality, meaning the technology adds no difference or benefits if the game was to not use the technology in its core, while "good" use of Augmented Reality is what utilizes it in the gameplay and is shaped by its use and interaction.

In the research done some Video Games with bad examples of use of Augmented Reality were found, but we will focus on two examples: Eggs Inc and Angry Birds: Isle of Pigs. Eggs Inc, in this video of gameplay, is a farming simulator game focused on raising chicken to produce eggs. It has an Augmented Reality mode that super imposes the player's farm over a surface and allows a free panoramic view of the farm as showed in Figure 11, but offers no use of the technology other than for visualization, as it brings no interaction or effect to the regular gameplay. The second game is Angry Birds: Isle of Pigs, a physics based game where the objective is to hit green round pigs with a sling shot propelled bird and to topple the buildings they are in. The game also super imposes itself on a surface but again brings no interaction with the use of Augmented Reality in the gameplay.

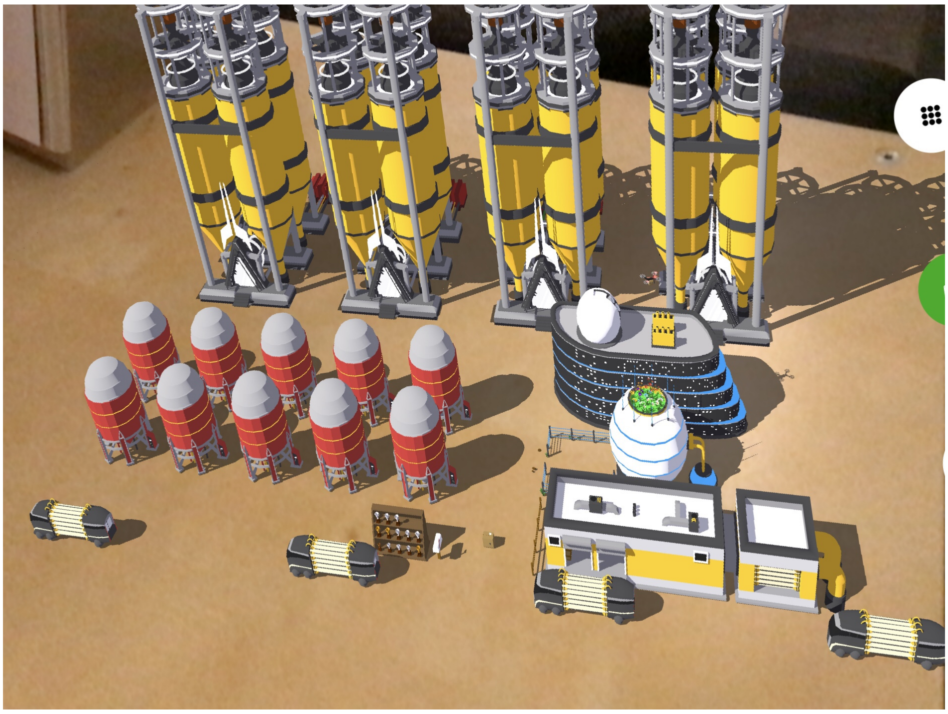


Figure 11: Gameplay of Eggs Inc, with the players farm super imposed onto the real world. (Retrieved from egg-inc.fandom.com)

On the list of Video Games with good use of Augmented Reality, the following are of note to discuss: Airport, ARise and Yume II Alice's Adventures. Airport, video of the game in action, is a simple airport management game that utilizes a blank airport poster that the player can paint the planes and objects, which then will be reflected through the tablet in Augmented Reality, becoming animated and usable in the Video Game to manage your airport. This is good use of Augmented Reality indeed the view of the real world has been augmented to the player in interactions. ARise is an Augmented Reality puzzle game where the player imposes a 3d map on a surface and the objective is to guide the player avatar up the level by using perspective, Figure 12, to connect the puzzles and make the character move forward until completion. A good example of use of Augmented Reality in gameplay using the interaction of perspective in it's core gameplay. The last is Yume II Alice's Adventures, a more high production Augmented Reality Puzzle Game with a more focus on story as well, the player also utilizes perspective and various spells in the map to advance the character forward in the level.

Based on the research and our analysis of what constitutes good use of Augmented Reality in a Video Game, the genre decided upon was to be a 3d puzzle, that will use perspective and the use of markers in it's core gameplay to keep the engagement and interaction with the technology, which will be further explored in the following Chapter.

Figure 12: Gameplay of ARise, in which the user is using perspective to connect the puzzles. (Retrieved from arcritic.com)

After these topics of Computational Thinking and Augmented Reality the following Chapter will try to combine these two topics to try and reach the objectives of this study, which is to develop Computational Thinking skills via playing a Video Game. To help in this Video Game, Augmented Reality will play a key role, but firstly an introduction to the idea and proposal in the following Chapter.

4

SHREWS: PROPOSAL

After a study of Computational Thinking, Augmented Reality and Video Games centered around Augmented Reality, this Chapter will propose the artefact intended to be developed based on the gathered information, data and research to develop a Video Game that will aim to train and focus on the four principles of Computational Thinking and Augmented Reality centered functionalities and engagement.

4.1 ARTEFACT DESCRIPTION

As with the methodology of Design Science Research in this project an artefact will be developed to solve the problem, which is to train Computational Thinking via playing a Video Game with the support of Augmented Reality.

To develop a proposal of a Video Game to train Computational Thinking, we have to look at the four main characteristics that compose it such as decomposition, abstraction, algorithms and pattern recognition. In Video Games that tackle these four characteristics there's a genre that stands out among them and it is puzzle games, which make the player think and understand various concepts and problems to solve. A very popular puzzle Video Game that tackles these concepts is Lemmings, Figure 13, a 2d puzzle-platformer Video Game developed by DMA Design and published by Psygnosis for the Amiga personal computer, and later other platforms, in which the objective of the game is to guide a group of humanoid animal characters through various obstacles in a 2d view from a point A to a point B. In this game the player uses different skills to safely guide the lemmings to the exit door by altering the map or changing the behaviour of the characters to avoid danger and give safe instructions to the end point.

In a research on scalable game design (SGD) used to give young students Computational Thinking acquisitions by Nickerson et al. (2015), Lemmings was used as a tool to train problem solving skills on some success for young students better understanding of computational skills and decision making. Other concurrent projects and games that came out during the creation of this dissertation include RoboTIC (Schez-Sobrino et al., 2020) and ARQuest (Gardeli and Vosinakis, 2019), which both take the use of Augmented Reality and try and develop Computational Thinking skills in players. Both showed good motivation from the participants and positive conclusions on the training of CT.

Figure 13: Image of the gameplay of Lemmings, where the objective is to guide the humanoid characters to the exit. (Retrieved from stelioskanitsakis.medium.com)

On the premise these games are usually for all ages it was decided to create a puzzle game in a 3d environment akin to that of Lemmings, which in opposite is a 2d game. As Augmented Reality will be used as a tool that will utilize Augmented Reality centered interaction into the gameplay to bring a more attractive and immersive play with the puzzle solving. As the original Lemmings name is based off the rodent animal of the same name, it has also been decided to call this project **"Shrews"**, from the name of another small rodent animal as homage.

This puzzle Augmented Reality 3d game will have as a concept the problem solving of leading a group of characters with player inputted instructions over a 3d map environment.

Figure 14: Concept art of the title of the game.

As to better classify and organize the proposed game an ontology for game classification was used called **"Ontojogo"** by Teixeira et al. (2020) developed by Salete Teixeira, as a system to better classify the properties and attributes of games.

```
Shrews =iof => Game[name='Shrews', description='Shrews is a puzzle solving game
    where the player helps guide creatures over a map.', age_rating=7, storyline='
    fiction'];
Shrews =is_available => Digital Offline;
Shrews =has_mode => Casual;
Shrews =has_perspective => Third-Person;
Shrews =with_input => Movement-Based;
Shrews =with_input => Touch;
Shrews =has_player_number => Single Player;
Shrews =has_progression => Level-Based;
Shrews =belongs_to_platform => Mobile;
Shrews =belongs_to_genre => Puzzle;
Shrews =belongs_to_genre => Strategy;
```

4.2 SHREWS

The Shrews are the race of little humanoid animals that travel in packs, but they have the particular characteristic of having bad eyesight causing them to wander aimlessly without directions or indications. But among them a rare Shrew designated the **Builder** in Figure 15 might have very good eyesight and is their job to guide the pack of Shrews along their way.

Figure 15: The builder has wide open eyes.

The common **Adult** at Figure 16 Shrew has poor eyesight and wanders aimlessly without the builder indicating him where to go and opening the path forward, they might take a fall and be fine but too high and they might hurt themselves.

Figure 16: The adult is characterized by having closed eyes.

Lastly the **Elder** at Figure 17 Shrew is the less resilient and slowest of them all, he moves in the same manner as an adult Shrew but is less durable to falls or other hazards it might encounter along the way.

Figure 17: The elder is characterized by having closed eyes and a moustache.

4.3 MAP

The maps of the game are laid out in blocks with one being a starting point and the other the end point, it is the objective of the player to instruct the builder Shrew to build the objects and lead the way to the end point. The maps are a 3d environment, composed of many blocks in width and height, displaying them in various heights changes the landscape of the map, as in example at Figure 18.

Blocks have types of terrain, dirt blocks, rock blocks and water blocks. As for shapes blocks can be cubes, upwards triangles or downwards triangles and arcs. Blocks come in different properties and different types.

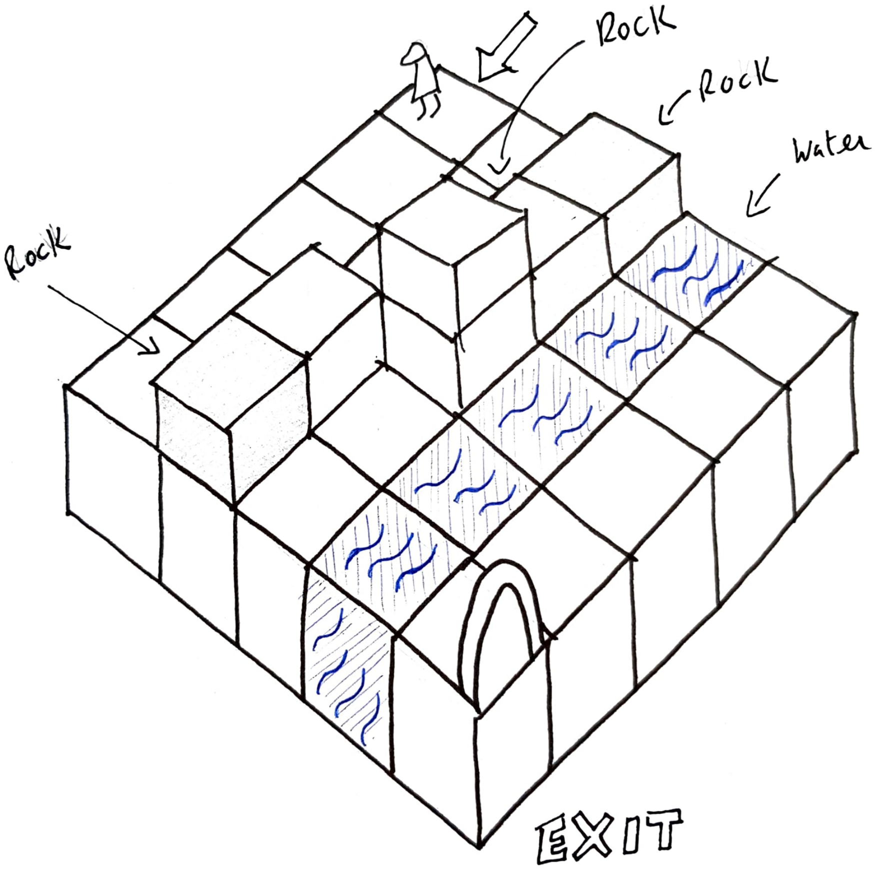


Figure 18: An example of a map of the game.

The three types of terrain and their properties are as follow:

- **Dirt**: Dirt blocks are normal blocks the builder Shrew can interact with by constructing and walking on them.
- **Rock**: Rock blocks are blocks that cannot be interacted with, meaning the builder Shrew cannot shape them but can walk over them.
- **Water**: Water blocks are blocks that cannot be walked over but can be constructed by making bridges.

The shapes and forms blocks can take by the builder Shrew are:

- **Squared**: Are the shapes all blocks take naturally. Dirt blocks in this shape can be interacted and shaped into others. Water blocks can be shaped into bridges. These squared blocks can be walked over except for water having the property of not being able to be walked over.

- **Triangle**: Triangles or Ramps and Slopes are dirt blocks that have been shaped by the builder Shrew and allow the other Shrews to climb and slide down into the next available squared block, be it dirt, rock or water.
- **Tunnel**: Tunnels are dirt blocks that have been shaped from dirt blocks and allow the Shrews to move through a block to the other side or another tunnel.
- **Bridge**: Bridges are blocks of water that have been shaped and allow the Shrews to walk over water.

4.4 TASKS

Tasks are the actions the builder Shrew performs to make a path for the Shrews. These are varied with some defining traits and characteristics that will cause the player to think how to build them to safely carry the Shrews from start to finish.

The player will instruct the builder to do these tasks by tapping on the blocks and selecting the according Augmented Reality card. The tasks are as following:

- **Move**: Instructs which blocks the Shrews can navigate, the player will tap the blocks and use the Augmented Reality card to signal these blocks are for the builder to instruct the Shrews to walk on. Example in Figure 19.
- **Ramp**: Creates next to the selected, or more, dirt blocks a ramp allowing the Shrews to go up or down in height to another block, depending on the direction they are walking from they will climb, or slide down the ramp. It is to note that the builder Shrew does not slide down the ramps, only the adults and elders, depending on how many ramps elders slide down the more hurt they get on landing. These are the only buildings the builder Shrew can construct without needing to shape a block. Example in Figure 19.
- **Bridge**: Constructs a bridge over blocks of water or more connecting blocks over a space, allowing the Shrews to walk over the water blocks. Example in Figure 19.
- **Tunnel**: Builds a tunnel through a block to move to the other side, this is only possible on dirt blocks and not rock blocks. Example in Figure 20.
- **Dig**: Destroys a selected block near the builder Shrew to open up space on the map for construction. Example in Figure 20.

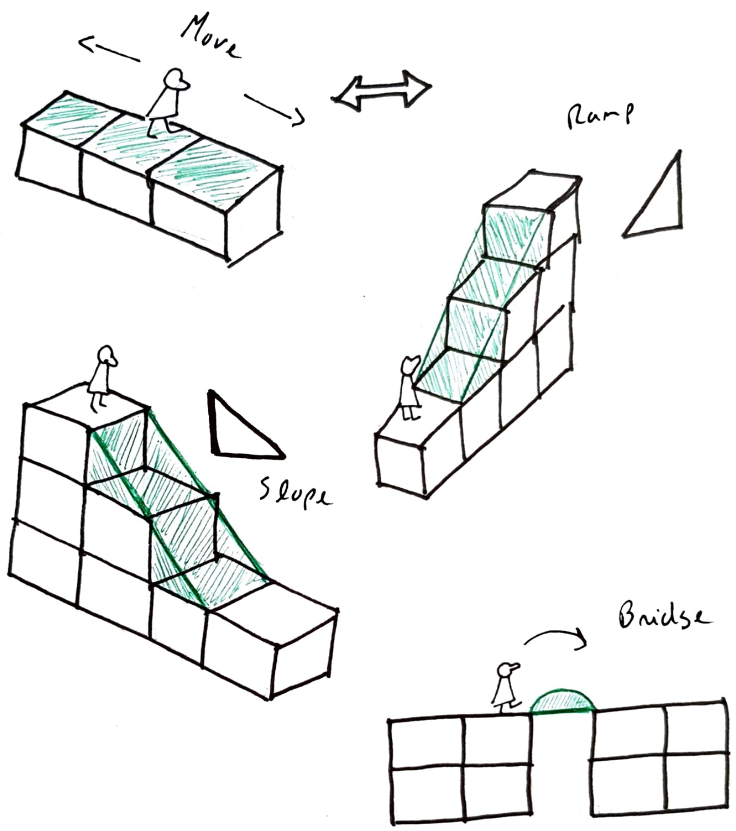


Figure 19: Skills of the builder Shrew.

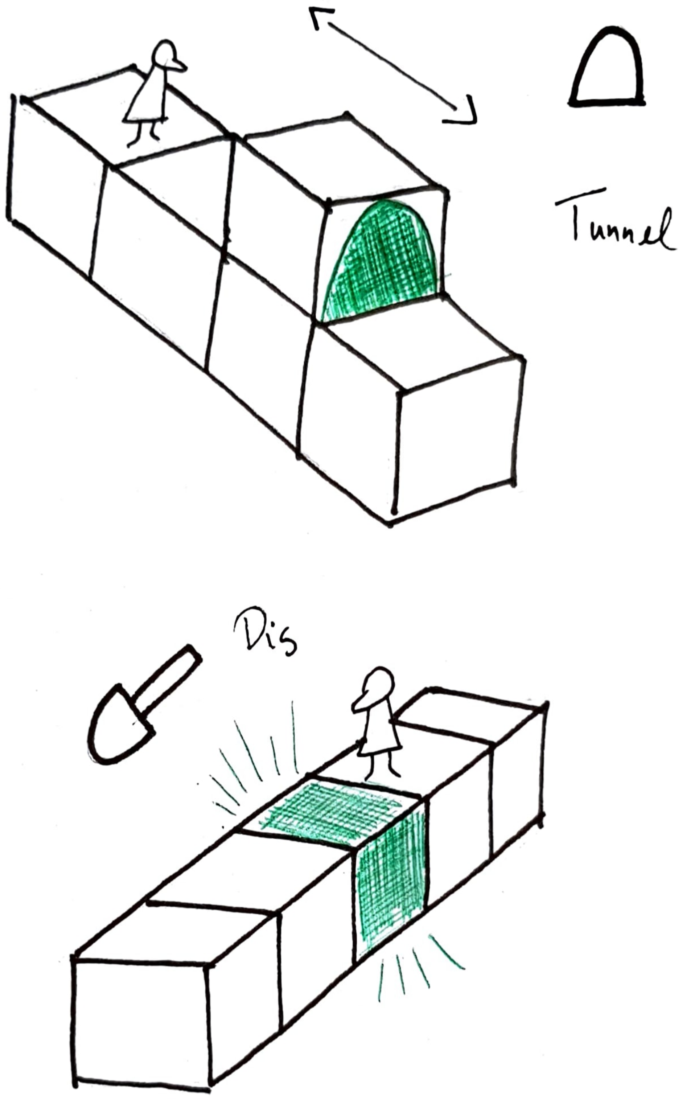


Figure 20: More skills of the builder Shrew.

4.5 GAMEPLAY AND EXAMPLES

The objective of the game is to transverse a group of Shrews over the map, from the start to the exit. For this the player will use Augmented Reality cards to instruct the builder where to build such pathways for the Shrews. The player will find a flat surface preferably to generate a map and will use perspective to analyze the map and terrain and figure out the best paths around the map to guide the Shrews.

The player will focus on the builder or the blocks and use cards to instruct which action the builder will do. For example tapping on a block and choosing a card will make the builder move to that spot to build the path forward. The remaining Shrews will walk back and forth nonstop on the pathway the builder Shrew has moved in. The game will have a scoring system based on various statistics throughout the session like time it took to solve the puzzle, how many tasks were used and how many Shrews made it safely to the end.

The Figure 21 shows how the player will use a device to see the Augmented Reality. The player will use a smartphone device or tablet as a scope into the game and see the Augmented Reality map and gameplay happening. Moving with the tablet around the map the player will be able to see the whole map from different perspectives. Using the Augmented Reality cards to read from the application will cause the various tasks of the builder Shrew to execute.

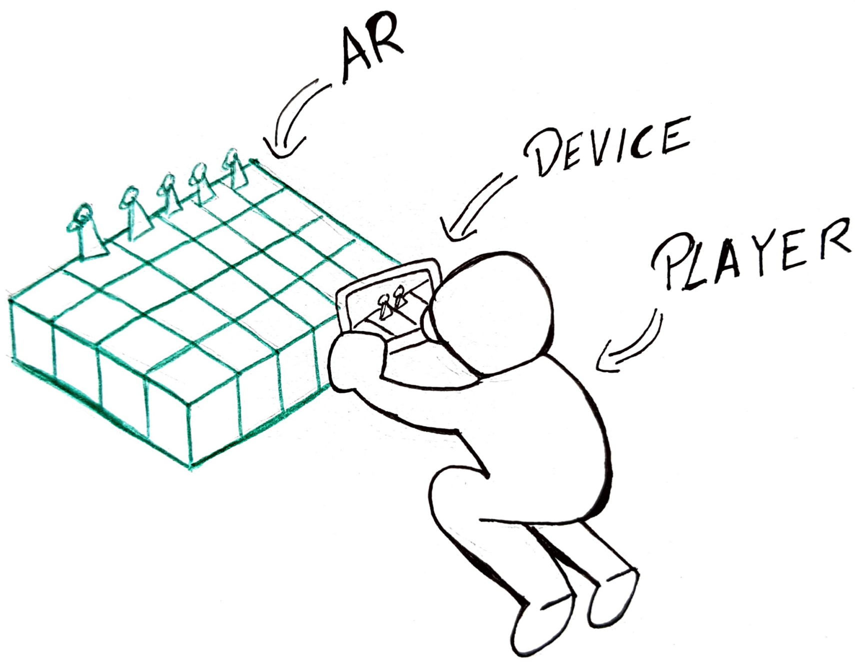


Figure 21: Example of user interaction with the game.

Taking into account the template map in Figure 18 we will now go over three examples of possible solutions for the map given, to show the various possible solutions for clearing a map and the many possibilities players might find while solving them. The template map holds various normal dirt blocks, three rock blocks and a river composed of five water blocks.

We will go over three scenarios to solve this puzzle, but there can be more solutions, it is up to the player to find them to promote each interpretation of the problem.

4.5.1 *Example 1*

In this example, Figure 22, it has been illustrated in colors the blocks and the tasks associated that the builder Shrew will perform. Illustrated in green are the blocks indicated to walk over, followed by a block in brown that the builder will construct a tunnel, followed by red, a bridge to cross the water and the remaining three tiles in green is to walk over to the exit. This is a more simple approach to this puzzle as it is straight forward, without going up in height and down to cross the the higher blocks.

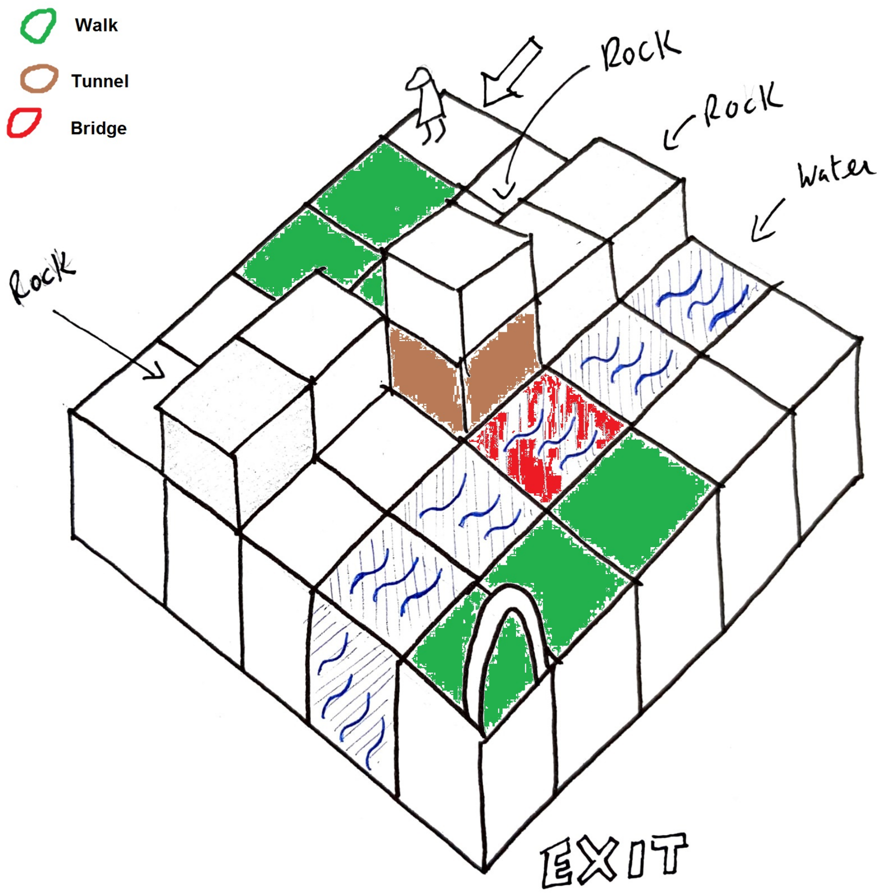


Figure 22: Example of a first solution to the puzzle.

4.5.2 *Example 2*

In this example, Figure 23, we have another straightforward example but with other tasks used. We have the green being where the Shrews will move, with orange the block where a ramp will be built to climb up to the next block, at the top and right next in blue another ramp will be constructed to go down, as the builder Shrew does not slide down the ramp if quick to build as the other Shrews walk slower, it can build a bridge in the red over the water block and move to the exit safely. This example is also straightforward but uses more tasks to solve the puzzle if the player wishes so.

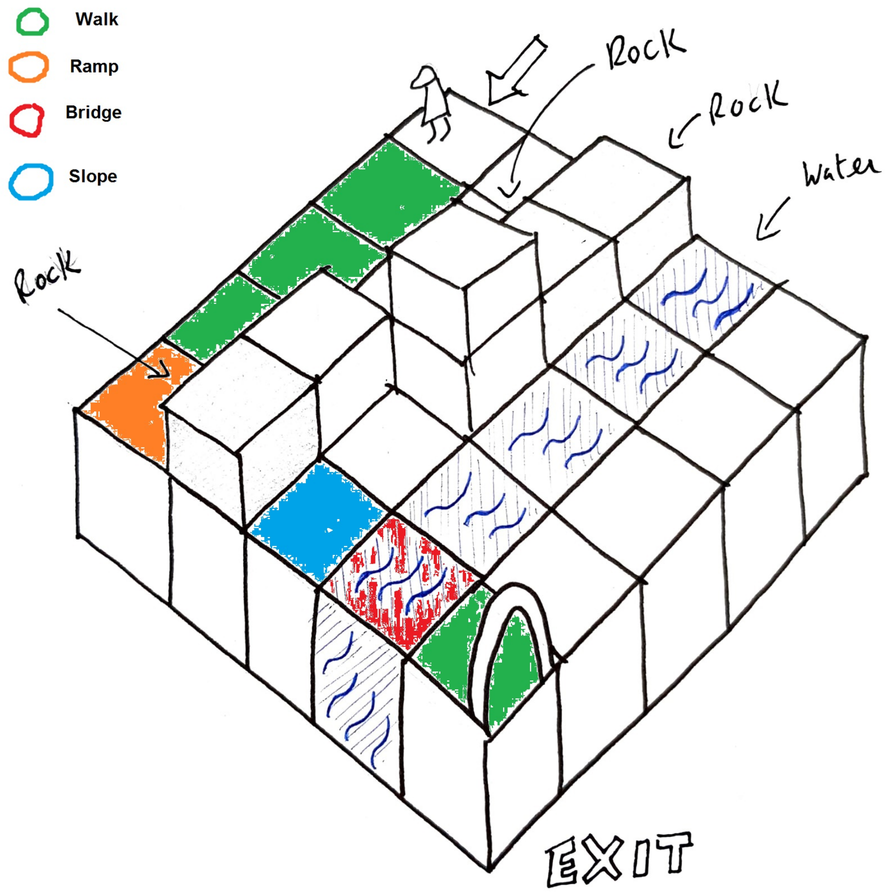


Figure 23: Example of a second solution to the puzzle.

4.5.3 *Example 3*

In this example, Figure 24, we have a more complex example with a lot of tasks used to solve the puzzle. First we have in orange a ramp being built to climb into the rock block, green to signal to walk over this block, and right after another ramp to go up into the following block up, in purple the builder will dig destroying the block to make it easier to construct below after. In the process of this in blue a ramp to go down is built. As the purple block was dug and destroyed the ramp in orange will remain and the Shrews will climb up and fall down one block of height with no danger and slide down the blue mark safely as well. In green the Shrews move and in red a bridge is built over the water block with finally a green mark to walk, leading to the exit.

This is a more complex example as it deals with more properties and variables of the game. Take for example the purple dirt block, we made a ramp to climb and dig, but it could have simply been made into a tunnel with the blue mark being a ramp to climb down. If the player would have made the mistake of digging two dirt blocks with the purple and the one below, the Shrews would have fallen down two blocks in height and the elder Shrews would have been hurt from the fall, causing the player to lose points. It is these properties and patterns players will learn to notice and play around to make the best out of the puzzle.

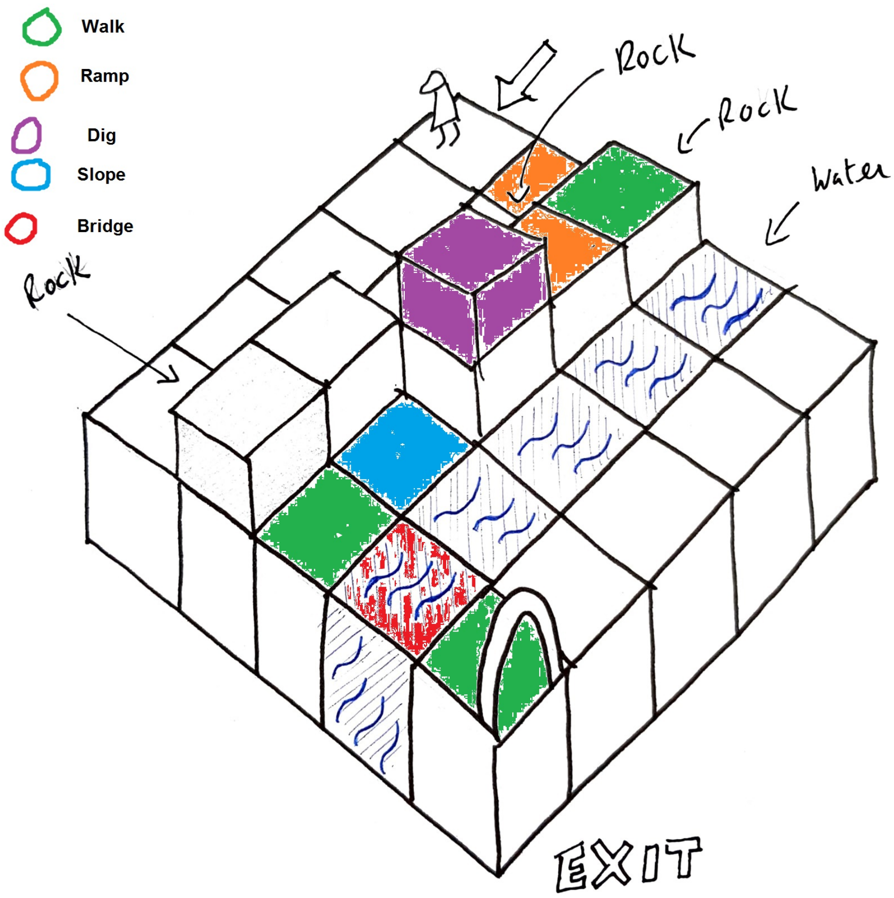


Figure 24: Example of a third solution to the puzzle.

4.6 Shrews and Computational Thinking approach

As the objective is to train Computational Thinking skills in the players, the gameplay will focus on the four thought processes, this way the players will experience the problem solving methods used in Computational Thinking and train them in game to solve the puzzles. For example:

- **Pattern Recognition**: By utilizing skills and how they interact with the blocks to give pathways players will start recognizing patterns. Identification of the functions that the buttons perform and how they interact with the blocks, with each button only working in certain blocks and with different rules will develop pattern recognition in the players;

- **Decomposition**: The main character must overcome one obstacle at a time, by decomposing the big puzzle into pieces, each block presents an obstacle and problem to solve with the tools available, giving the player the aspect of decomposing the large puzzle in small problems;

- **Algorithm**: The objective of the game is to trace an algorithm that will lead the Shrews from point A to point B. The sequence of steps to solve the puzzle, with so many different ways of solving it, each player has their own algorithm to create and take the Shrews to the exit;

- **Abstraction**: The difficulty of the puzzles are measured by the number of existing blocks and overall dimension of the map. It's important to abstract how flat surfaces of blocks can be paths as opposed to groups of blocks in different heights, as well as the different types of terrain. Visualize the challenge map and work out the best exit route for the character by analyzing the tools, movement and blocks will give abstraction to the player;

4.7 SYSTEM ARCHITECTURE

The Figure 25 represents the system as a whole, by components and interaction. The components that make up this system are the **device**, such as a smartphone or tablet, that server to give the user interface and in which the player will use with the software that will be developed and used to act as the **digital combiner** to manage the digital information in the real world. This software will be tasked with the execution of the Video Game and creation of the Augmented Reality environment through the **camera** and **tracker**, so that the player will use as a lens through the **user interface** to see the information.

The device will use it's **camera** to track the surface to project the Augmented Reality and also use it's **camera** to identify the various Augmented Reality **markers** to execute the Shrew's tasks.

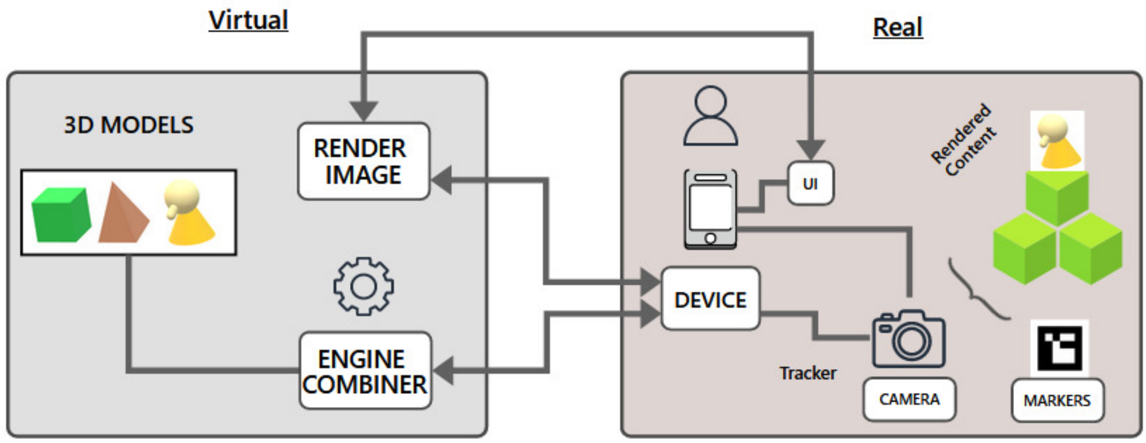


Figure 25: The system architecture of Shrews with the various components.

After the proposal of combining Computational Thinking and Augmented Reality to create this project named "Shrews" and it's explanation, brief conclusion of the document will be given, up to this point in the following Chapter.

5

IMPLEMENTATION

The objective of this project is to develop an Augmented Reality Video Game capable of training Computational Thinking skills. As discussed in the proposal Chapter, the foundation of the rules and game were set and defined, therefore in this Chapter, the technologies that helped put together this project will be explained, as an explanation of what they are, what they contribute to the project and how they fit together.

5.1 TECHNOLOGY APPROACH

To join the various technologies and their interactions over the large system, in Figure 26 it is detailed the large architecture of the technologies used in this project. The main technology and engine of the game is Unity 3-D® responsible for the game program. To aid in the Augmented Reality technology the software development kit Vuforia ® was used in the Unity engine to generate the AR. Lastly external assets like 3d models, images and audio were created, imported and used in Unity.

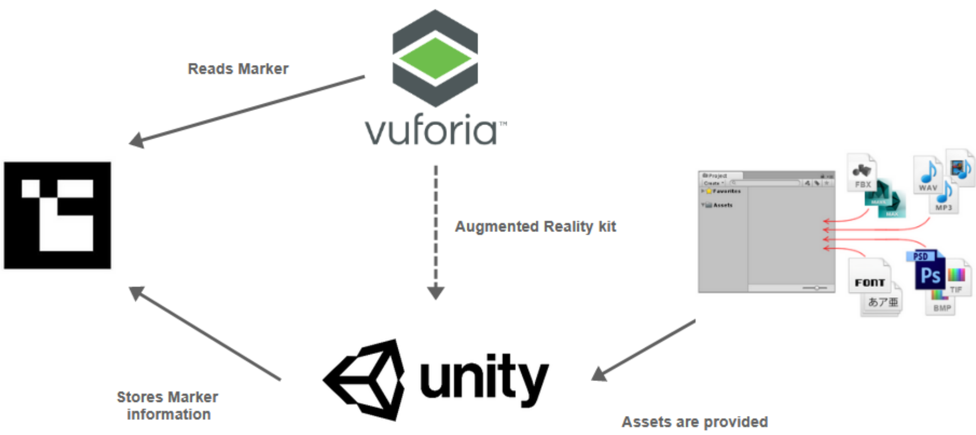


Figure 26: Architecture of the implementation and technologies.

5.2 UNITY

Unity 3-D® is a cross-platform game engine developed by Unity Technologies that was released in 2015. It is a widely known and popular game engine due to its beginner friendly and indie development approach, not being restricted to Video Game development but also for simulations and other areas such as film animations, architecture and construction. Unity boasts various advantages as a game engine such as:

- Various platform support, meaning games can be ported easily to many different systems
- Good graphics and visuals management with good terms of performance
- User friendly interface for developing
- Low cost compared to competitors, with even a free version available
- A proper IDE for developing
- Good support and documentation for aiding in work

Figure 27: Unity logo.

Is it to note the Unity version used was v.2018.4 for LTS (Long Term Support) as it is a most stable version with stable support usage of Vuforia's Augmented Reality development kit. Unity's basis allows for users to create game projects and manage various objects within the Unity framework within an interface like in Figure 28. The most common and important being Scenes and Game Objects from which fall into Assets, Effects, Cameras, UI and Audio. Setting a new Scene a user can define which objects are present, what they interact with or how via the properties panel, by giving various characteristics such as color, shapes, rigid bodies or physics effects and even custom scripts, handled by C code that make use of the

Unity library and imports to alter the object with many ways. An example of the properties for a game object at Figure 29 shows a Shrew model and various properties attached to the object.

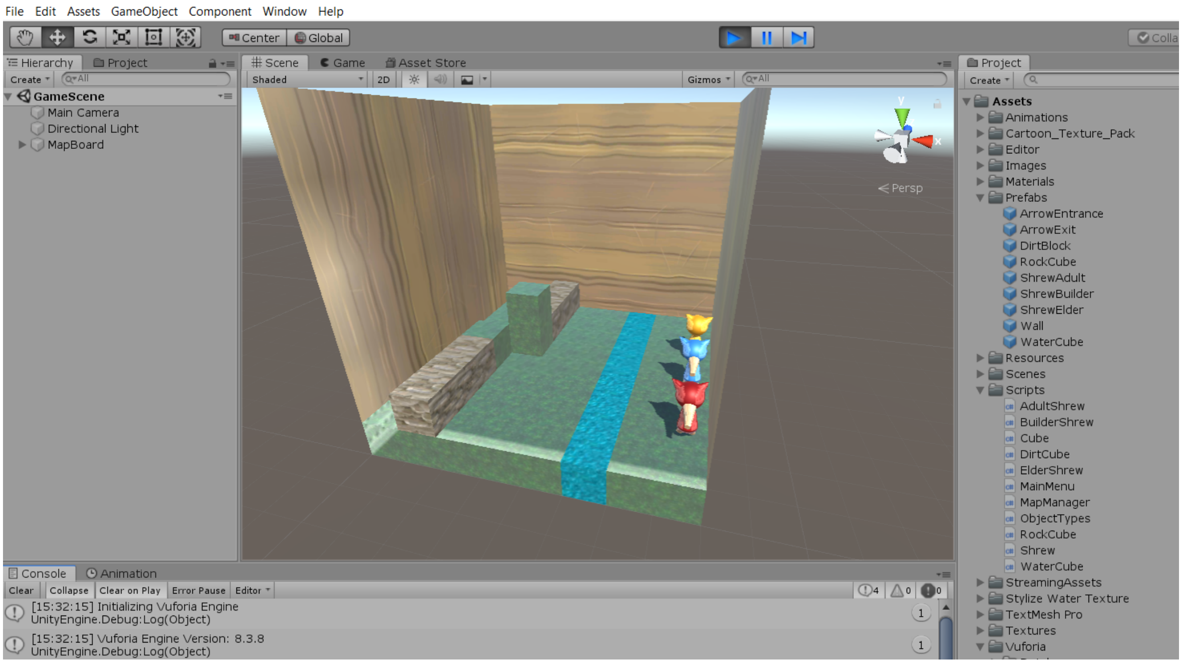


Figure 28: The general view of a Scene in Unity.

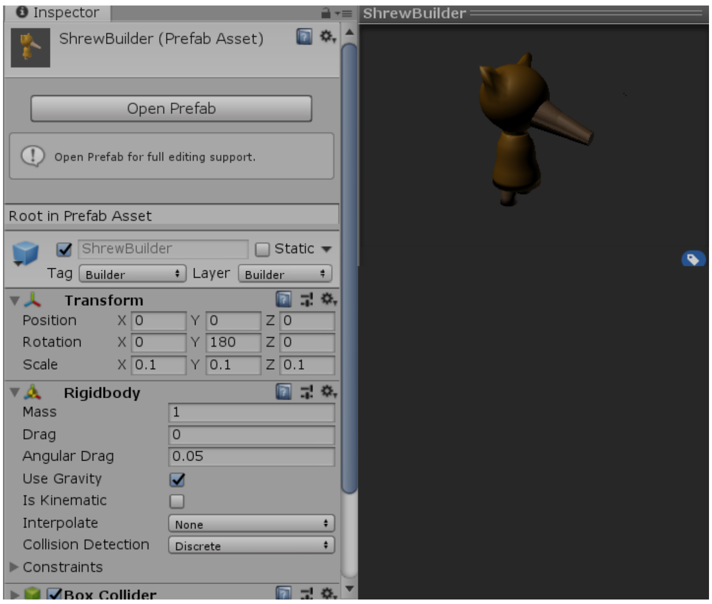


Figure 29: The properties view of game objects in Unity.

The following extract of C# code is an example of a piece of code in the movement test script of the game.

```
private void MoveShrew(ObjectTypes.Direction direction) {
    var pace = 50;
    Vector3 vector = new Vector3(0, 0, 0);

    BuilderShrew shrew = GameObject.Find("ShrewBuilder").GetComponent<
BuilderShrew>();

    //Turn Shrew
    if (shrew.Direction != direction)
    {
        shrew.TurnDirection(direction);
        return;
    }

    //Move Shrew
    pace = 50;
    switch (direction)
    {
        case ObjectTypes.Direction.UP:
            vector = new Vector3(0, 0, pace);
            ShrewObj.GetComponent<BuilderShrew>().TurnDirection(ObjectTypes.
Direction.UP);
            break;
        case ObjectTypes.Direction.RIGHT:
            vector = new Vector3(pace, 0, 0);
            ShrewObj.GetComponent<BuilderShrew>().TurnDirection(ObjectTypes.
Direction.RIGHT);
            break;
        case ObjectTypes.Direction.DOWN:
            vector = new Vector3(0, 0, -pace);
            ShrewObj.GetComponent<BuilderShrew>().TurnDirection(ObjectTypes.
Direction.DOWN);
            break;
        case ObjectTypes.Direction.LEFT:
            vector = new Vector3(-pace, 0, 0);
            ShrewObj.GetComponent<BuilderShrew>().TurnDirection(ObjectTypes.
Direction.LEFT);
            break;
    }
    ShrewObj.transform.position += vector;
    SoundManager.Instance.PlaySound(moveSFX);
}
```

5.2.1 *Scenes*

Scenes are where you work with the content and game objects inside the Unity editor, with scenes being able to hold all the information of game objects to be used once the game is running. A game can be made in a single scene but for a more complex project and for better organization we can go and group various scenes with each dictating a function or objective to fulfill. For this project various scenes were created to better organize the game objects to be loaded and run on certain times.

The Scenes created for the project were six in total, being a main menu, a testing scene for the movement of the player character to teach how the builder Shrew moves within the game, a scene to experiment on the actions of the builder Shrew on the various types of cubes in the game to teach the different types that exist and where they interact and move to, and three scenes each for the beginner, intermediate and advanced levels of the game, each with different puzzles to solve to test the players skills and experience in problem solving via Computational Thinking skills.

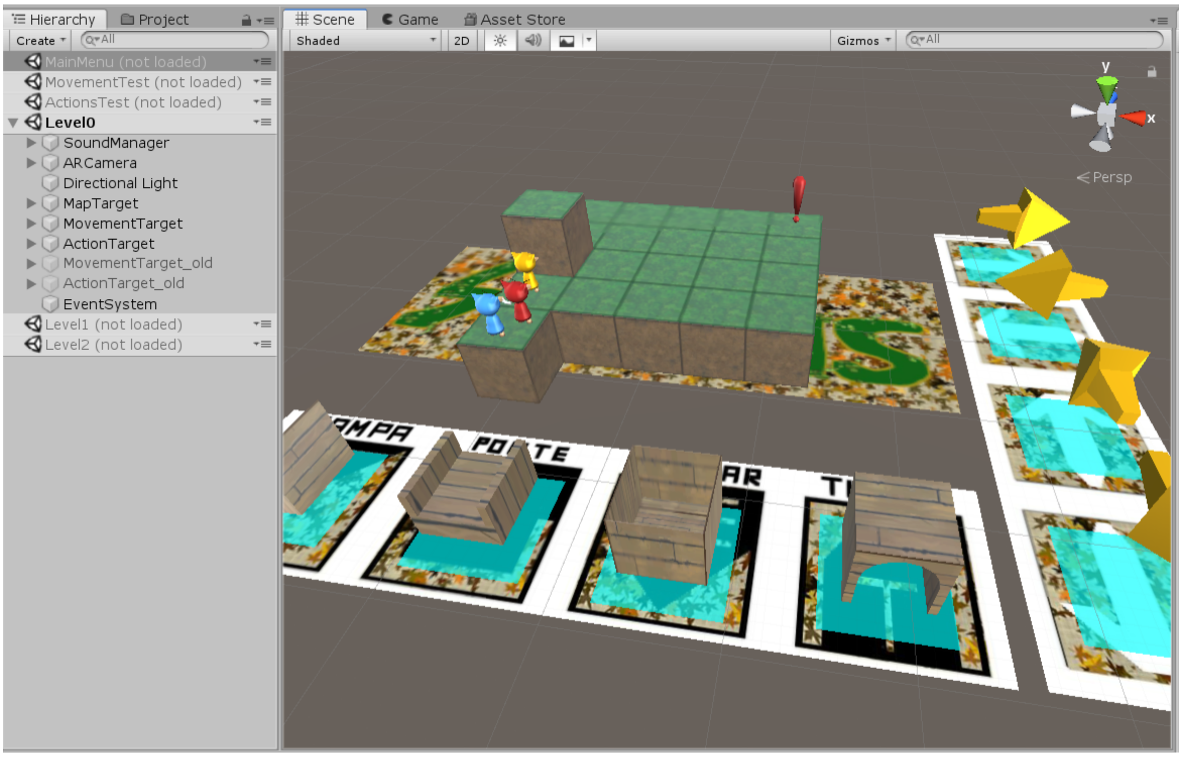


Figure 30: The Unity Editor showing the view of a scene and all the game objects associated with the scene.

Game Objects in a scene are rendered in an hierarchical order, from top to bottom, with these objects being able to be set as children creating a tree like system where the lower nodes inheriting their some of their properties such as the positioning on the scene, then on each of the children, sub positioning can be set relative to the parent node object, shown in Figure 31 a tree of the Game Objects in the beginner level scene. When organizing scenes we had to take attention to the situation of the scene in question, as for example, if it is solely a menu scene, or will it be projected with the Augmented Reality camera turned on.

As for the main menu scene where it is only UI that is presented, the placement of the objects are to be placed relative to the camera's position as to show the whole UI in view and filling the screen with visual indicators such as images, a background image, menu buttons spread as not to leave the scene empty and void. Measures of configuring the screen resolution to be as responsive as possible so that different devices showed consistency in the presentation of the UI were taken by taking an approach of focusing the game's resolution on a 16:9 widescreen display.

As for the Scenes that utilize the Augmented Reality camera, placing UI Game Objects over the camera is possible as long as not to obscure the vision of the player when playing the game, therefore simple text indicators and small buttons as an overlay were created shown by Figure 32, keeping the information of the game to the player, while also presenting on screen information and options.

Figure 31: An example of the beginner level scene with the associated game objects and their children.

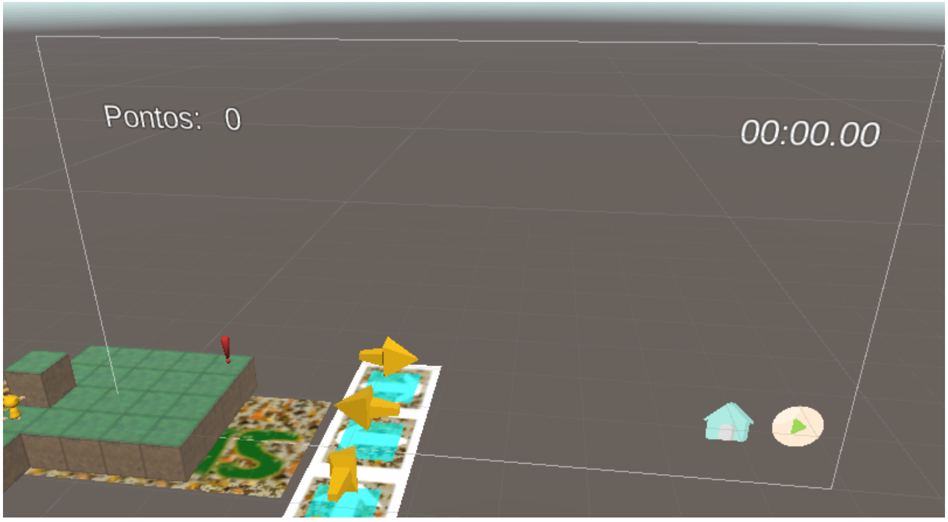


Figure 32: An example of the placement of game objects as an overlay over the game scenes.

5.2.2 *Game Objects*

Game Objects in Unity are the most important concept, since everything in a game, is a Game Object, but with different types and properties. A Game Object can be a character, an item, a camera or a light, and each of these types have different properties and can be shaped and defined to fit the objective and purpose of it. These properties and features are known as Components and they need to be defined and set to Game Objects, as Game Objects by themselves do not accomplish much other than being a static container.

For this project various Game Objects were created and added onto scenes to represent the terrain on maps, the characters of the Shrews, the Vuforia Augmented Reality Camera, the lights in the game and also the Augmented Reality target markers with Augmented Reality buttons as well. Examples during prototype phasing at figures 33, 34 and 35.

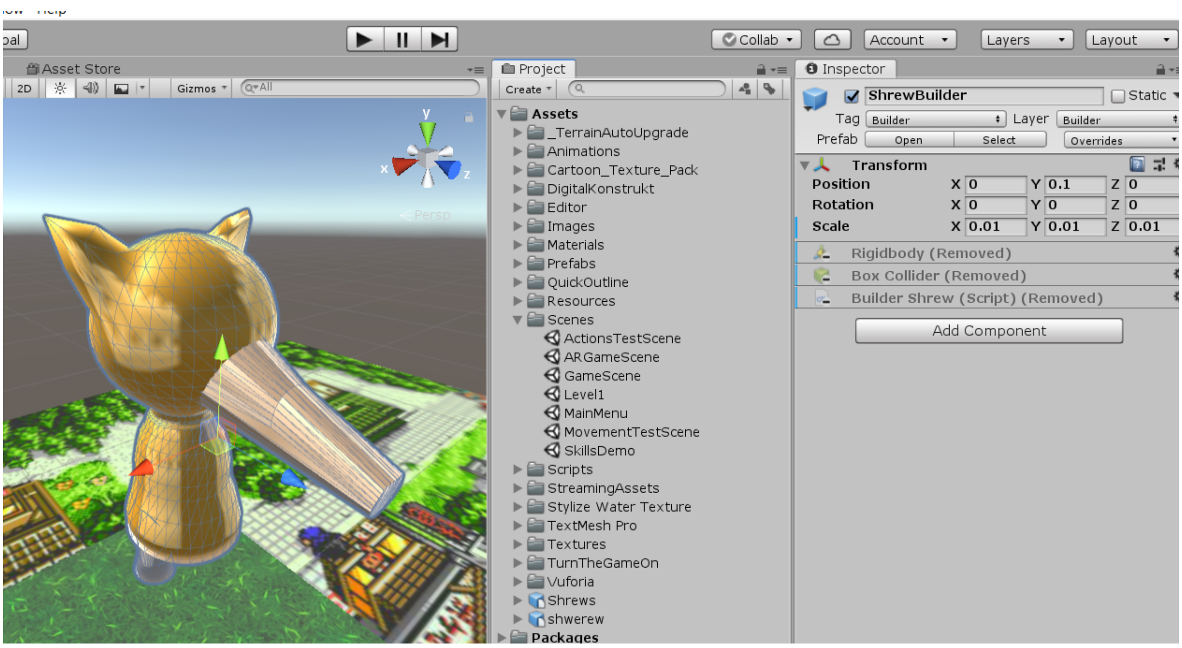


Figure 33: A builder Shrew game object with only a Transform component in its properties.

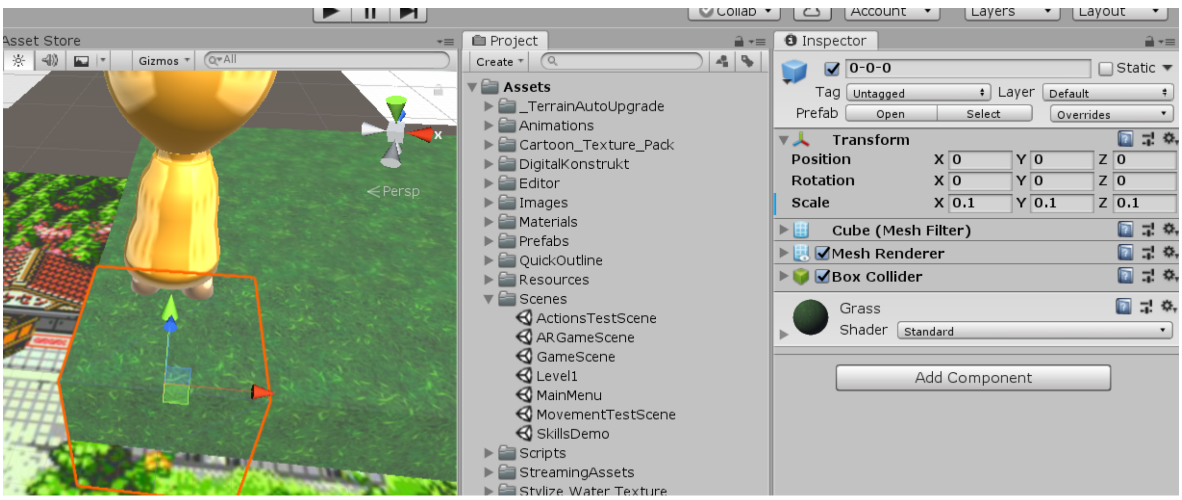


Figure 34: A Cube Game Object with its components set of Transform, Meshes, Colliders and a Texture.

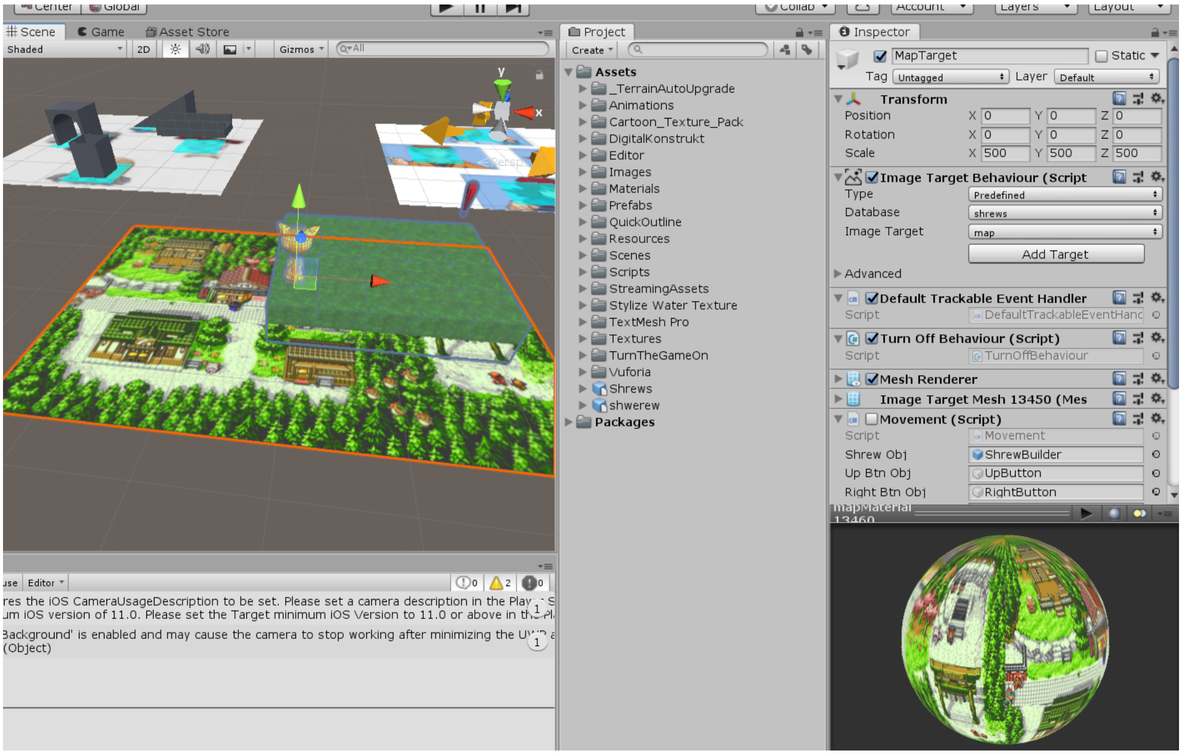


Figure 35: The map target image with various components related to Vuforia such as the image target behaviour script from Vuforia and a custom movement script.

For some of the game objects created, they have properties that are intrinsic and static to them, and need to be placed various times like the cubes that make up the map. To avoid mismatching and making the game unorganized and having more workload to assure the game objects are placed correctly and properly, any that can be repeated should be turned into whats called a Prefab, which will be discussed in the following Chapter.

5.2.3 *Prefabs*

Prefabs in Unity allow the re utilization of Game Objects. Once we have a set Game Object to be re used in various scenes, the best solution is a Prefab, which acts as a template that can be shaped into yet again other Prefabs. Any edits or changes to an existing Prefab is automatically reflected on all instances and scenes, further saving time and keeping the system consistent. For the purpose of the project various Prefabs were created for the Cubes, Shrews and Structures of the game.

On Figure 36 is the list of all the prefabs in the project. The important ones to note in this project are the cube prefabs (see Fig 37), the building prefabs (see Fig 38), and the Shrew prefabs (see Fig 39). On all prefabs the scales were set accordingly with the correct sizes for consistency and presentation in the game, also within these prefabs are their according scripts which dictate a lot of the logic and rule checking of the game, by identifying the component and script of a certain object we can accurately predict which objects are where and what they are and do.

As for an example of use, take Figure 40, which shows an imported free 3d model of a ramp on their default scale and texture on the left, compared to the same modified 3d model on the right, but oriented to use in the game with a set scale, rotation, texture and script components, therefore we can modify the game object to our needs, save it and re-use it throughout the game without needing to modify it each time it is imported.

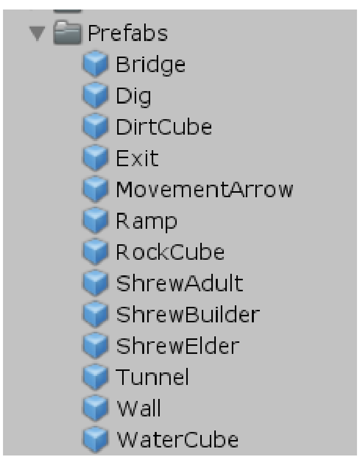


Figure 36: The list of prefabs in the Shrews game project. Modifying these game objects reflect the changes across all game scenes.

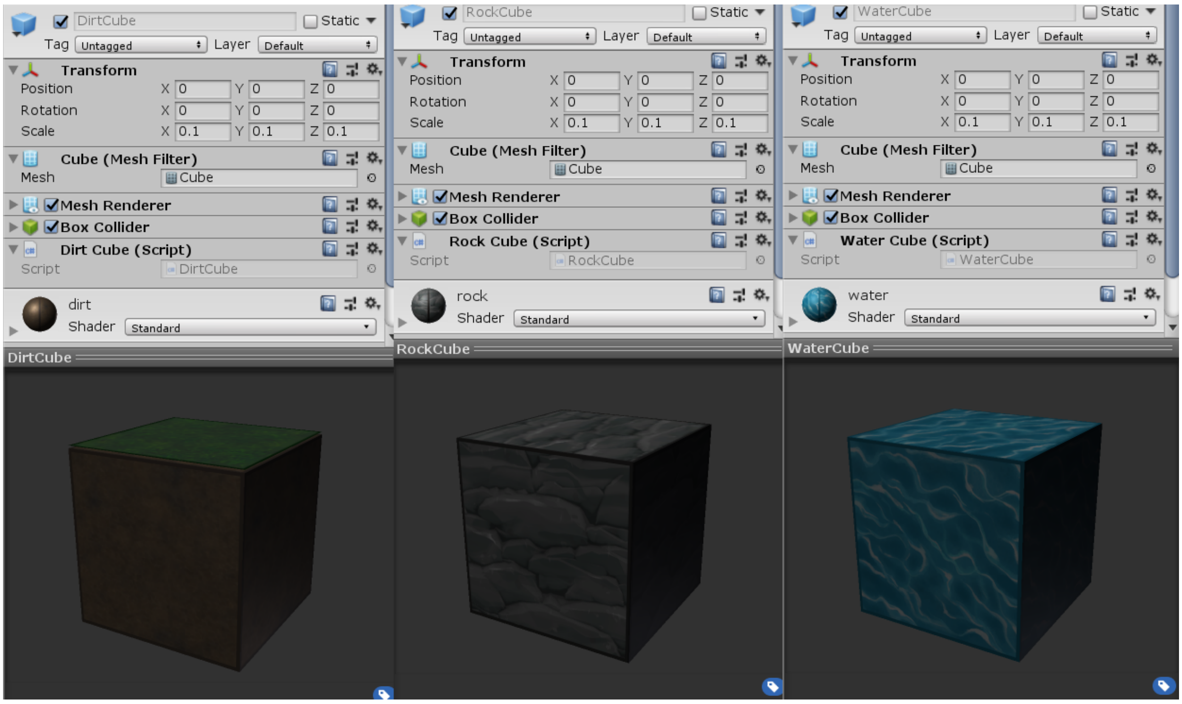


Figure 37: The three types of cube prefabs have their scales set to 0.1 to properly be represented in the game, also each with their according script component and game logic.

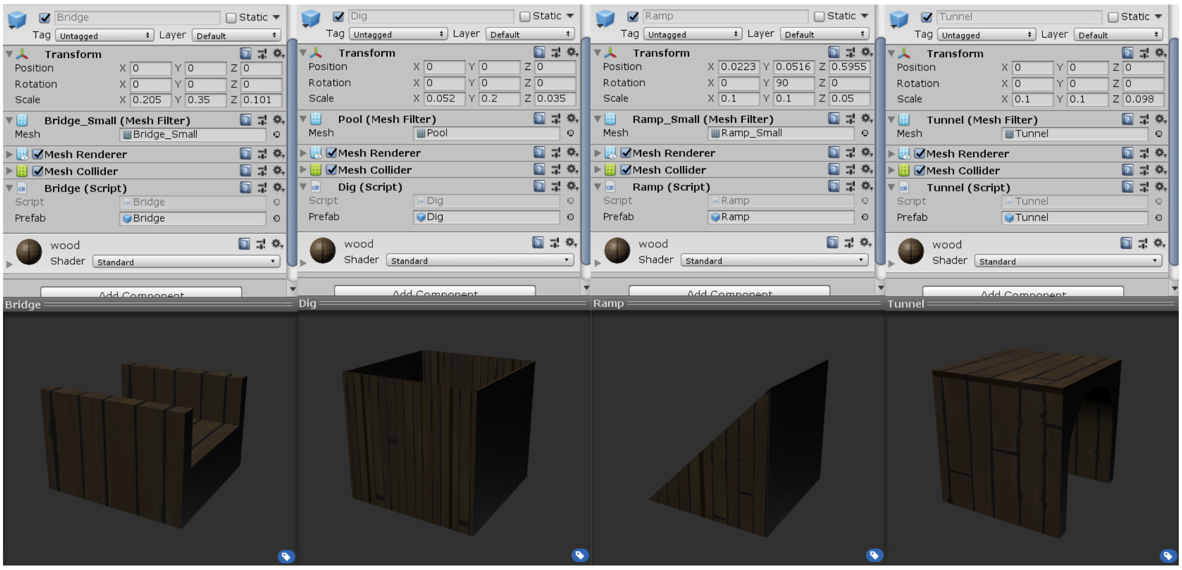


Figure 38: The building prefabs have each different scales set from their original imported 3d models, so that they are visually consistent with the game's aspect. They also carry their own scripts with game logic.

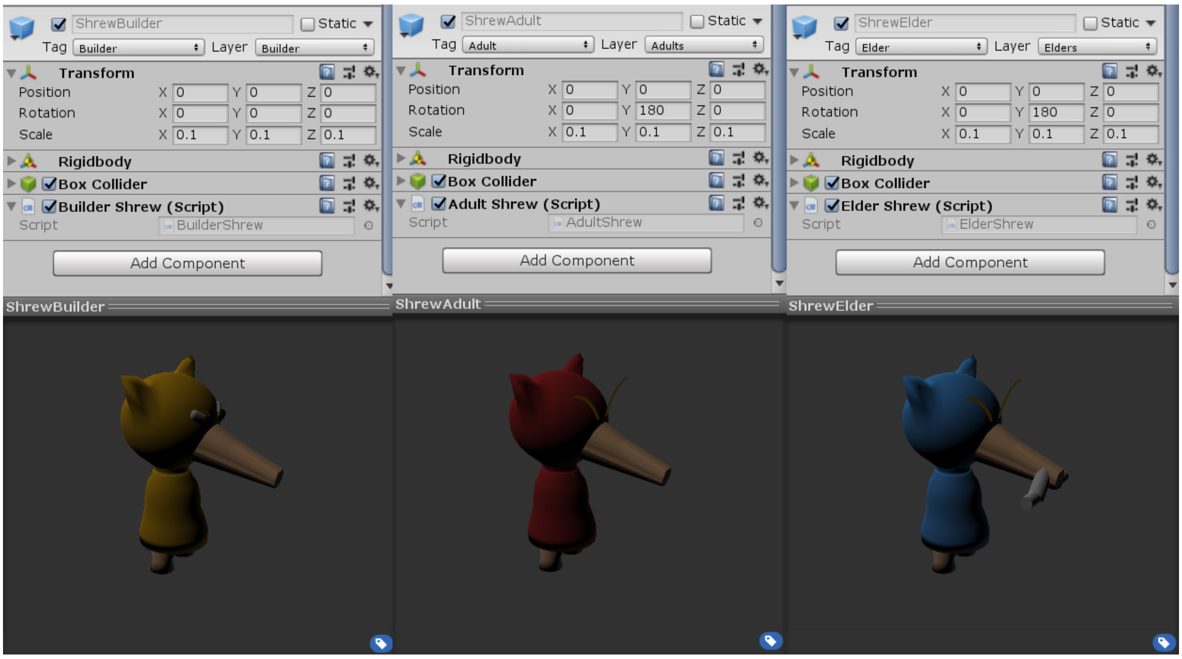


Figure 39: The three Shrew characters in game with their distinct appearances and scales set, they also carry script components each with their respective game logic.

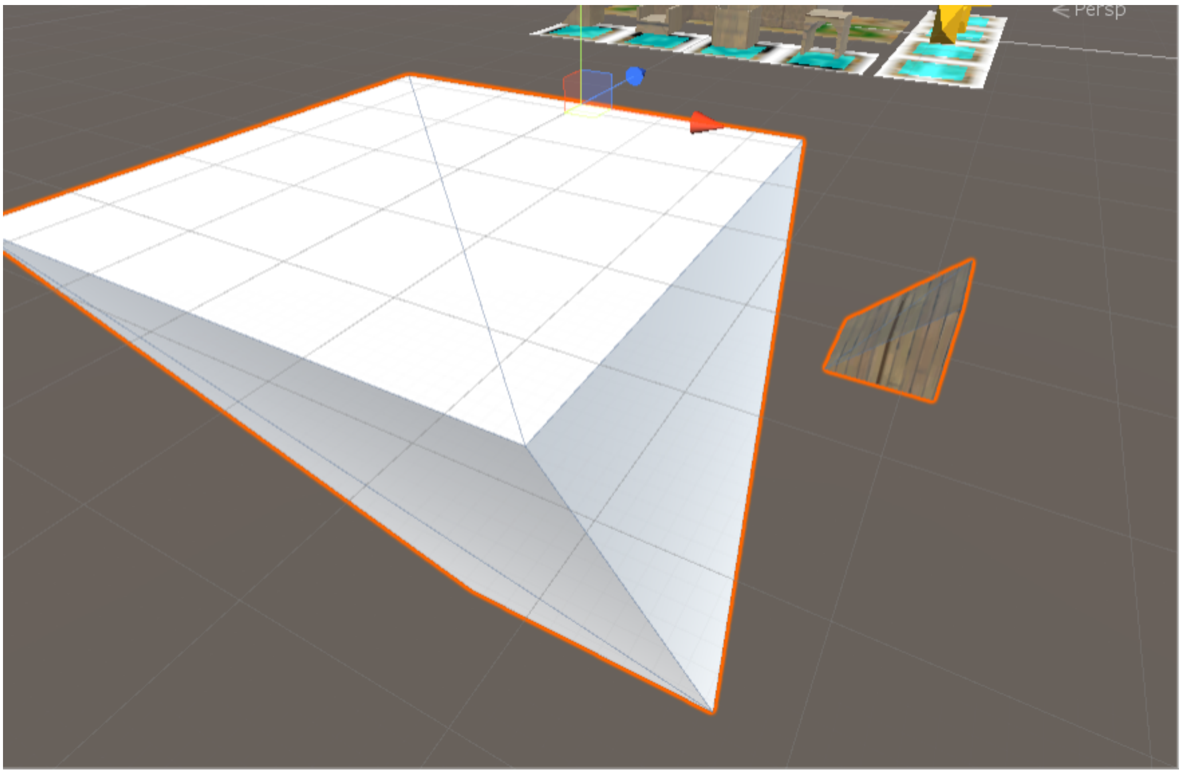


Figure 40: This Figure demonstrates the difference between and imported 3d model on the left and the final result of a modified one on the right, saved as a prefab to use in the game.

5.2.4 *Scripts*

Unity has a lot of already built in components and behaviours for its game objects but on developing games there is a point the programmer needs to go beyond the scope of already available tools. That is why Unity offers the creation of custom components using scripts that allow to trigger game events and modify just about anything with the available Unity engine. These scripts are supported in various languages with the main staple being the C# programming language. These scripts are then attached to game objects giving them whats called a mono behaviour, that inherit various Unity functions such as Start() that runs on the first frame of the game and Update() that runs every 1 frame of the game.

For this project various scripts (see Figure 41), were created for the movement of the Shrews, the logic in the Shrews actions and other game logic and rules for the gameplay.

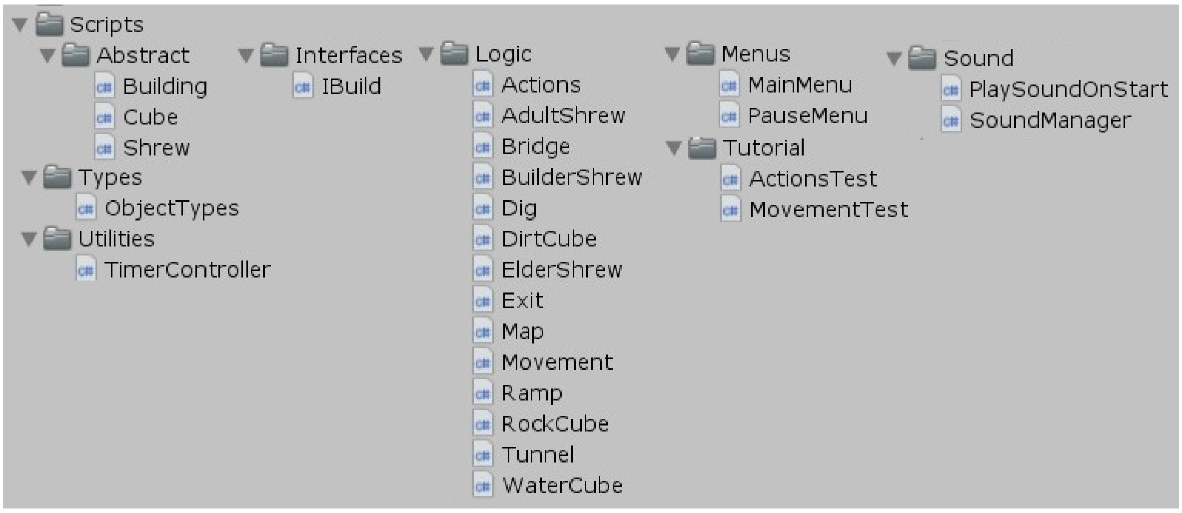


Figure 41: The list of all the scripts in the game.

The development and organization of the project followed an Object Oriented Programming approach, with the use of abstractions in the buildings, cubes and Shrews. These abstract classes (see Listing 5.1), have properties and methods pertinent to every class that inherits from them, like the builder, adult and elder Shrew. These helped generalize and handle code organization much easier.

```
public abstract class Shrew : MonoBehaviour
{

    public int X { get; set; }
    public int Y { get; set; }
    public int Z { get; set; }
    public double Speed { get; set; }
    public int FallDamage { get; set; }
    public ObjectTypes.Direction Direction { get; set; }
```

```
public bool isOnRamp { get; set; }
public List<Vector3> path { get; set; }
public List<Vector3> positions { get; set; }
public List<ObjectTypes.Direction> directions { get; set; }

public virtual void Start()
{
    this.Direction = ObjectTypes.Direction.UP;
    this.X = 0;
    this.Y = 1;
    this.Z = 0;
    this.path = new List<Vector3>();
    this.positions = new List<Vector3>();
    this.directions = new List<ObjectTypes.Direction>();
}
...
```

Listing 5.1: A piece of code from the Shrew abstract class

To touch on the important implementations points, we'll discuss the implementation of the logic of the inner puzzle or matrix, how the movement is handled both in Unity and internally in the code.

On the surface, the puzzle is presented by a map in a group of cubes, each identified in their name by the coordinates of their position inside a 3 dimensional matrix of 5x4x4, represented in Figure 42.

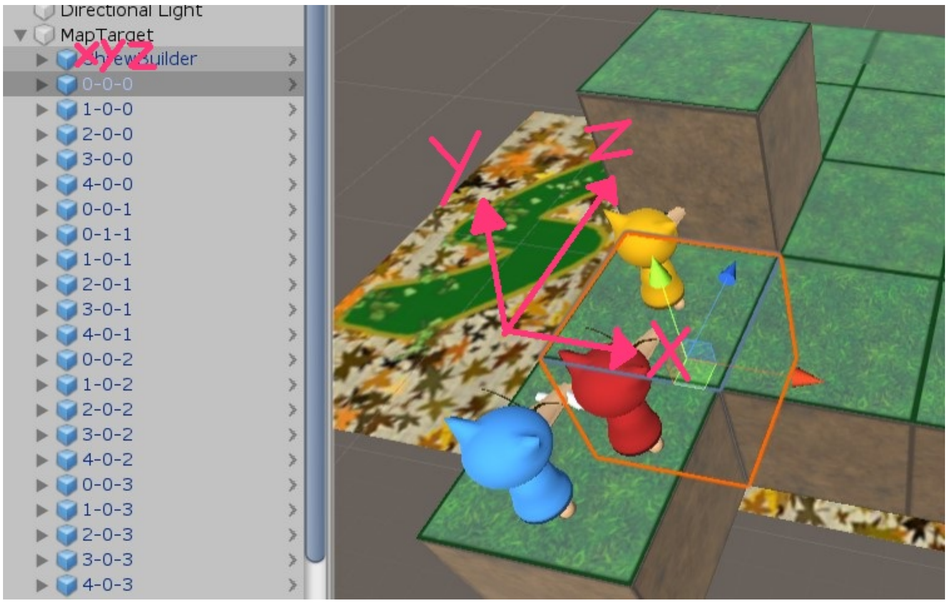


Figure 42: The organization of the three coordinates of the game.

Knowing exactly the positions of every object in the 3d matrix in run time allows to dictate and handle the rules of the game with precision. The game objects follow certain coordinates that need to be implemented in the Unity scene editor, but internally saving the map and object coordinates when loading the map is easily done. Take the following excerpt of code that runs when loading the beginner level of the game at listing 5.2. Each game object with their appropriate coordinate in their name, is stored inside the matrix, which then is easily accessible when handling game logic. Since every game object of the game has their according script attached, for example dirt cube or ramp building, we can identify looking into the coordinate for what specific object is in that position.

```
void Level0()
{
    //x-y-z
    //Level0
    //height 0
    MAP.SetValue(GameObject.Find("0-0-0"), 0, 0, 0);
    MAP.SetValue(GameObject.Find("1-0-0"), 1, 0, 0);
    MAP.SetValue(GameObject.Find("2-0-0"), 2, 0, 0);
    MAP.SetValue(GameObject.Find("3-0-0"), 3, 0, 0);
    MAP.SetValue(GameObject.Find("4-0-0"), 4, 0, 0);

    MAP.SetValue(GameObject.Find("0-0-1"), 0, 0, 1);
    MAP.SetValue(GameObject.Find("1-0-1"), 1, 0, 1);
    MAP.SetValue(GameObject.Find("2-0-1"), 2, 0, 1);
    MAP.SetValue(GameObject.Find("3-0-1"), 3, 0, 1);
    MAP.SetValue(GameObject.Find("4-0-1"), 4, 0, 1);

    MAP.SetValue(GameObject.Find("0-0-2"), 0, 0, 2);
    MAP.SetValue(GameObject.Find("1-0-2"), 1, 0, 2);
    MAP.SetValue(GameObject.Find("2-0-2"), 2, 0, 2);
    MAP.SetValue(GameObject.Find("3-0-2"), 3, 0, 2);
    MAP.SetValue(GameObject.Find("4-0-2"), 4, 0, 2);

    MAP.SetValue(GameObject.Find("0-0-3"), 0, 0, 3);
    MAP.SetValue(GameObject.Find("1-0-3"), 1, 0, 3);
    MAP.SetValue(GameObject.Find("2-0-3"), 2, 0, 3);
    MAP.SetValue(GameObject.Find("3-0-3"), 3, 0, 3);
    MAP.SetValue(GameObject.Find("4-0-3"), 4, 0, 3);

    //height 1
    MAP.SetValue(GameObject.Find("0-1-1"), 0, 1, 1);
}
```

Listing 5.2: The beginner level map being loaded

The movement of the game is handled both by updating the internal coordinates of the builder Shrew inside of the matrix, while also moving the game object coordinates, relative to the map target, which the objects are children objects. For an example, when the builder is on top of the dirt cube "0-0-0", which is the first cube, the current position of the builder will be 0-1-0 inside of the a matrix. Moving forward, will update the builder to 0-1-1, as the player has moved in the Z axis. To handle visually the information, the characters must translate their coordinate appropriate with the cube that they are positioned inside of the matrix.

For the sake of consistency, in the context of the Unity coordinates, the pace at which the characters move is of 0.1 units, in the X, Y and Z axis, with the specific rule that while on a ramp, the position on the Y axis is relative to the height times 0.057 units (height x 0.057). The following piece of code at listing 5.3, shows an example of the movement method.

```
private void MoveShrew(ObjectTypes.Direction direction)
    {
        var pace = 0.1f;
        Vector3 move = new Vector3(0, 0, 0);
        int x = 0, y = 0, z = 0;

        //first , Shrew needs to change direction on the first tap
        BuilderShrew shrew = GameObject.Find("ShrewBuilder").GetComponent<
    BuilderShrew>();

        if (shrew.Direction != direction)
        {
            shrew.TurnDirection(direction);
            return;
        }

        //second, if the direction is the same as the shrew already is in, he
    moves.
        //+x up
        //-x down
        //+z right
        //-z left

        switch (direction)
        {
            case ObjectTypes.Direction.UP:
                move = new Vector3(0, 0, pace);
                z = 1;
                log += "up\n";
                break;
            case ObjectTypes.Direction.RIGHT:
                move = new Vector3(pace, 0, 0);
```

```
                x = 1;
                log += "right\n";
                break;
            case ObjectTypes.Direction.DOWN:
                move = new Vector3(0, 0, -pace);
                z = -1;
                log += "down\n";
                break;
            case ObjectTypes.Direction.LEFT:
                move = new Vector3(-pace, 0, 0);
                x = -1;
                log += "left\n";
                break;
        }
        shrew.directions.Add(direction);
        MakeMove(move, x, y, z, direction);
    }
```

Listing 5.3: The method responsible for moving the builder Shrew

5.2.5 *Vuforia Camera and Virtual Buttons*

Unity's presentation works by the Camera Game Object that exists within the system, this camera is responsible by looking at the view intended for the game to be played and run from. For this project to utilize the camera tied to the device it is running from, the use of the Vuforia package and their already existing Game Objects were used, and the most important being the Augmented Reality Camera Game Object.

The Augmented Reality Game Object functions similarly to the default Unity Camera Game Object as it presents what the player sees, but utilizes the devices camera to see through it, so it can display the Augmented Reality objects in the real world as shown in example in Figure 43. A more technical explanation of the inner workings of how Vuforia works is present in Chapter 5.3, with the explanation of Augmented Reality Markers in Chapter 5.5 of this document.

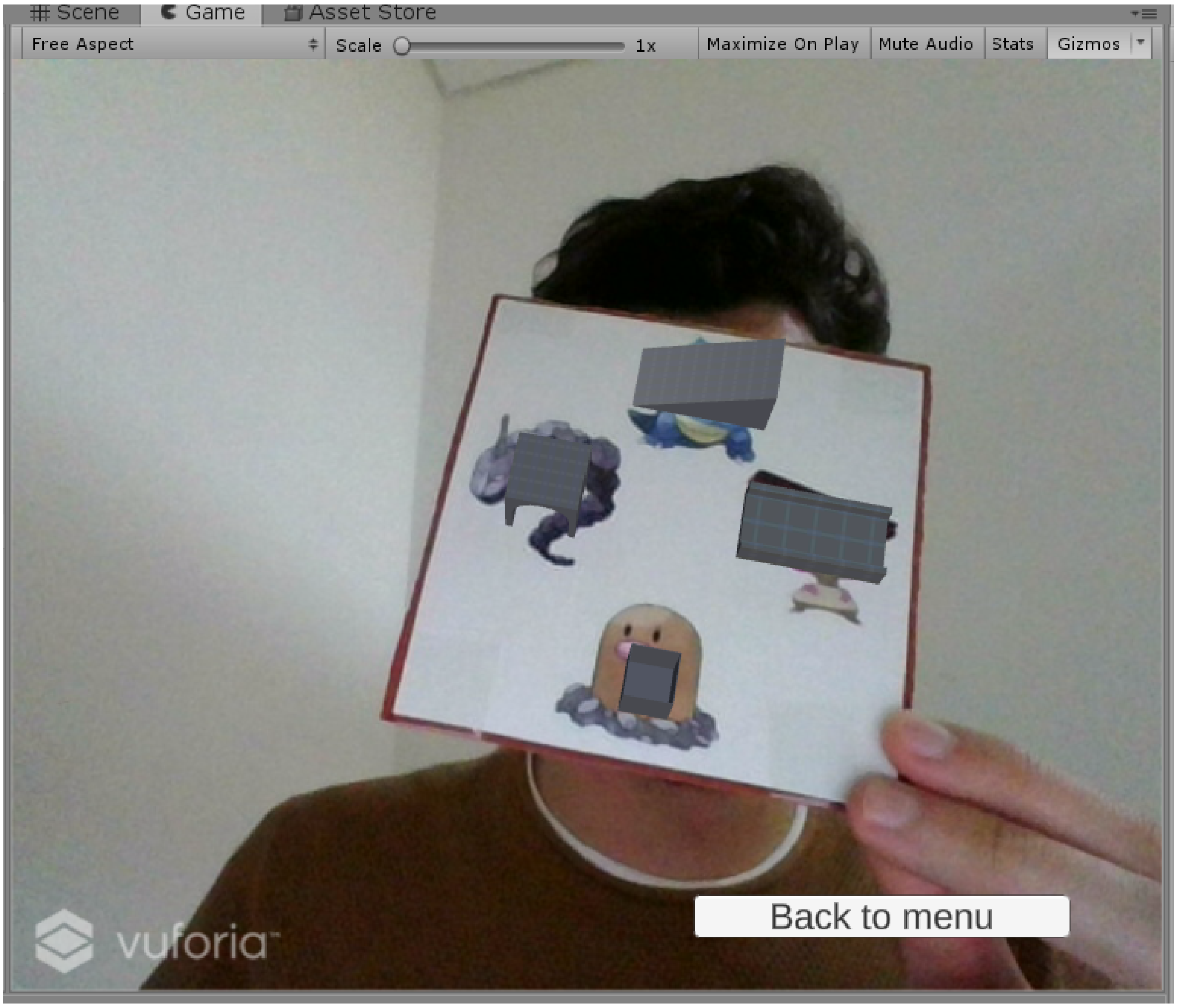


Figure 43: The Vuforia Augmented Reality camera uses a device's camera to track for image targets to display the AR.

Another aspect and functionality created for this project was the feature of Virtual Buttons. Augmented Reality Buttons or Virtual Buttons are a feature Vuforia gives that allows for interactivity with image targets or screen interactions to the real world. These buttons are added via Unity, layered on top of an image target, placing them in an area, that once detected interaction in said area, will signal a behaviour to be processed defined by the programmer via custom scripts.

There are a total of eight virtual buttons in the game, with four being responsible for the movement of the builder Shrew and four for the actions the player and builder Shrew can create. These server as the buttons for direct interactive gameplay.

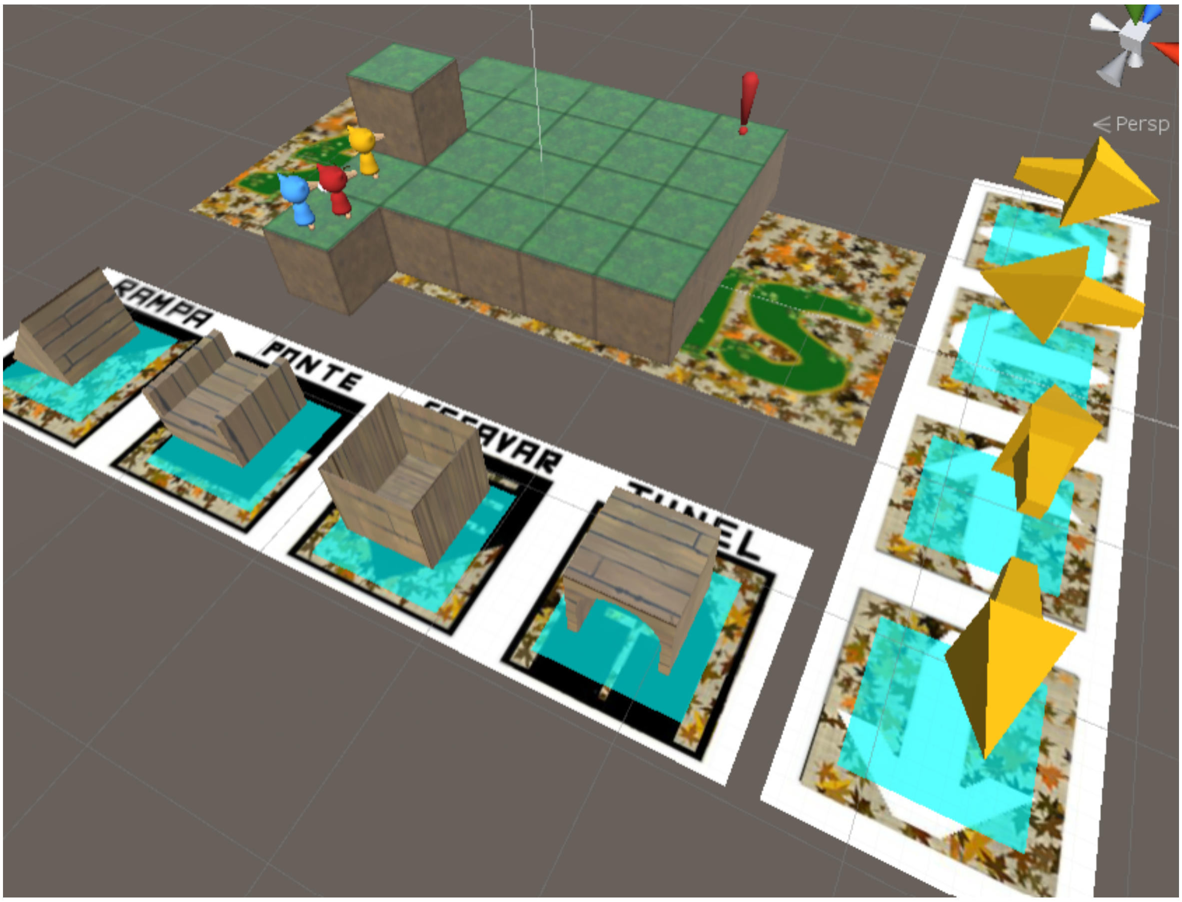


Figure 44: The virtual buttons, in blue, are put on top of the image target's area to indicate which areas of the target are interact-able.

5.3 VUFORIA

Vuforia ® is an Augmented Reality software development kit used for mobile devices to enable the creation of Augmented Reality applications. For the purpose of this project, a library extension to Unity was used to develop the Augmented Reality the game uses. This projects uses as basis the version 9.8 library. Vuforia is used to generate the Augmented Reality elements in applications via the use of Augmented Reality cards. Augmented Reality cards have specific imagery and or shapes that are recognized by Vuforia and know where to generate the AR. More information in the markers section of this Chapter.

Vuforia works by analysing images and finding features to identify them when searching from the camera. It turns them into grayscale, and looks for features (see Fig 45) indicated by the yellow features. These features are what identifies an image, by looking at patterns and seeing the shapes and small points of notice in them.

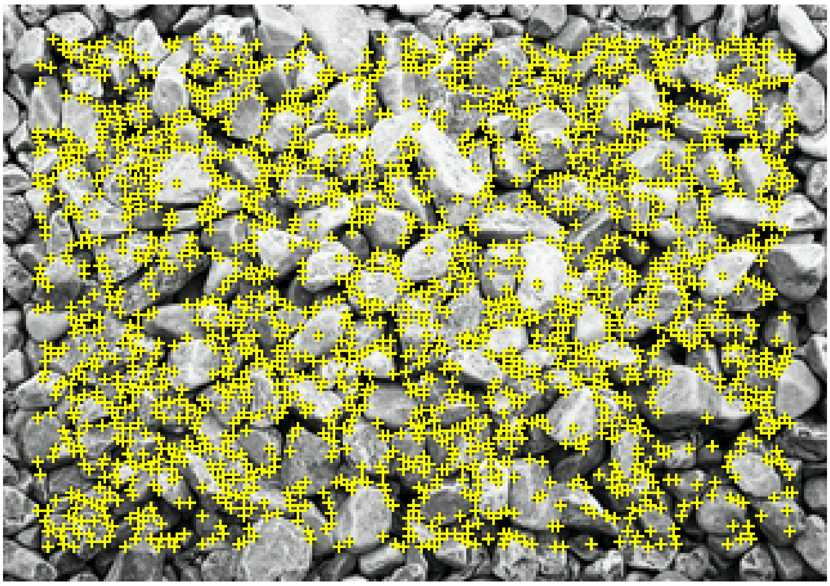


Figure 45: Features detected by Vuforia.

The quality of the marker is rated on many factors (see Fig 46). One of them being features, which are more easily detected if the shapes are more edged, for example a square shape is better recognized than a circle has it has four corners for four features, while a circle has none. The second is the contrast, contrasting colors and hues are better for detection than monotone or too similar colors.

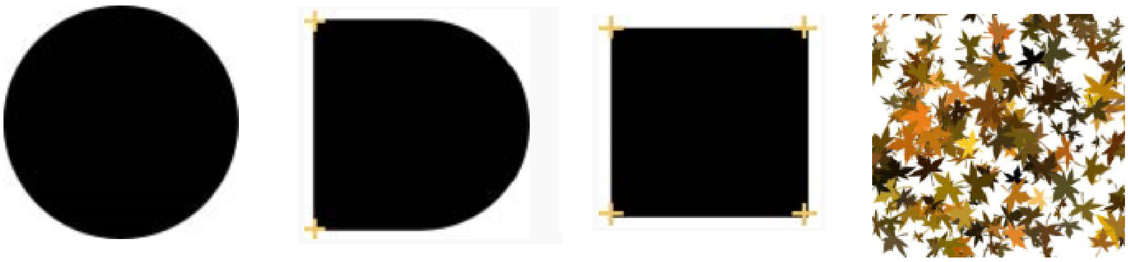


Figure 46: Rating of images are based on feature recognition.

After analysing and defining the markers, Vuforia generates a database with all the coordinates needed to recognize the markers based on their features, that can be imported in Unity.

First to configure Vuforia one must create an account over at developer.vuforia.com to adquire a license to use the development kit. A free version of Vuforia was used for this project. Choosing the Get Basic license over at the License Manager page, Figure 47, a free version will be created and a license key given, Figure 48, this license is then pasted in the Vuforia configuration inside Unity Editor as in Figure 47.

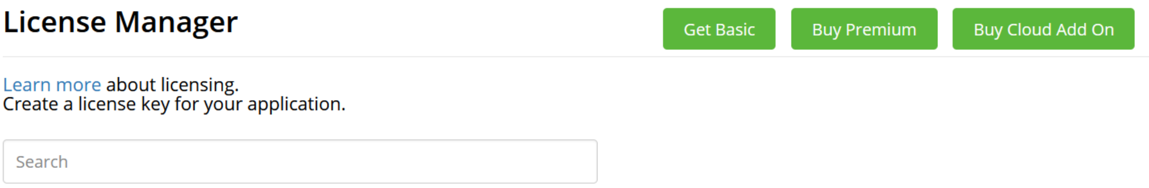


Figure 47: License Manager page in the Vuforia developer website.

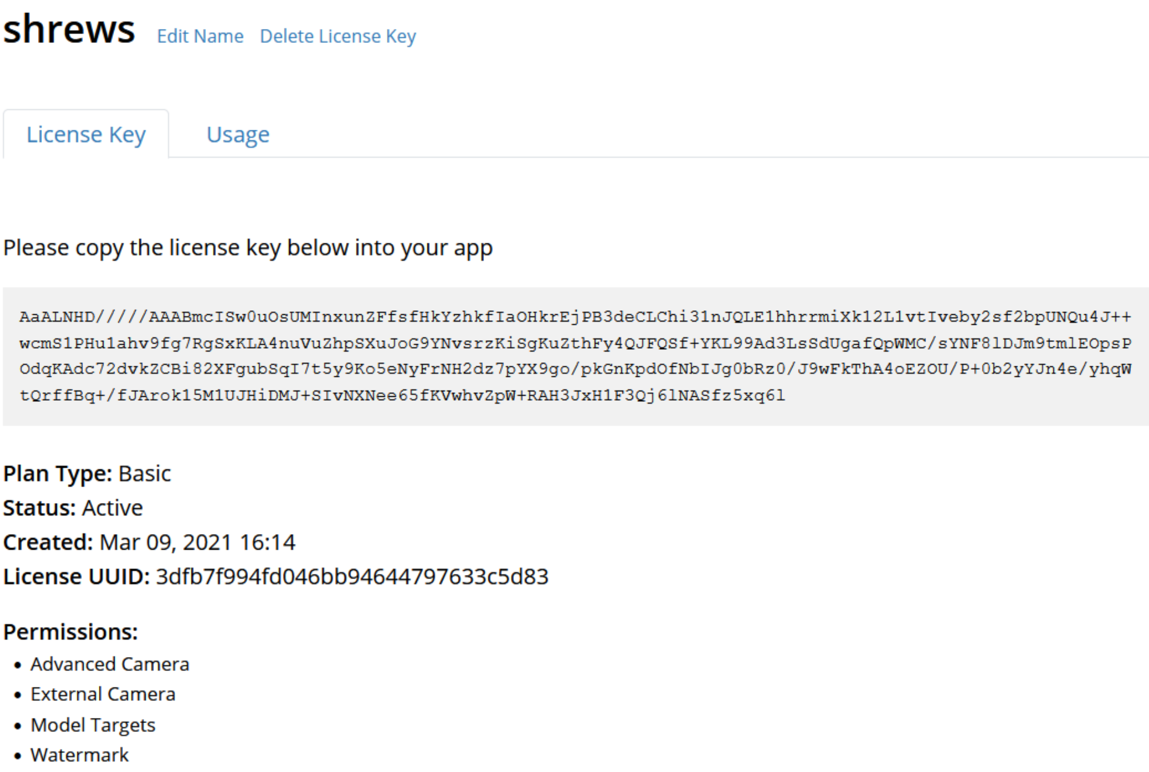


Figure 48: Information about the free Vuforia license generated.

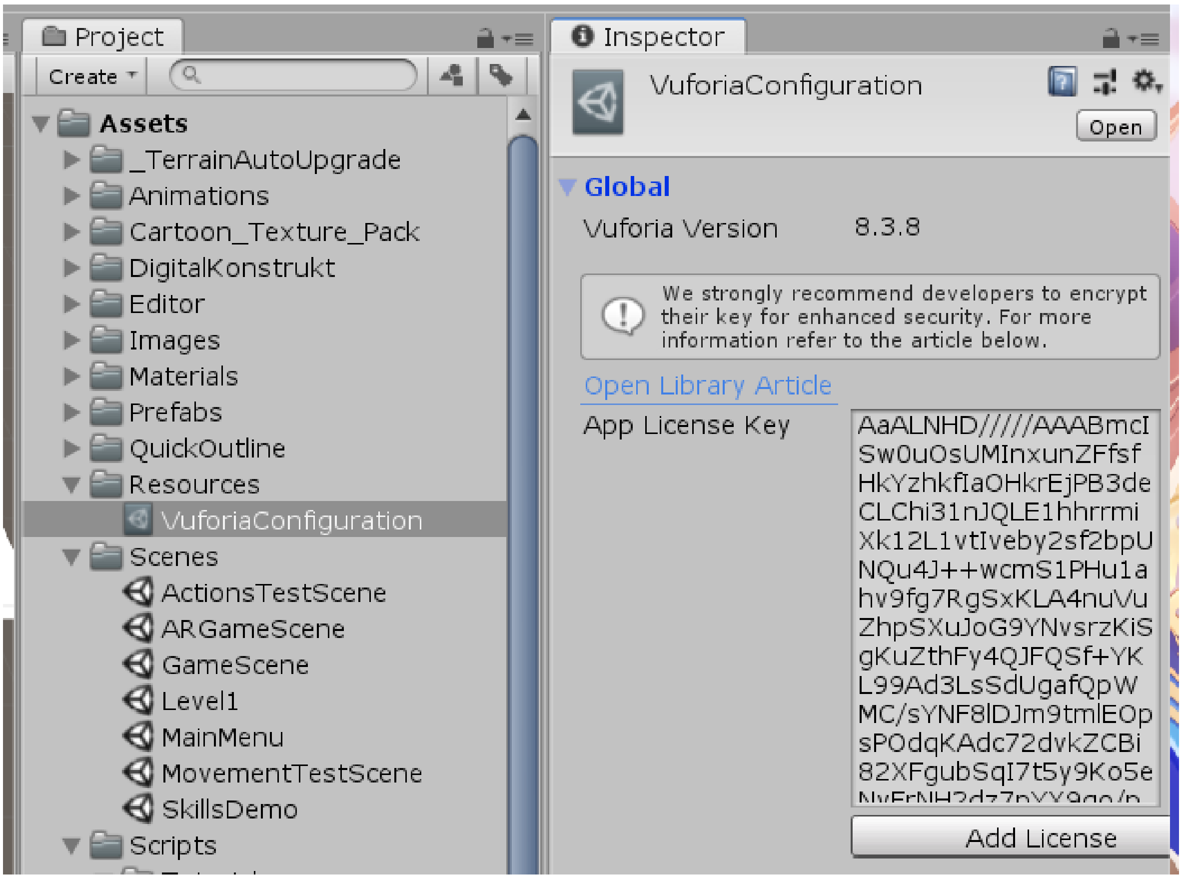


Figure 49: License pasted in the Vuforia configuration inside Unity.

After the license is configured, going back to the Vuforia developer website, opening the Target Manager page 50, a database was created that hosts the information of the image targets used for the application. Pressing the "Download Database" button provides a group of files that once imported to Unity, will have all the information needed to have Vuforia identify the Augmented Reality markers inside the Unity game engine. Three image targets were created, actions, arrows and map 51.

Vuforia analyses the images used and searches for unique features and patterns inside the image, that will then be used to recognize the target with more precision and ease, as an example in 52 the arrows image used to movement have many unique points and features of reference allowing Vuforia to easily identify the image as a Augmented Reality target.

After the targets are defined in the target manager, pressing the Download Database (All) button and choose the Unity editor option, will provide us with all we need to import directly into Unity. With this the database is saved within the game, and the target markers can now be selected under the Image Target game object.

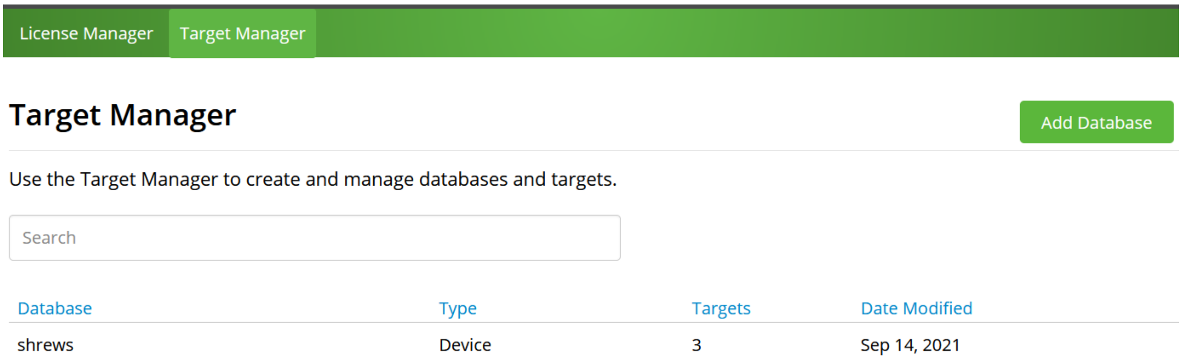


Figure 50: Target Manager page on the Vuforia developer page.

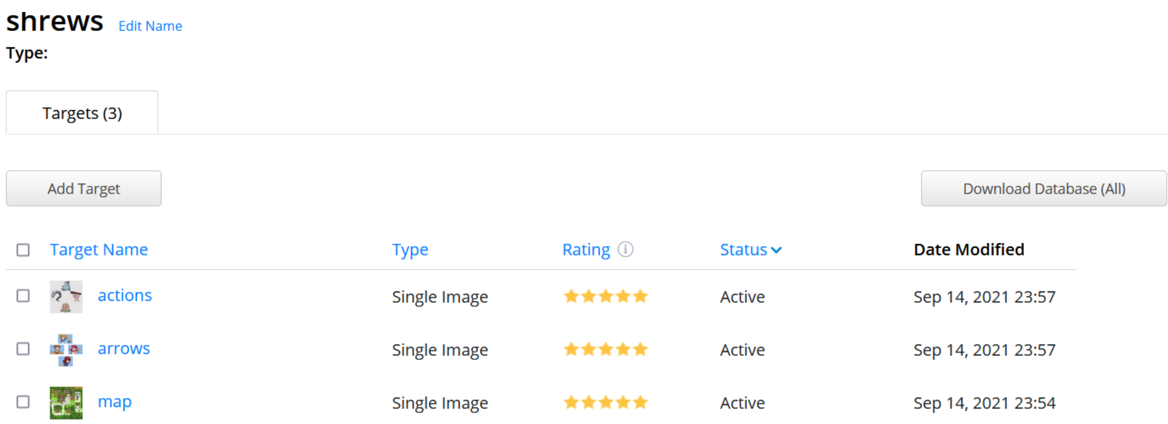


Figure 51: Images in the Vuforia database for the project.

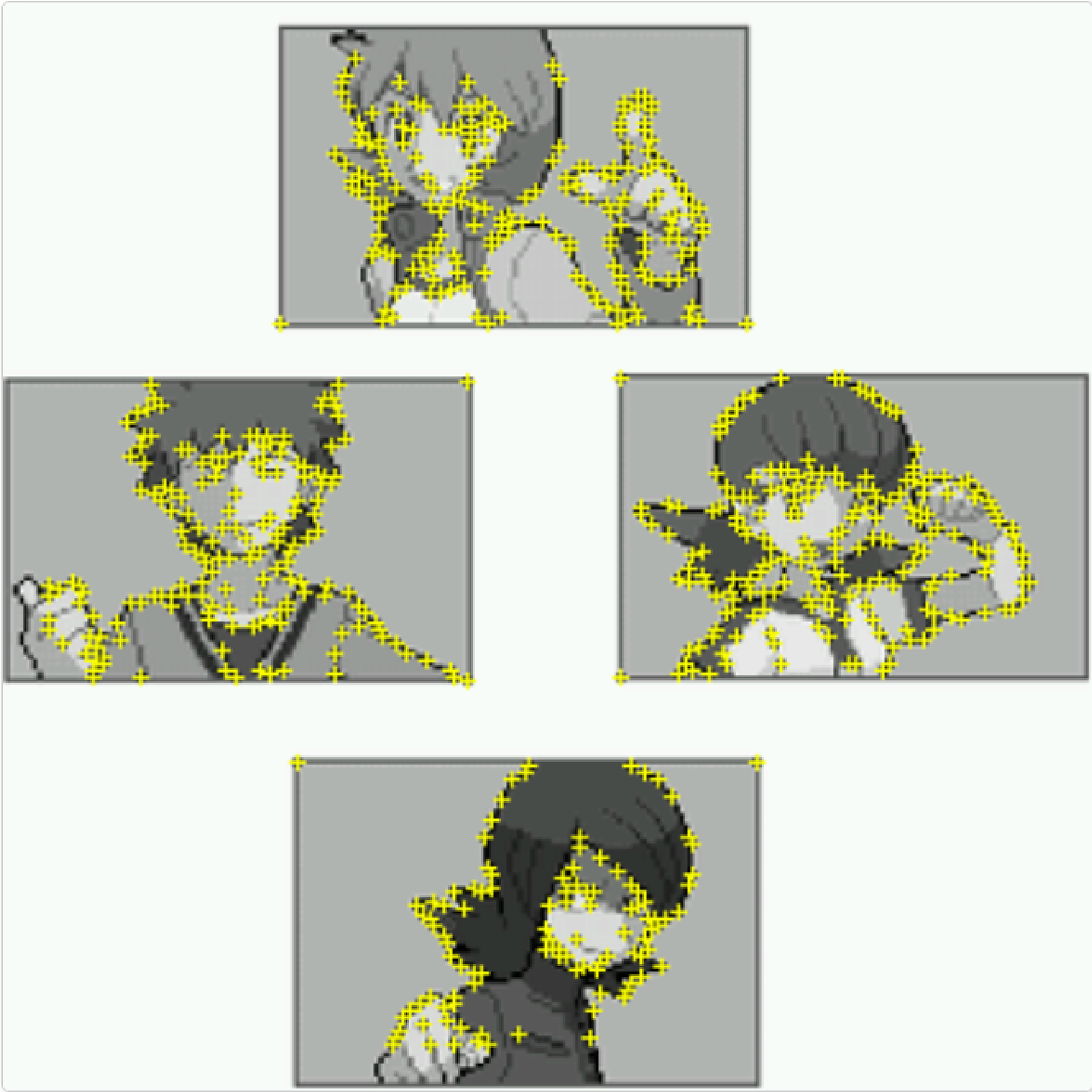


Figure 52: The features of the arrows image target, showing the points of reference that Vuforia uses to read the Augmented Reality image.

5.4 ASSETS AND ART

The work needed for the creation of the artefact was not only for the creation of the Video Game logic itself, but also the visual and aesthetic look of it. Good visuals and presentation is key to transmitting visual information to the player without needing to explain it in words. For this task various software was used in various creations.

5.4.1 *Visuals*

Blender ® is a free and open source 3-D computer graphics software tool used to create animations, visual effects, art, 3-D models, 3-D applications and more. For this project the only 3d model created in Blender was the basis of the Shrew model, shown in Figure 53. Being composed of simple shapes as the head is a circle, with two ears, followed by a cone shape to symbolize the nose, with a torso in a vase like shape and two feet.

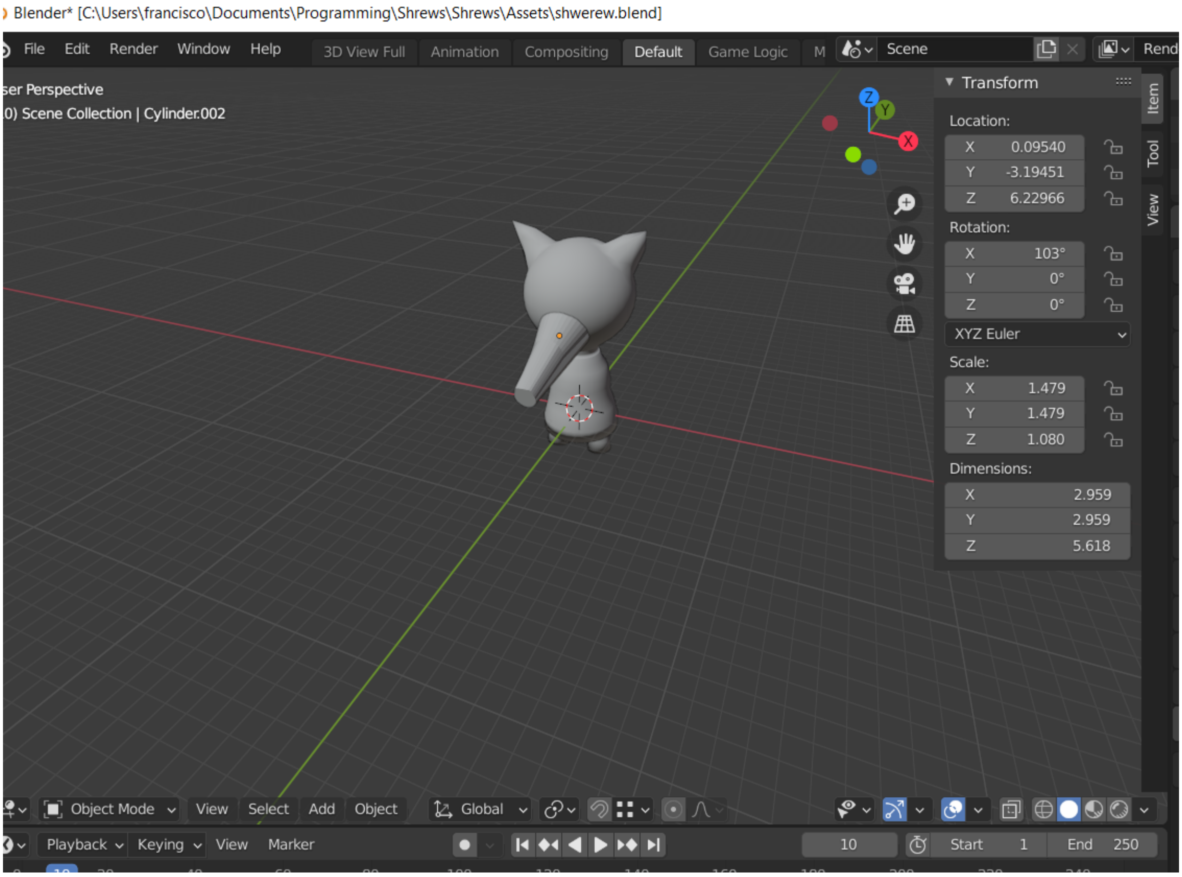


Figure 53: The Shrew model in Blender.

To speed up the projects development, free preset models were obtained from the Unity store as basis for the game objects used in the game. These include, the cubes used for the map and the structures such as bridges and ramps as in Figure 54. With these, free textures were also obtained from the Unity asset store to be applied in the project's game objects to identify which cubes are which types and to add a better look to the preset structures, such as the grass texture in Figure 55.

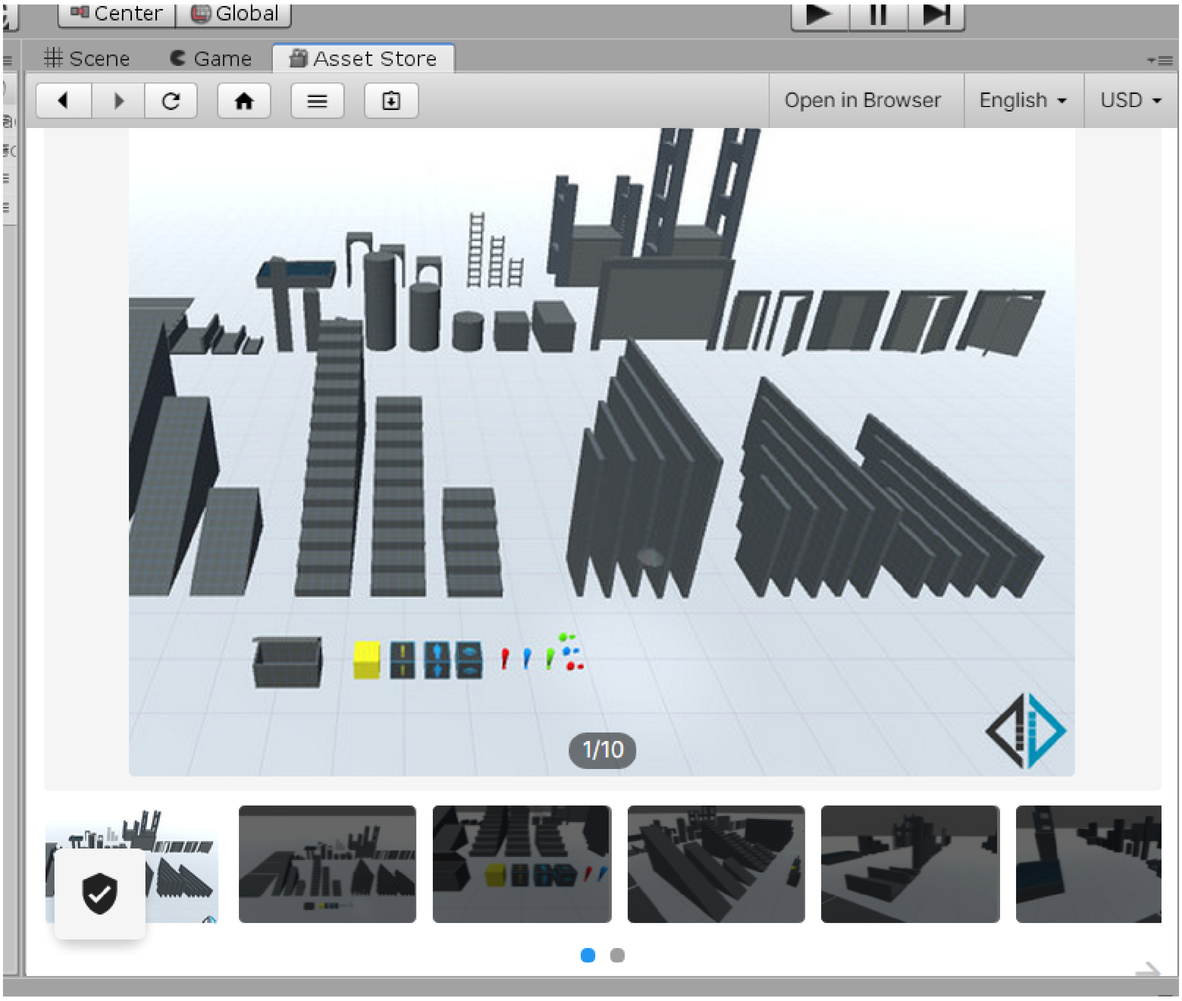


Figure 54: A free 3D models pack available in the Unity asset store to download.

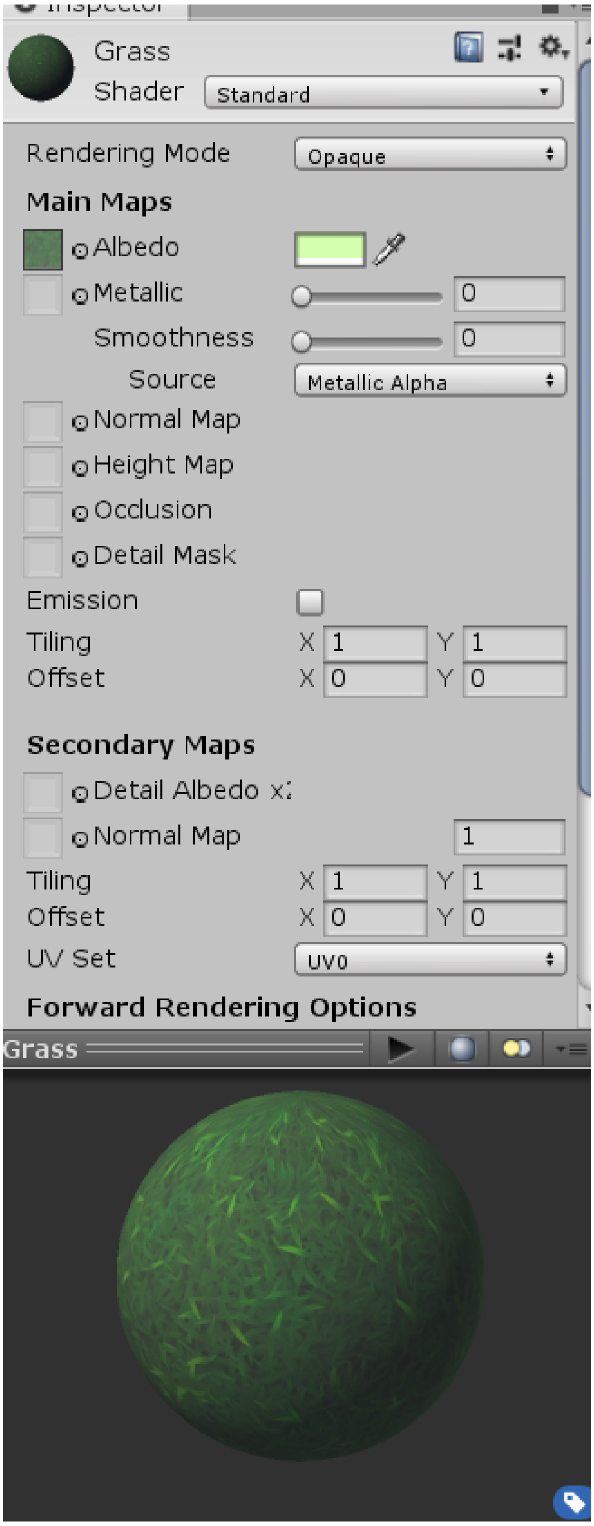


Figure 55: A grass texture also obtained from the asset store, with these texture we can overlay them on top of 3d objects to change their appearance.

As the main shape of Shrews is the cube, used to detail a block and a group of cubes a map, it's important to highlight the separation of the areas to the player. In initial prototyping some basic textures were used for the grass, water and rock imagery, but when applied left no borders to visually identify the number of blocks, limiting space awareness. From initial testing and prototyping imported textures from the asset store were used and applied directly into the game objects. Looking at Figure 56, the map is composed of various cubes, but due to the textures, it looks as if it's a continuous plane, not transmitting the sense of space awareness and cubes.

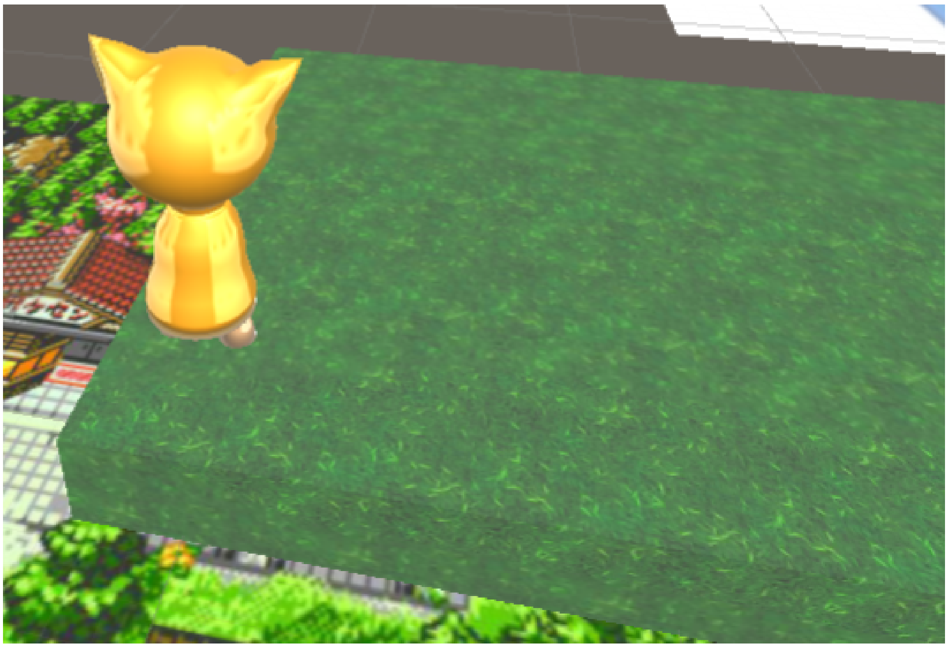


Figure 56: The initial look of the map looked bland and did not give the impression of the amount of cubes in a map.

From team feedback, this approach showed confusion and was recommended to separate each individual cube visually. To circumvent this, new free to use textures were researched to use to change the visuals again (see Fig 57). To help with image editing the software Gimp was used. Gimp is an open source high image manipulation program that allows for great image manipulation. Gimp was also used for almost every image editing associated with the project.

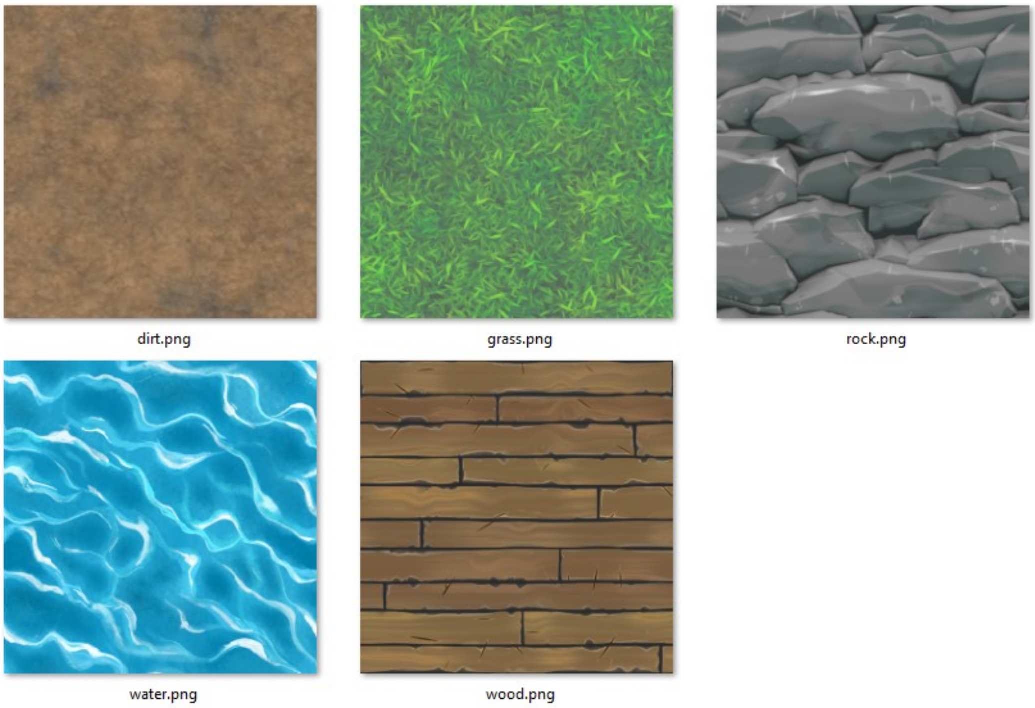


Figure 57: The updated textures used in the game.

Through Gimp, each texture was loaded and edited to have borders around them, seen by Figure 58. Adding borders to textures, once imported to Unity and applied to 3d shaped objects with faces, the image is applied on every side of the object. This results in a visual distribution of the various objects that make up the map (see Fig 59) and buildings built by the player, giving a better representation of the puzzles and moves possible to solving the puzzle.

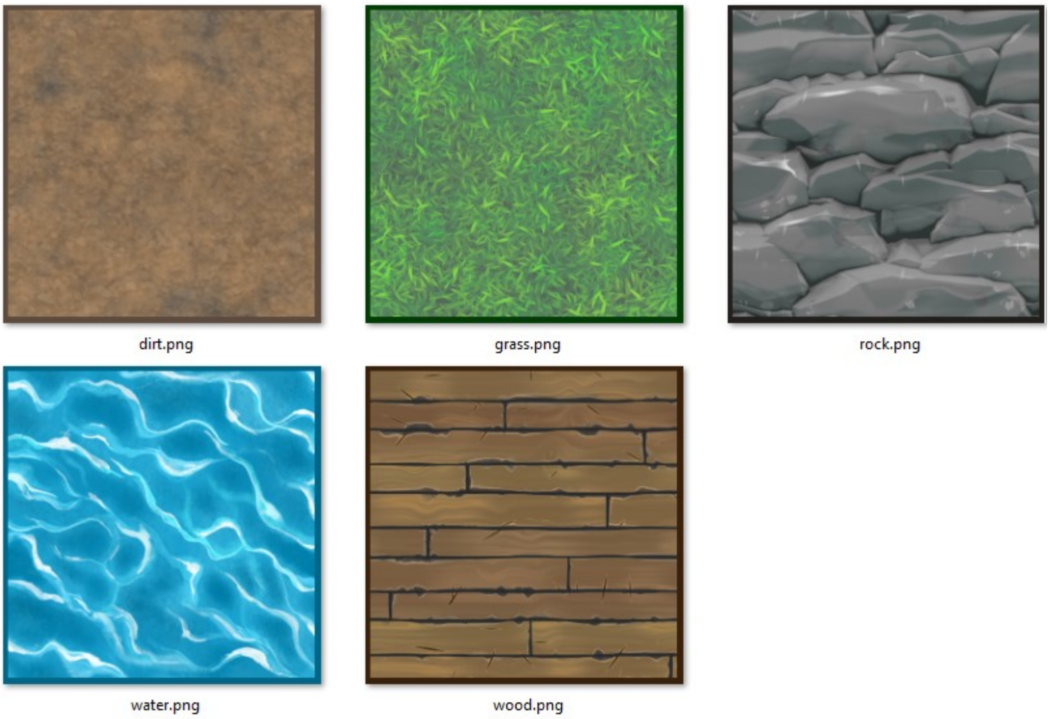


Figure 58: The updated textures used in the game, with borders added to them.

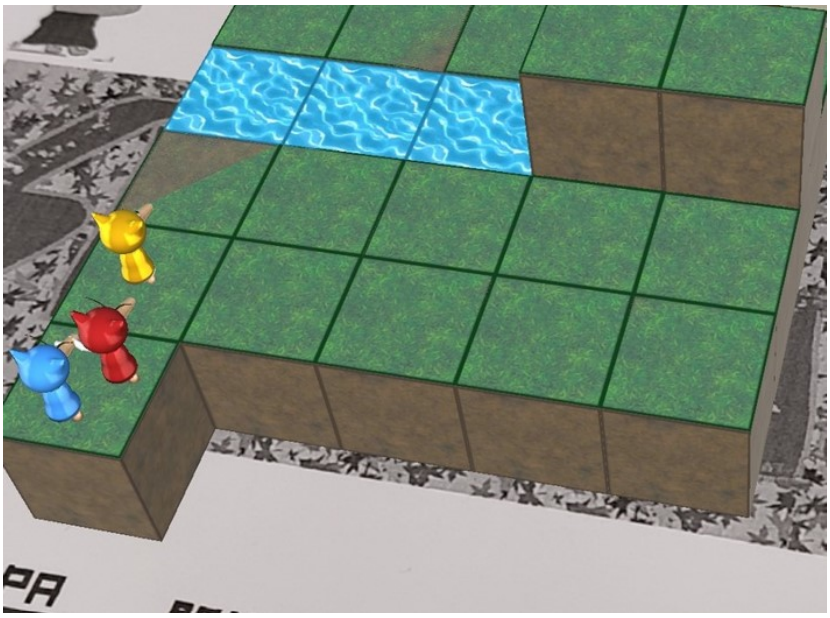


Figure 59: Adding borders to the textures and applying them to the cube game objects transmits the information on the number of cubes that exist.

The only cube type that has a different approach is the dirt cube. To give the look of dirt and grass on top, the cube was modified within Unity to have a quad game object put on top of it (see Fig 60), with a grass texture applied, giving it the distinct look it has from the rest.

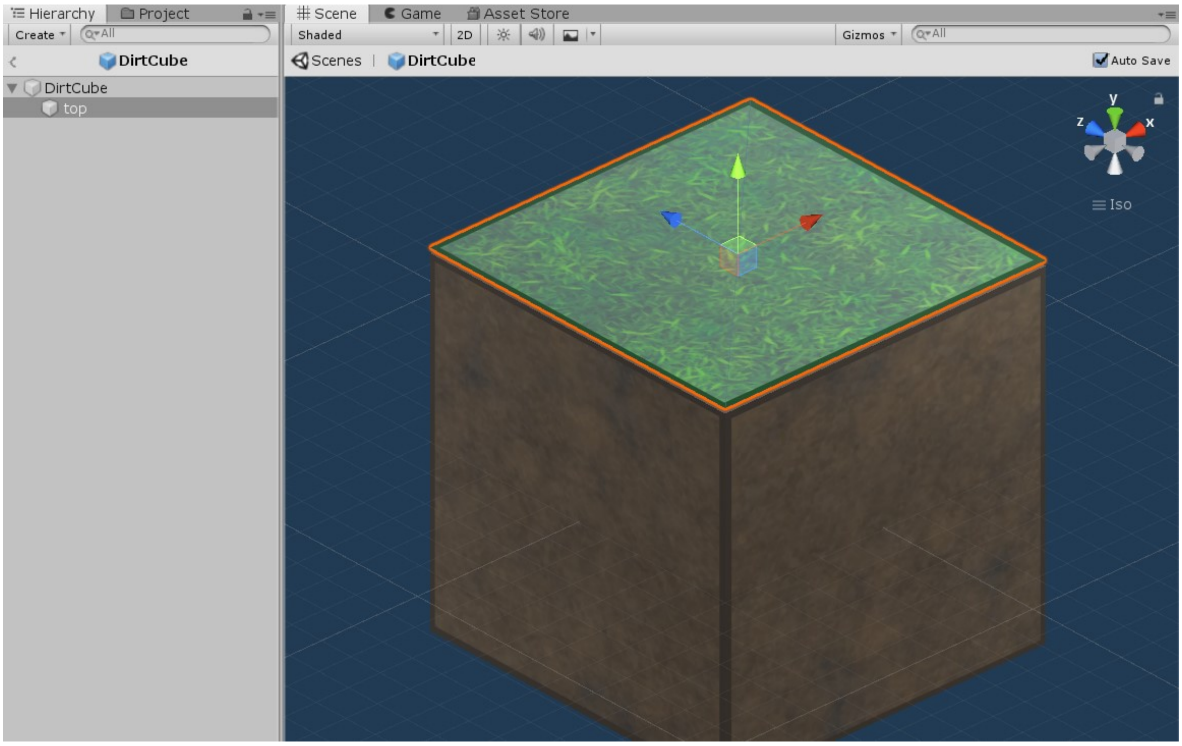


Figure 60: The dirt cube with a quad game object shape layered on top of it, with a grass texture.

5.4.2 *Audio*

As visual information is important so is audio in a Video Game. Audio can be used to alert the player to interactions, such as a correct input or a wrong input and help in the immersion and emotion. For Shrews, various free royalty sound effects and music were collected to be used in the project (see Fig 61).

Figure 61: The list of all the sounds and effects in the game.

The music and sounds used in the game include:

- The menu music
- The puzzle stages music
- After reaching the exit, another more upbeat music starts playing
- Moving has a sound effect
- Invalid movement has a sound effect
- Building has a sound effect
- Trying to build on an invalid cube has a sound effect
- Completing a puzzle has a sound effect

5.5 MARKERS

AR Markers are images used to serve as an identifier for targetting where to present Augmented Reality in the real world. They can be used to display virtual objects or register an input. These are directly related with the technology Vuforia in this project as it is the technology that identifies and renders the processing of the Augmented Reality in the project.

Since Vuforia rates the quality of the image based on its characteristics and uniqueness of patterns, a couple of images were chosen in early prototype testing with unique identifiers for the game to help the camera easily and more reliably identify the Augmented Reality markers to track. Initially prototype markers were used for testing purposes, identified in Figure 68, Figure 69 and Figure 70, being very specific and detailed images it was easy for use and have Vuforia detect them due to their unique patterns and features.

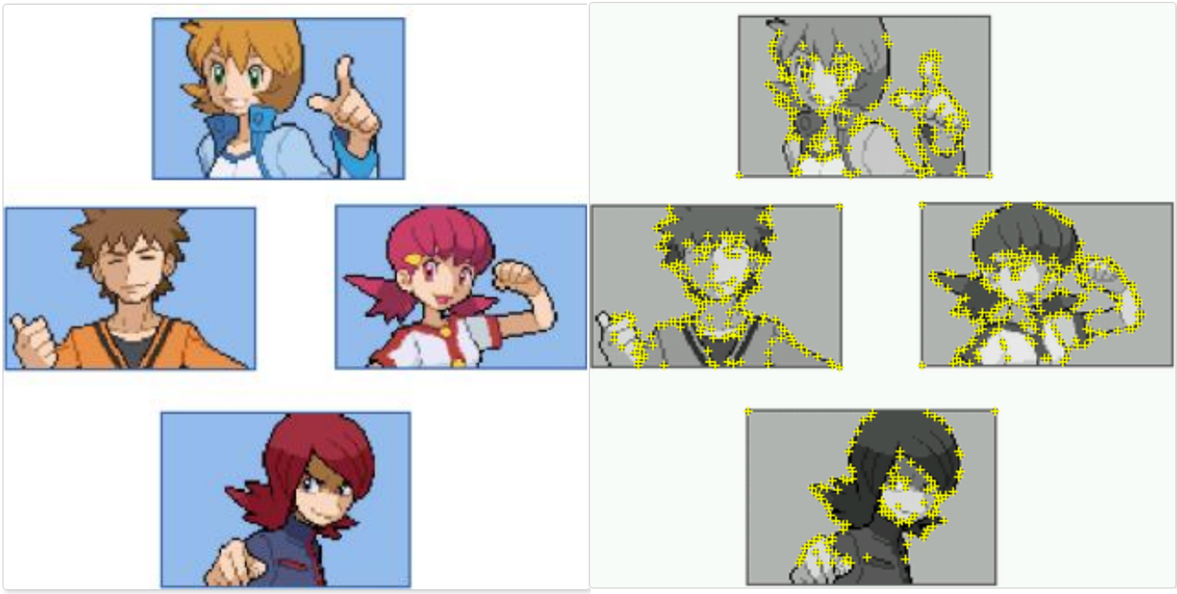


Figure 62: The prototype arrows target image overlaying the directional arrows of the movement.

Figure 63: The prototype tasks target image overlaying the structures and actions of the shrew.

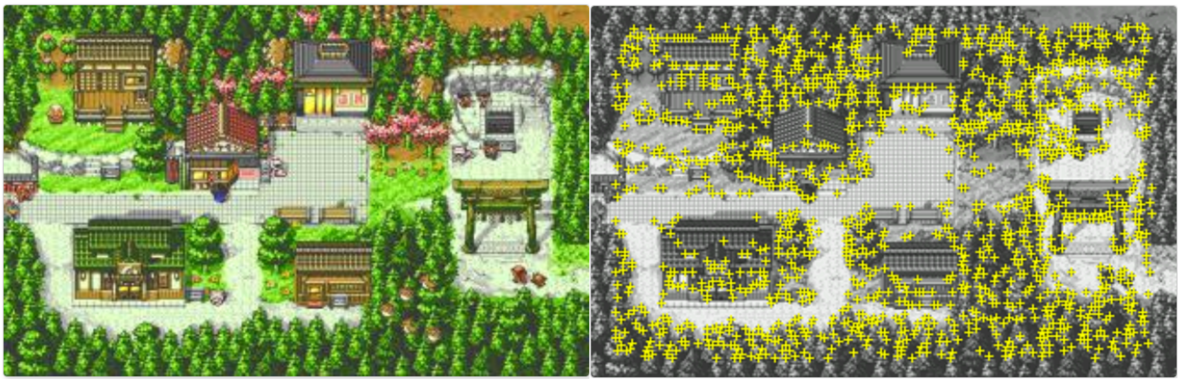


Figure 64: The prototype map target image overlaying the game map.

These prototypes markers proved reliable and worked very well for detection for displaying and detection for identifying the virtual buttons presses for the game in the testing phase. Over the development of the project it was decided to use more intuitive and specific markers with more visual cues than the prototype images as they were not very telling of their functions on a first impression. Designs of more simple and less cost for printing images were created in Figure 65 to serve as the definitive images, but running them through Vuforia detected them as a not very reliable image target with very low features as seen in Figure 65.

To circumvent this problem we had to go back to the images and distinctively add more features to them as seen in Figure 66, so with the simple change of adding more spacing in the shapes of the images, Vuforia was able to pick up more features giving it a better rating for picking up with the camera.

Figure 65: The initial design for the action markers did not generate enough features for good detection.

Figure 66: The initial design for the action markers after being redesigned for better detection with good features.

Even still further tests with the markers ended up with mixed results, as even though the markers were very well featured, the picking up of interaction on the virtual buttons was not working as intended. Various external factors such as the paper quality and light in the environment interfered with the buttons causing them to not register easily, or simply activating by themselves due to the light interfering with the black background of the markers.

Therefore another revision was needed, by returning to the Vuforia guide lines on creating good image targets. It was identified that having patterns behind images was a good idea to have it properly recognize when light comes into contact with it. The redesign of the markers came in the form of adding a patterned background with leaves as shown in fig 67. This further improved the quality of the marker and stability.

Figure 67: The addition of patterned backgrounds improved the features and allowed for better detection over conditions with poor light.

In the end it came to three different markers to be used in this project>

- "Movement marker" (Figure 68): The image target used for the movement in the game, with the images signifying a direction to move in the game.
- "Actions marker" (Figure 69): The image target used for the player actions in the game.
- "Map marker" (Figure 70): The image target used for showing the map of the game.

Figure 68: The marker for the movement buttons.

Figure 69: The marker for the actions in the game, denoted by the name of the action on top.

Figure 70: The marker used for the map of the game represented by the logo created for the game.

On latter testing stages with the markers, positioning and ergonomics of playing the game came in question, as to how should the markers be interpreted by someone new. The direction marker left doubt as to the placement of it and to identify instantly what direction the arrows were pointing caused confusion. Would it be confusing to give players to print a marker sheet with three different markers that would need to be assembled and separated complicated? As a way of simplifying the combination of markers a game board approach was used. Instead of printing the three markers individually, one simple image with all the markers aligned in the correct order was conceived, therefore creating the Shrews game board in Figure 71. A full printable version of the game board can be found in the appendix section A.

The creation of the game board proved more positive as users instantly and easily recognized how to play the game with just a few instructions on how to play, leaving no confusion to the movement arrows.

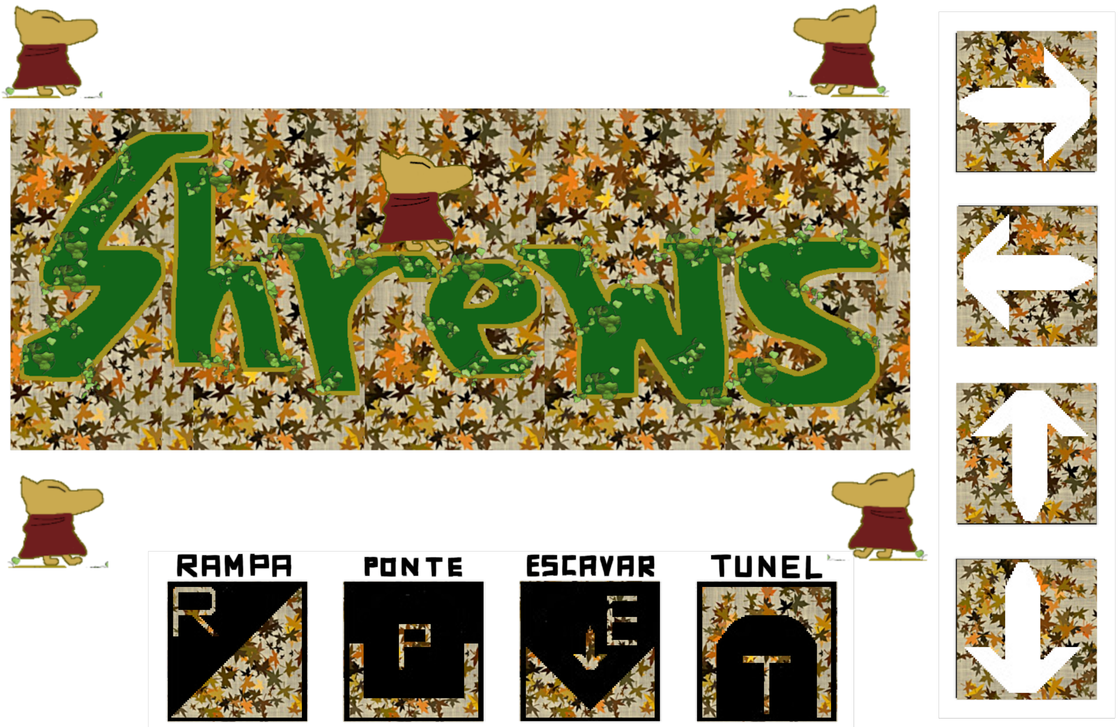


Figure 71: The three markers together make up the game board with all the correct positioning.

6

SHREWS: ARTEFACT

This chapter, is to show and explain the final result of the Shrews prototype game. An overview of its menu, functions and explanation of the functionalities of the app with some added examples of the levels are detailed. This artefact has been developed in the Portuguese language to test with their native speaking testers, with the possibility of other language options in the future.

6.1 OVERVIEW

On opening the game the player is greeted with the Unity logo and brief splash screen of the Shrews logo, followed by the main menu of the game (see Fig 72). In it are three options, "Jogar"(Play), "Testar"(Test) for instructions and "Sair"(Exit) for exiting the game.

Figure 72: The main menu of the game.

The "Jogar" option opens up three other buttons (see Fig 73), each representing one of the possible three game levels. The "Iniciante"(Beginner) button starts the beginner level of the game, the "Médio"(Medium) starts the intermediate level and the "Avançado"(Advanced) begins the advanced level of the game. These stages will be explained in Chapter 6.2.

The second option "Testar" opens another sub menu as well (see Fig 73), that contains the options "Movimento"(Movement) and "Construir"(Build). "Movimento" is the option (see Fig 74) to test the movement buttons and to familiarize the player with the builder's movement within the game and also serves to test the markers and movement logic. The player will learn that moving the Shrew in a direction it is not currently in, makes the character turn around. This is an important detail to be able to efficiently build ramps and other buildings in certain situations.

Figure 73: The sub menus of the game, with the left being the Jogar sub menu and the right being the Testar sub menu.

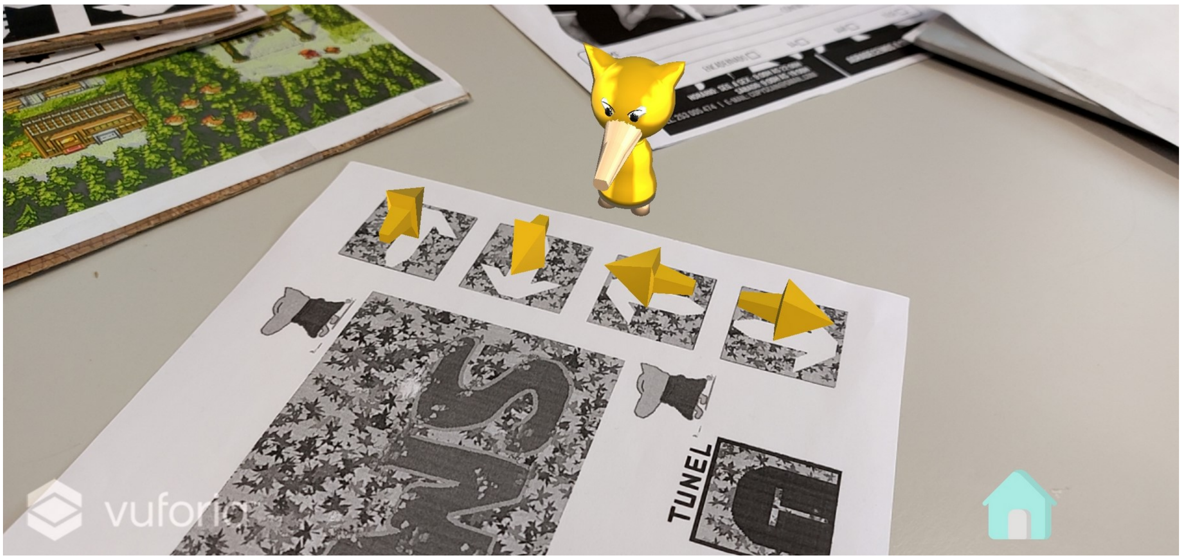


Figure 74: The movement test option shows the player the movement arrows and allows to move the builder shrew to understand the movement.

The "Construir" option (see Fig 75), is to show the player the type of existing cubes and how each action button acts on which type of cube in the game with the various buildings. Showing the player each action and the according cube it has an effect on, transmits the information of which buttons work where, and where the builder Shrew can perform actions.

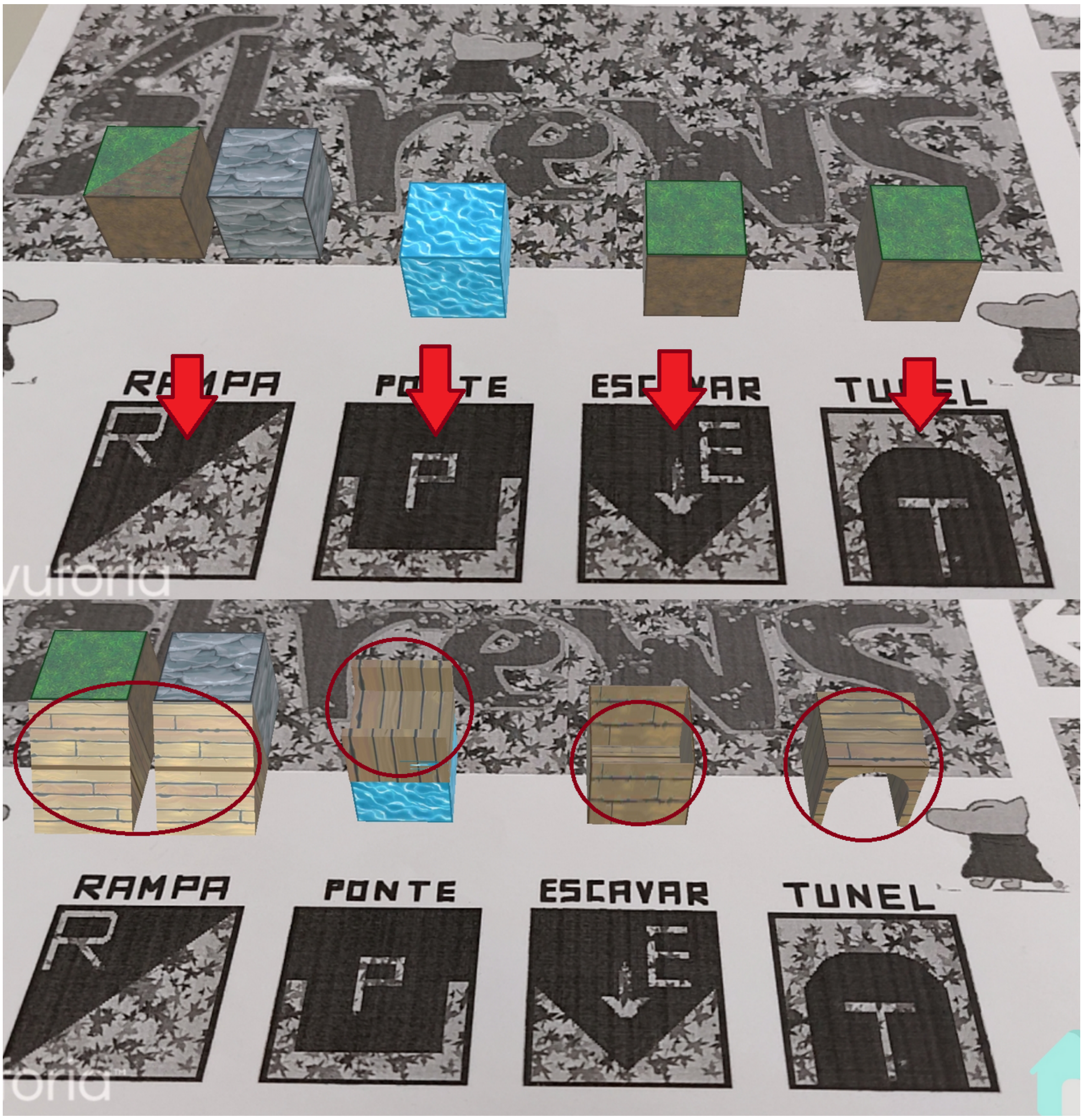


Figure 75: Pressing the virtual buttons will display which buildings interact with which type of cube.

6.2 DEMO

The playable stages open up the device's camera to begin tracking for the markers to display the Augmented Reality and virtual buttons. To begin playing the player must press the play button on the bottom right corner to begin the timer and activate the virtual buttons. At the top left a score is shown and top right a timer. Pausing the game again, will disable the virtual buttons, pause the timer and show an overlay with the Shrews logo and an indication of the game being paused. Also next to the play button is a home button to return to the main menu. A view of the overlay and options are at Figure 32.

6.2.1 *Beginner Level*

The "Iniciante" or beginner level is used as an introductory and experimental level to have the player learn the movement of the Shrew and familiarize with the virtual buttons. The level (see Fig 76) is composed solely of dirt cubes with one in front of the player. The reason for the cube in front of the player is to present an initial obstacle to force the player to move out of the way, or to try and put the knowledge and actions to the test. Making the player think and act over the tools available is intended to begin exercising and testing their Computational Thinking skills over the presented problem.

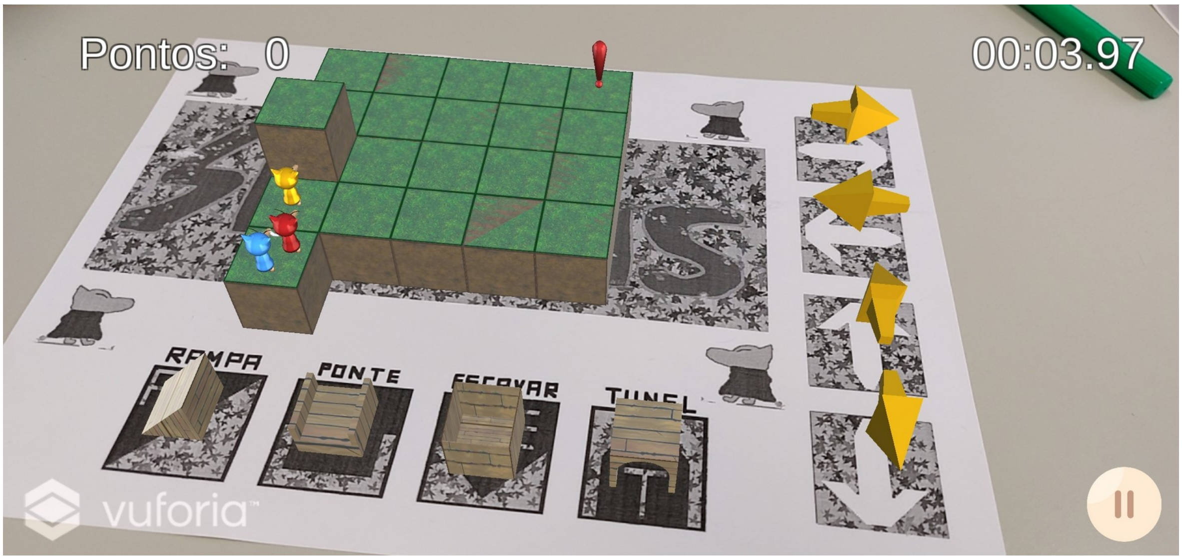


Figure 76: The beginner level of the game.

The scoring system is built simply that, moving the builder gives 1 point, performing an action gives 10 points. While playing the game, moving the builder and performing actions, the changes are directly reflected onto the Augmented Reality as Figure 77 shows. The objective is to touch the red exclamation mark at the end of the stage.

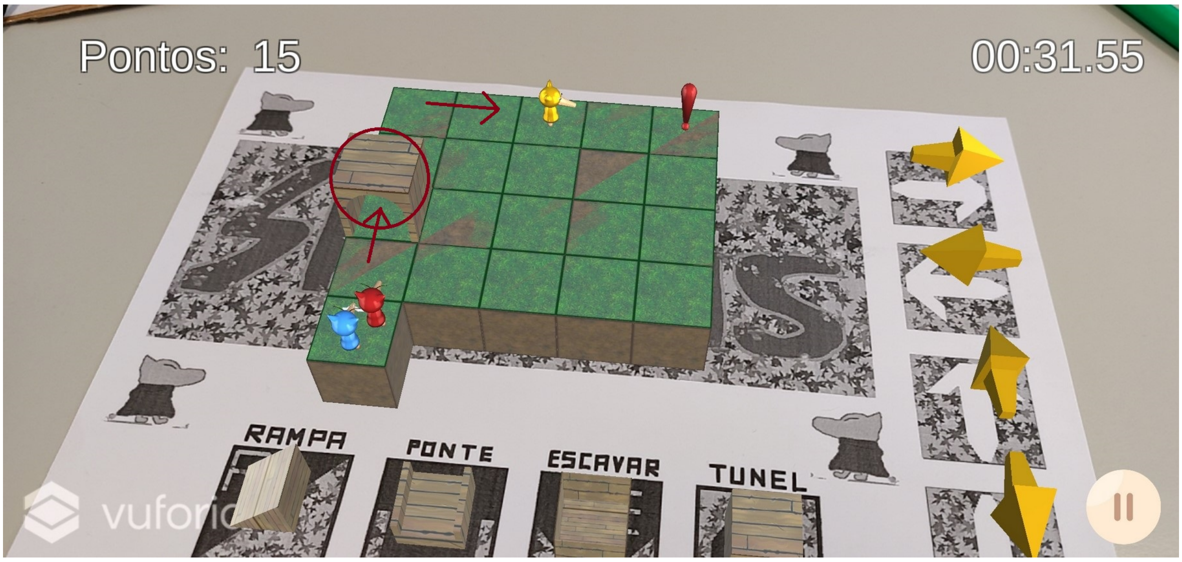


Figure 77: Gameplay of the beginner level.

After reaching the exit sign with the builder, the game will automatically shift into the adult Shrew and the elder Shrew to follow in the footsteps of the builder at Figure 78, therefore laying out an algorithm of the solution to the player.

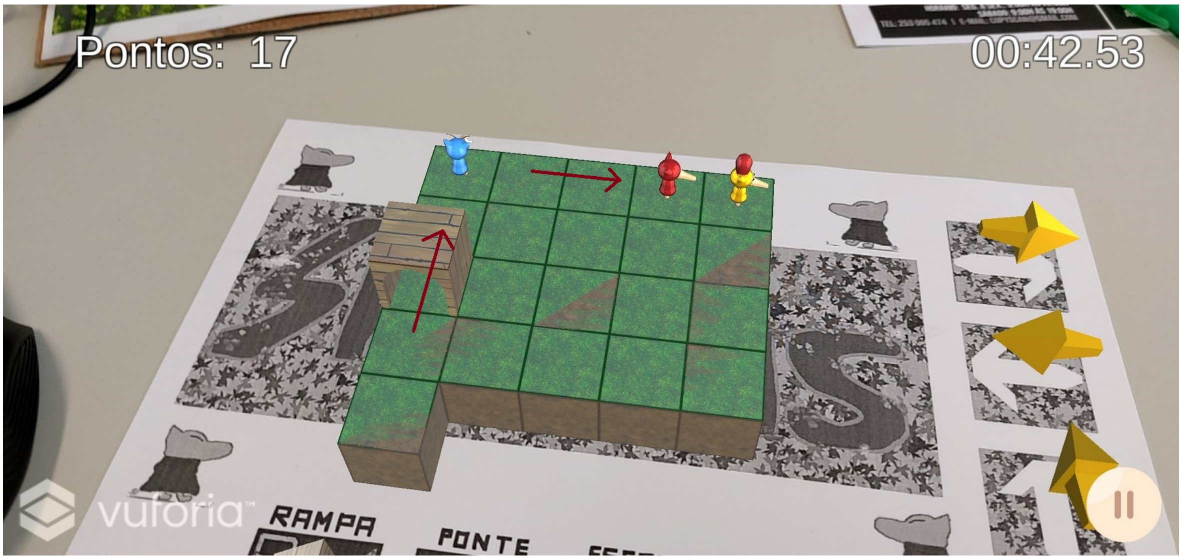


Figure 78: The adult and elder Shrew follow in after the builder.

After completing the stage the player is presented with the results screen at Figure 79, showing the points, the time it took to complete the puzzle and a final score in the mix of the score, the number of steps and the time it took.

Figure 79: The result screen showing the final score.

By playing this level the player is introduced to the game, learns to move the character and interact with blocks giving them a basis of pattern recognition and abstraction by knowing the rules of the game.

6.2.2 *Intermediate Level*

The intermediate level at Figure 80, has the map designed to test the player on the previous knowledge of the previous beginner level. Now with the notion of movement and the action to build over dirt cubes, they can explore further cubes such as the water, and a wall of dirt cubes preventing the player from simply proceeding forward into the exit sign.

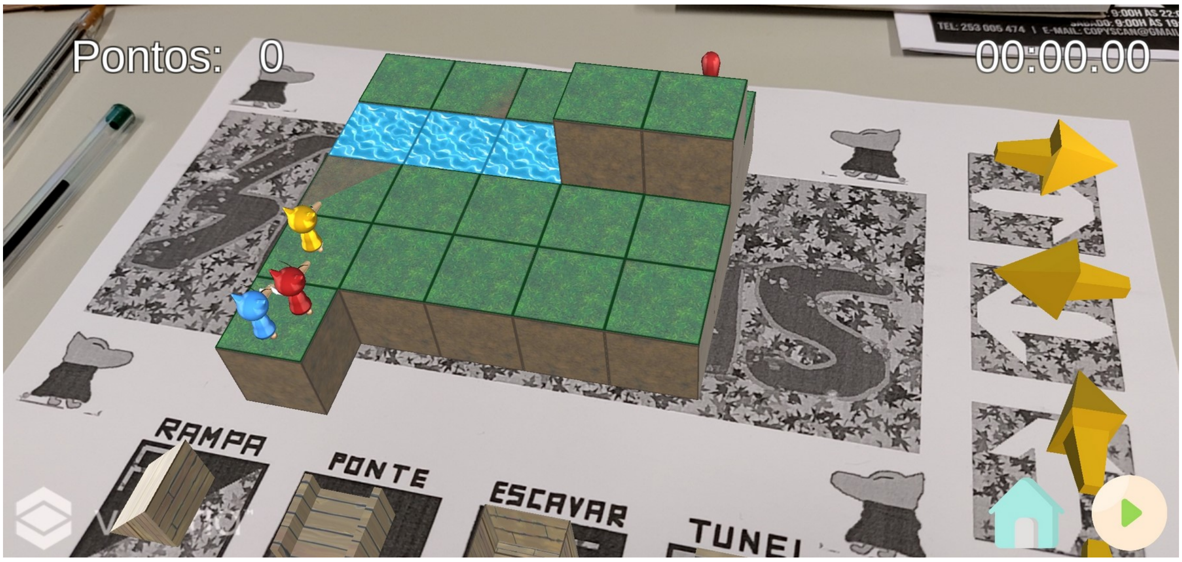


Figure 80: The map of the intermediate level.

To take use of the Augmented Reality technology the player can inspect the map by rotating the board and moving it to take into account the terrain and the cubes behind the wall before beginning to make his move in the game seen in figure 81.

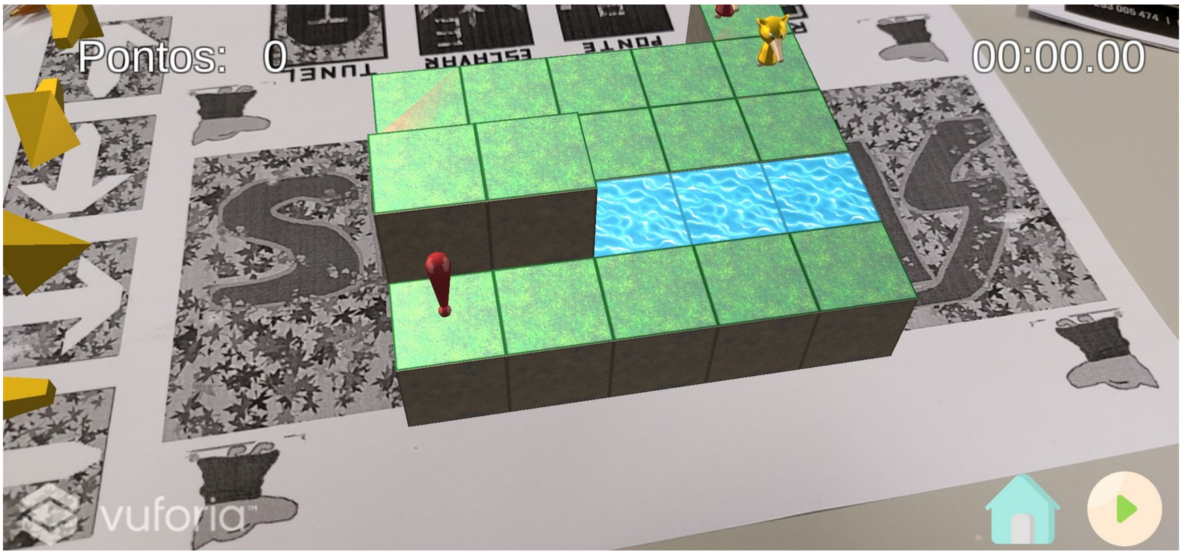


Figure 81: Rotating the board the player can see the map from different perspectives.

This map allows the user to find different paths and use different actions to cross the obstacles. Figure 82 shows the possible paths and actions possible to cross the map and reach the exit. Example 1 shows a path on building a bridge and simply crossing over the river, example 2 shows a path with building ramps to go over the cube wall and example 3 shows a path building a tunnel next to the exit. This way the player can learn and understand how ramps work and are built a tile apart to make space, the tunnelling of dirt cubes to cross and the use of bridges in water to go over water cubes.

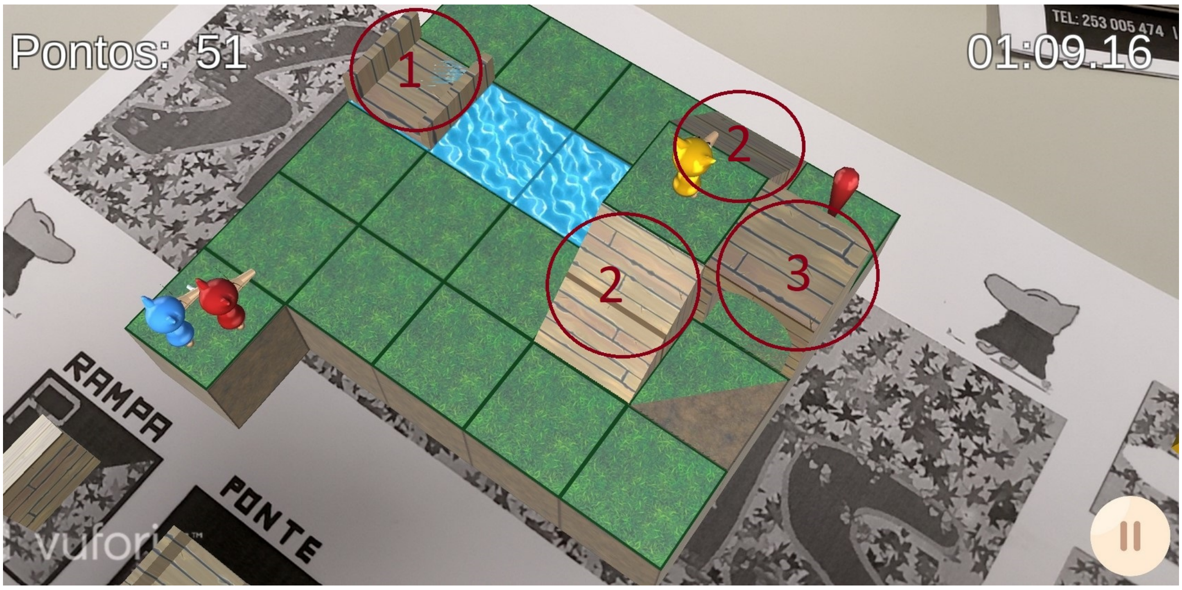


Figure 82: The possible combinations of actions give different paths to completing the puzzle.

6.2.3 *Advanced Level*

The advanced level at figure 83, has the more complex map structure, to test if the player has increased the knowledge and developed strategies to clearing the puzzle easily with the previous experience and tools learned over the course of their play. Being composed of all types of cubes and limiting the mobility of the player to think about the best strategy of moving out and reach the exit.

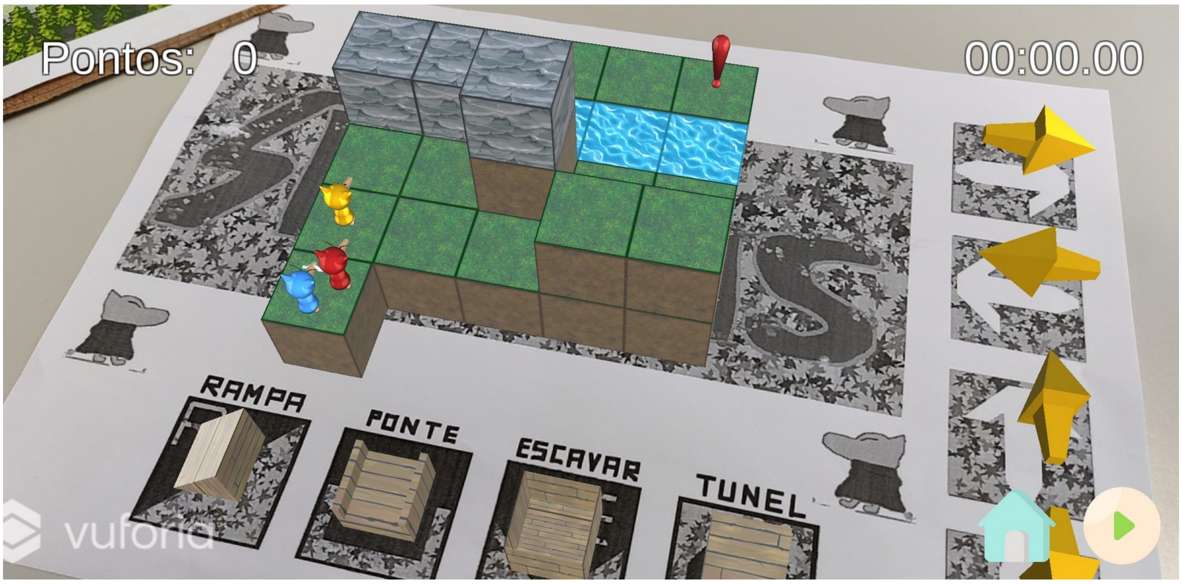


Figure 83: The advanced level of the game.

The layout of the map needs to be examined before hand on the positioning of the various cubes, to determine the plays preemptively, by utilizing the ability to see the entire Augmented Reality and to interact with it physically the user can see the map by moving their game board as they like (see Fig 84). This level has many approaches and requires the use of various actions from the builder to perform and reach the exit sign, with Figure 85 some of the examples to perform. Example 1 shows a path with building ramps over the rock cubes, example 2 shows a path on building a tunnel through the dirt cube and then a bridge to cross the river and example 3 with building ramps over the right wall and a bridge to cross the river. This level puts to the test the Computational Thinking skills acquired from the past levels and information absorbed by the testing and tutorial levels.

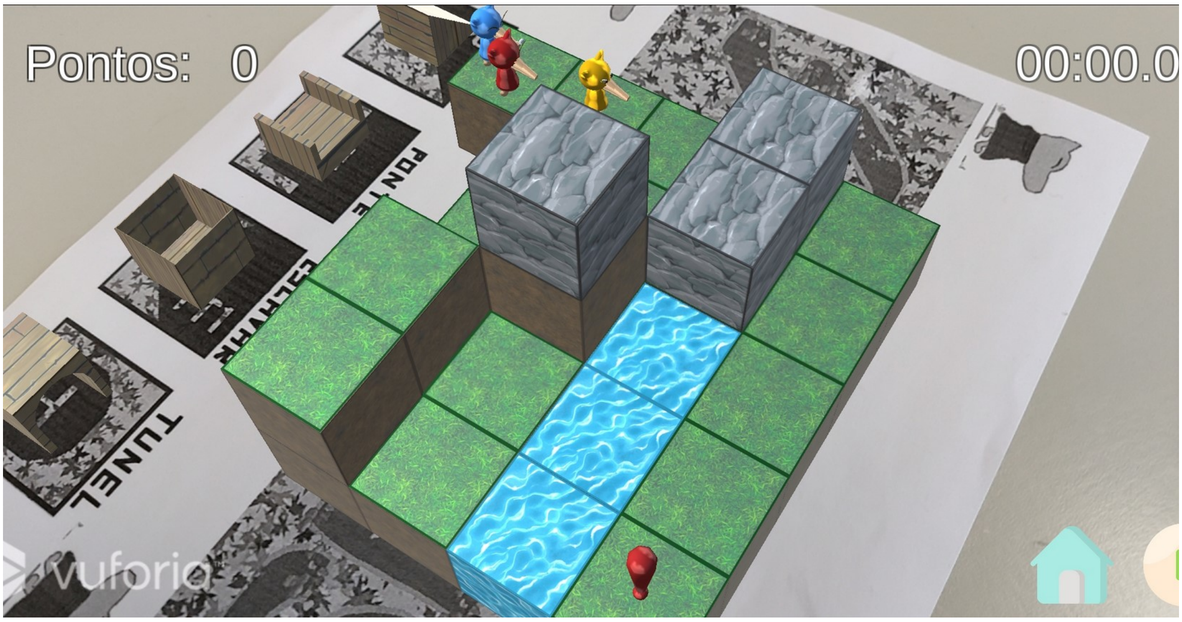


Figure 84: Visualizing the map from another perspective is important to judge the players moves.

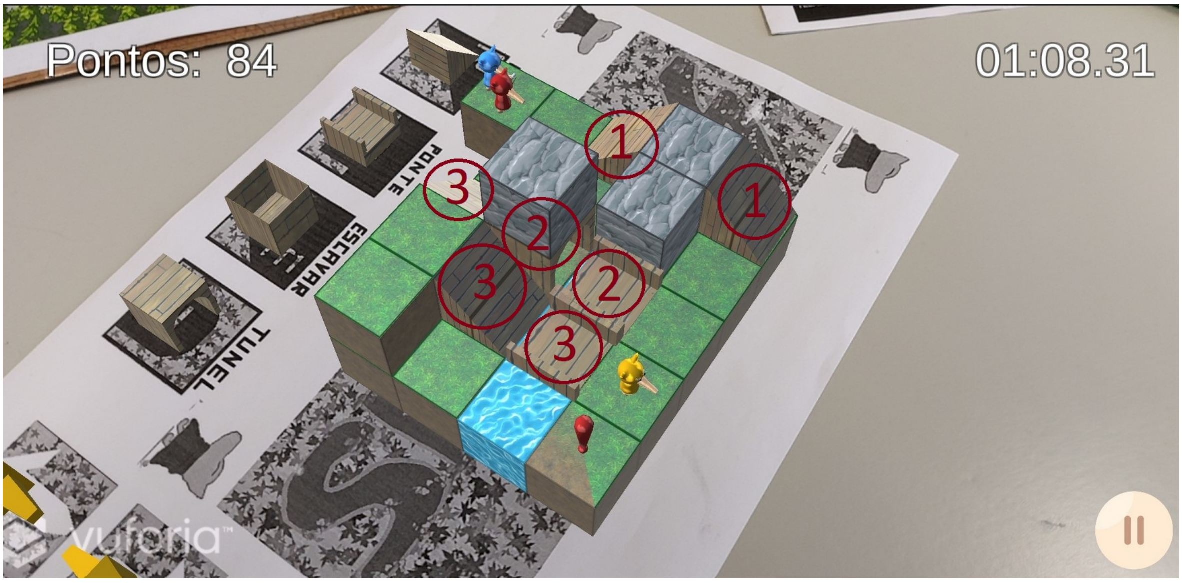


Figure 85: Examples of the various moves that can be performed to complete the puzzle.

7

EXPERIMENT AND DISCUSSION

The objective of this project was to know if it possible to train Computational Thinking skills via playing an Augmented Reality Video Game. For this purpose testing on the game with potential players was needed. The complication of the COVID-19 outbreak hindered the possibility of large scale testing with a younger demographic of children, so other means of testing and collecting feedback were performed. One through a public lecture, another through individual participation.

A lecture (see Fig 86), on Augmented Reality and Computational Thinking was organized to a group of teachers and students at the university of Minho of the degree of Informatics Education, both in person and remotely through video streaming for around a group of 12 people. The purpose of this lecture was to talk about concepts discussed at the earlier chapters of this document at Chapter 3 and Chapter 2.

A lecture on Augmented Reality technology, what it is, how it is shaping the world and the future of it and the concept of CT, the importance and how to shape and practice it were presented to the audience. Following the introduction to the two topics, a presentation of projects that tackled the two topics in the training of Computational Thinking were also introduced and explained, the two projects being Make Your Hero and Shrews. An introduction on the concept of Shrews, the ideas, rules and gameplay were explained and how Computational Thinking concepts were introduced in the game. Followed by this, the prototype game was presented, showing off the menus and options, with the options to test and teach the movement of the builder and the actions of the builder were explained.

A small demo of the beginner level was explained to the audience to familiarize with the game better. Followed the presentation a download QR code was provided that gave everyone in the audience access to the APK (Android Package) file to install on their personal devices and printed copies of a game board were also distributed for the audience to try and play the game for themselves.

During the presentation the audience proved very interested in the use of the game and participated actively to clear all the levels, with the audience trying out various solutions to the puzzles and absorbing well the Computational Thinking skills provided by the game to solve the puzzles. After the presentation and experimentation a survey was also provided to the audience data and feedback collection, which is present in appendix B.

Other than the lecture given to collect feedback, individual testing was also made to test the usability and experience with the game, to assess on various age groups the different aspects and opinions on the ergonomics, way of playing and approach to develop Computational Thinking skills by understanding the game and their Computational Thinking concepts. The second format for collecting data was to create a more general presentation on the topics and present them individually to other participants outside of the lecture. Following a small presentation, an introduction to Shrews was given and the video game and game board given to test on their individual devices, concluding in more data collecting via the survey present in appendix C.

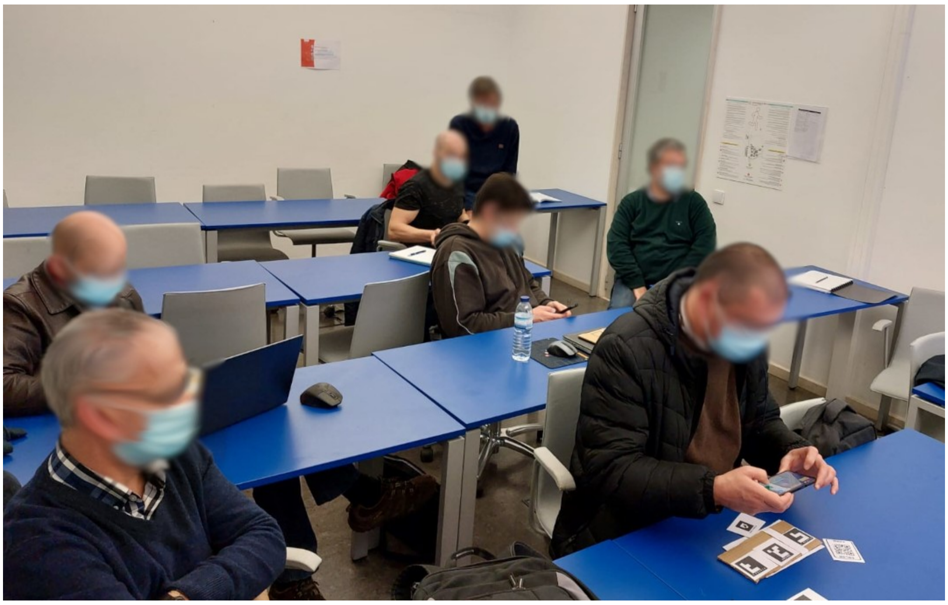


Figure 86: A photo from the audience present at the lecture experimenting with the Augmented Reality projects.

7.1 RESULTS

Following the lecture and individuals who volunteered to test the game, the data was gathered via the surveys mentioned in the previous Chapter. From the pre testing the demographic information was analysed, followed by the post testing questions that analyse the usage and experience with the game. The total amount of surveys answered and therefore population were of 10.

The gender of testers were 60% male and 40% female. Ages comprised from 10% between 9 and 15, 10% between 15 and 18, 10% between 18 and 23 and lastly 70% from 23 and beyond. Most were composed of students in the masters degree for Informatics Education, with another in electrical engineering. Most were still studying with another being finished with their degree. The demographic of the lecture was composed of mostly students in the masters degree for Informatics Education, with some knowledge of CT. For the purpose of information, all the questions with scales were ranked from 1 to 5, 1 being "Disagree Strongly" and 5 "Agree Strongly".

For the concrete questions that were important to the project, the first one, "I understood how the markers/buttons worked" (see Fig 87), the responses were very positive with 80% agreeing strongly (5), 10% agreeing (4) and 10% indifferent (3), concluding it was understood how to use the inputs to work with the game.

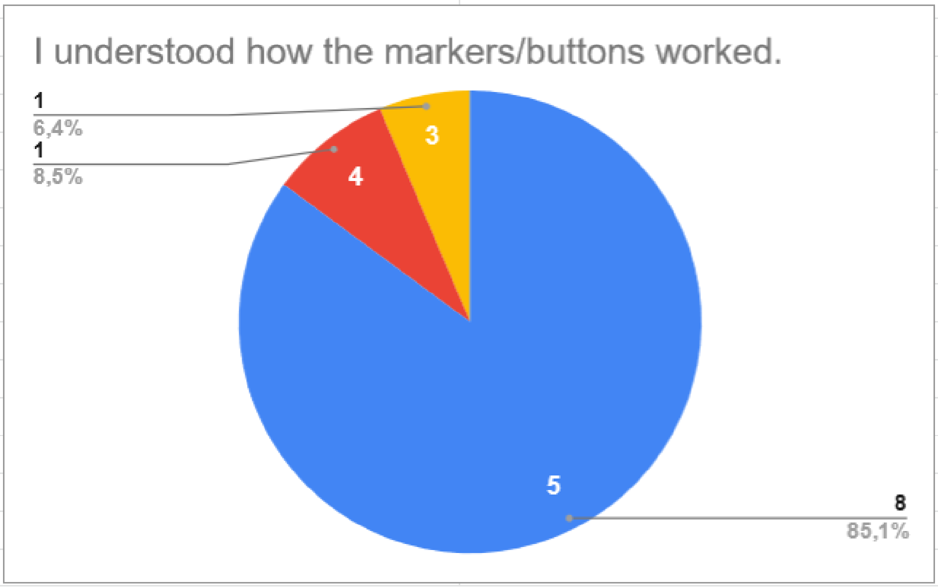


Figure 87: Pie chart showing how people understood how to use the markers or buttons, value in the middle, total outside.

The following question "I understood how to give instructions to the Shrew", 80% agreed strongly and 10% agreed and again 10% indifferent, indicating the understanding of how to build with the Shrew and the buttons was positive.

"I understood how the movement of the Shrew works, in relation to the type of cubes" (see Fig 88), with 90% understanding very well how to move the Shrew in relation to the types of cubes, such as the water cubes and the cubes in front.

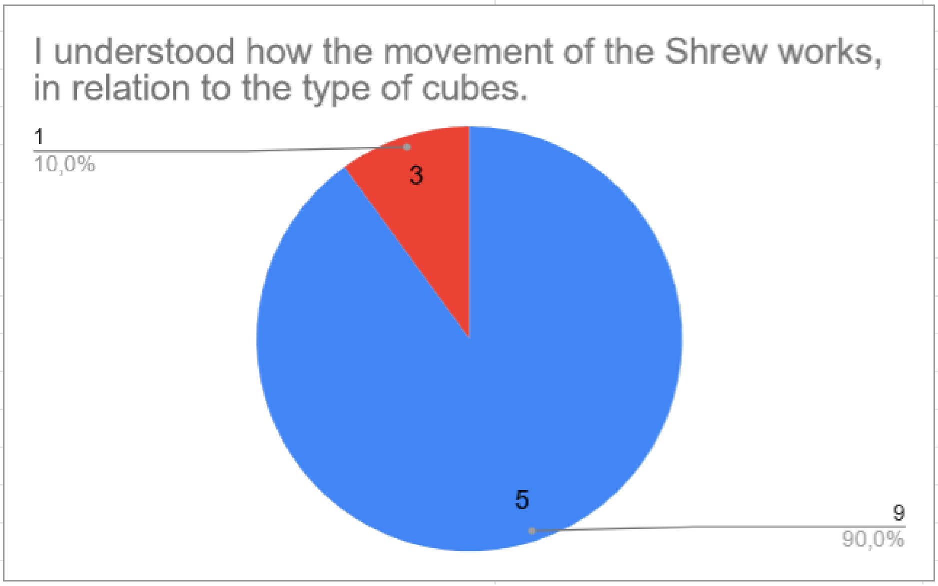


Figure 88: Pie chart showing the players understood how to move the Shrew in relation to the types of cubes.

"I understood in which cubes I can build with the Shrew." also 90% responded with strongly agree showing they understood the different patterns and rules to the different buildings.

No one had difficulty completing any of the levels, being able to find out the puzzle solutions by playing the game. As for the question "I had no difficulty in the usability of the buttons." (see Fig 89), The responses were mixed, with 50% feeling neutral (3), 20% having little trouble (4), 20% having no trouble (5) and 10% having trouble (2). The difficulty was pointed to be on the detection side of the buttons with the Vuforia technology, and not the purpose of using them to play the game, being a technical problem of Vuforia's detection system. The final question "In general, I liked the concept of Shrews" 90% responded with strongly agree and 10% neutrally, reinforcing the potential of the game even in later written feedback. From all the inquiries, the question "I felt I utilized Computational Thinking to solve the levels with success" 100% agreed strongly, reinforcing the objective set out to test.

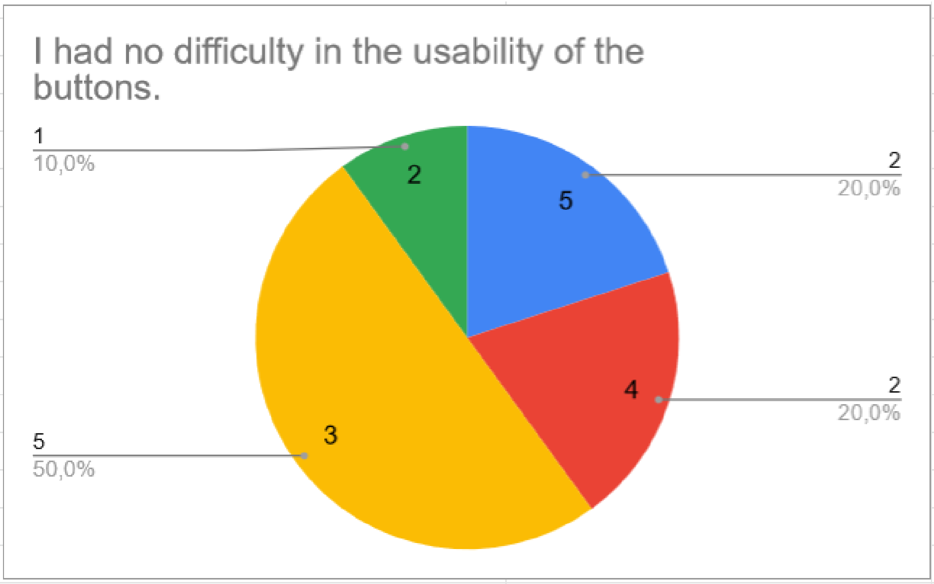


Figure 89: Pie chart showing more mixed results over the difficulty of using the virtual buttons.

From the survey results we can extract that overall the experience and feedback has been very positive from the teaching community following the lecture, with overall positive feedback on the game itself from later testers. The project from gameplay potential, visuals, audio and the concept of Shrews were praised with a very possible tool and game to develop and teach Computational Thinking skills. The only negative feedback seemed to be on the technical aspect of the recognition of the virtual buttons to play the game, with some difficulty to properly recognize the input to move the character and a small confusion on how to start the game when loading the levels, with no indication that it is needed to press the play button to enable the buttons.

7.2 FEEDBACK

Feedback on the game was collected during the lecture and in individual testing cases from people testing the game. From the lecture the feedback on Shrews seemed extremely positive with the people commenting on the presentation being excellent and the visual information being very good and intuitive.

The art direction and design as well as the music and sound effects to display sound information was praised. The usability and ergonomics were positive with the opinion on the buttons being fun and intuitive, but some difficulty on recognizing the pressing of the virtual buttons handled by Vuforia. Suggestions given on the game were about changing the scoring system to perhaps handle times by medals, to penalize the score more heavily on the timer and an implementation of game over with the the algorithm part of the game with the remaining Shrews following, being able to be negatively impacted by the players decisions.

Outside of the lecture the feedback was similarly positive on the same aspects and the concept of various possibilities and combinations of puzzles being possible in the future. The feedback on the game being positive as well as the experiences and the fact of the various aspects of the game being understood and applied proved positively for the objectives set at the start of this project, with the Computational Thinking skills being trained and used with the Video Game. The written feedback can be found in the appendix D.

8

CONCLUSION

Over the course of this document Computational Thinking and it's definitions have been discussed, it's growing importance and interest of study in our modern society as well as the benefits to develop these skills and how Video Games help shape our way of thought. Discussed Augmented Reality, the types of it and it's uses and the growing popularity of the technology in our society and growing impact, as well as the Video Games that utilize it in their core. Combining the two subjects the plan was to develop an Augmented Reality 3d puzzle game that would test players to develop Computational Thinking skills by adding the four main principles into the gameplay, to answer the research hypothesis of this work "It is possible to improve Computational Thinking skills via playing an Augmented Reality puzzle Video Game".

While developing and researching games many projects in parallel have spawned such as RoboTIC (Schez-Sobrino et al., 2020) and ARQuest (Gardeli and Vosinakis, 2019), which while similar in concept are different in gameplay. Projects tackling these concepts together are bound to appear more and more as Augmented Reality and Computational Thinking grow more and more in popularity.

Over the development of this project the objective of developing an Augmented Reality Puzzle Video Game was met. To reach this objective research was done on the two main topics of Computational Thinking and Augmented Reality to get a grasp of their definitions, uses, impacts and how they are shaped. Research of existing Video Games that tackled this topic of education and Computational Thinking, what made Augmented Reality games popular and their correct gameplay aspects as to not just use Augmented Reality to present visual information directly, but to provide interactivity.

Exploring technologies to use and experimenting on them to meet our objectives were a very positive experience. Being able to enrich and learn new tools such as Unity used to create Shrews, with the help of Vuforia to handle the Augmented Reality approach, blending two different technologies into an artefact. The process of game development, from conception, to architecture, experimenting, designing the visual presentation and audio was insightful on the whole process. The challenge of implementing and discussing the inner workings and architecture of the map, and the logic and rules of the movement and shrew actions were very challenging and interesting.

The work done didn't all go perfectly in every regard, as difficulties and changes were bound to happen during the development of the game since the original proposal. A few of the game's logic was not implemented fully due to time constraints when implementing the logic for movement and interactions when building, leading to the algorithm section to be left not working fully as intended. Another difficulty felt was the implementation of the Augmented Reality markers to use in the game.

As explained in Chapter 5.5, the markers evolved greatly over time with changes between them constantly, leaving the task of producing them and also work with Vuforia's technical limitations and difficulty in recognizing markers in certain environments and lighting. Regardless, reliable markers and interactions were implemented and overcome. The testing phase of intending of testing the game in a real classroom with real students was also cut short and limited due to the ongoing complications given by the COVID-19 pandemic. Even still with difficulties and obstacles along the way, the main objectives of this study were met.

With the studies and tests made to understand and collect date and feedback, the overall opinion given is very positive. Out of all the participants, the general opinion is that the Shrews Video Game is a very good and interesting concept, that the use of Augmented Reality was fun and Computational Thinking skills were nurtured and used for solving the puzzles presented to them. Interactions that were viewed by the team left a lot of room to analyse and learn how people's interactions are different from one another. Many handled the device in different ways, using the intended option of moving around the presented Augmented Reality to think before acting, others were thinking logically over the patterns of the behaviour of the cubes and buildings. The only feedback that left doubt in the project were the virtual buttons implemented. Vuforia's trouble in identifying a user's touch over markers presented a problem that would need to be addressed and worked upon, as well as more information being presented at the game's start, indicating that the user must press the play button first before playing.

The approach to developing the game and the functions and tools to match each of the four main aspects of computational thinking has brought the intended approaches that players were to experiment with. The **abstraction** of visualizing the the puzzle, game and tools available to finish the levels, the **pattern recognition** of each relation of the tools and blocks with each other and how they worked, the **decomposition** of the bigger puzzle in smaller problems of each cube as an obstacle to overcome and lastly the **algorithm** of solving the puzzle and seeing the solution in real time being presented. These interactions were key to give the associations of each aspect in the game, and user interactions to develop these skills.

Concluding, backtracking to the original question presented in this document, "It is possible to improve Computational Thinking skills via playing an Augmented Reality puzzle Video Game", we can conclude that it is indeed possible to train Computational Thinking skills via playing an Augmented Reality Video Game, from the study and research created initially, to the conception of the concepts of Computational Thinking being put and molded in Shrews, to the presentation and feedback gathered from players, we reach such conclusion with this dissertation.

8.1 CONTRIBUTIONS

For the whole duration of this work, the spirit of the team was always extremely positive and engaging. Working with promising and growing concepts and technologies is always a great opportunity to look into the future and feel like you're pioneering and contributing meaningfully to something greater. The chance at game development is always welcome as it is a sector that while it has started getting traction in Portugal, still has room to grow and opportunities to appear. Dealing with students and to tackle in their education and learning habits was also an amazing experience and challenge.

With this a shareable application and game board was created that can be distributed easily and played by printing the appropriate image marker. A lecture on these topics of the dissertation was given to an audience of future professors to give them ideas and a taste of what's to come in the future of classrooms. From this dissertation another contribution in the form of a paper (Saraiva et al., 2021), was created and submitted at the 2nd International Computer Programming Education Conference (ICPEC) as a short paper and accepted with good feedback from the conference and scientific community as a good and promising project for the purposes of education and technology. And lastly from this short paper, came also a citation (Ong and Nordin, 2021) for an article on the study of the growing impact and popularity of using AR and other technologies in learning tools.

8.2 FUTURE WORK

As the conclusion and contributions are finished, this section presents possible concepts, ideas or features to the game as a more polished project in the future. First the presentation and polishing of the visuals of the game, including models animations, real time movement, more high definition models and renewed and polished visuals for identity. Second, the addition of more levels with higher difficulty to challenge the player and put the game to higher testing and map imagination. Third, custom parameters for more tailored difficulty, as setting the amount of shrews or more arbitrary obstacles and properties to the movement of the Shrews.

From feedback gathered a few implementations were also suggested such as, leader board scoring based on medals and time finished, higher penalization based on time and the implementation of the game over feature with the logic attached to the blind shrew characters such as to execute the algorithm section of the game in a more interactive way and fun way. The implementation of the game over feature and the real time algorithm, with the logic being attached attached to the remaining shrews was always an initial idea but was cut and not implemented as intended due to time restraints.

Future implementations and topics to explore:

- **Multiplayer**: An implementation of adding multiplayer or cooperative play in which two or more persons work together to solve a puzzle and maybe compare and explore computational approaches of each player.
- **Puzzle generation**: Being able to generate random levels to give a fresh and new level to complete to give more replay value to the game.
- **Profile matching**: Being able to tailor puzzles and difficulty based on a player profile. If the player is more proficient and skilled, the game should limit or give handicaps as to provide a higher challenge to solve the puzzles.
- **Player perspective**: In the sense of perspective of a third person, which is what Shrews has, but an implementation in the first person view would give yet another way of playing the game, by narrowing the players vision to the immediate surroundings, making preemptive play harder.
- **Voice recognition**: In the sense of accessibility, having voice command inputs to play the game would give players with limited motor skills a way to play the game simply by using their voice.

BIBLIOGRAPHY

Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair Macintyre. Recent advances in augmented reality. ieee comput graphics appl. *Computer Graphics and Applications, IEEE*, 21:34 – 47, 12 2001. doi: 10.1109/38.963459.

Ronald T. Azuma. A survey of augmented reality. 6(4):355–385, August 1997. ISSN 1054-7460. doi: 10.1162/pres.1997.6.4.355. URL https://doi.org/10.1162/pres.1997.6.4.355.

Jacob Bell. Finally - 4 real-world uses for augmented reality, May 2014. URL https://www.theprimacy.com/blog/finally-4-real-world-uses-for-augmented-reality/.

Donna R. Berryman. Augmented reality: A review. *Medical Reference Services Quarterly*, 31 (2):212–218, 2012. doi: 10.1080/02763869.2012.670604. URL https://doi.org/10.1080/02763869.2012.670604. PMID: 22559183.

Debra Borkovich and Philip Noah. Big data in the information age: Exploring the intellectual foundation of communication theory. 11 2013.

Matt Bower, Cathie Howe, Nerida McCredie, Austin Robinson, and David Grover. Augmented reality in education – cases, places and potentials. *Educational Media International*, 51(1):1–15, 2014. doi: 10.1080/09523987.2014.889400. URL https://doi.org/10.1080/09523987.2014.889400.

Hamiyet Bursali and Rabia Meryem Yilmaz. Effect of augmented reality applications on secondary school students' reading comprehension and learning permanency. *Computers in Human Behavior*, 95:126 – 135, 2019. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2019.01.035. URL http://www.sciencedirect.com/science/article/pii/S0747563219300445.

Chris Candy. The history of augmented reality, October 2013. URL https://www.citationmachine.net/apa/cite-a-website/edit.

Anna Gardeli and Spyros Vosinakis. Arquest: A tangible augmented reality approach to developing computational thinking skills. 05 2019. doi: 10.1109/VS-Games.2019.8864603.

James Gee. Learning by design: Good video games as learning machines. *E-learning*, 2, 03 2005. doi: 10.2304/elea.2005.2.1.5.

Lori Grata and Katharine Roche. A review of the importance of computational thinking in k-12. *eLmL 2018, The Tenth International Conference on Mobile, Hybrid, and On-line Learning*, pages 32 –34, 10 2020. ISSN 2308-4367.

Tomasz Grzegorczyk, Rafał Śliwiński, and Joanna Kaczmarek. Attractiveness of augmented reality to consumers. *Technology Analysis and Strategic Management*, 04 2019. doi: 10.1080/09537325.2019.1603368.

Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004. ISSN 02767783. URL http://www.jstor.org/stable/25148625.

Matvii Kovtun. Webar web augmented reality (web ar), Nov 2020. URL https://perfectial.com/blog/webar/.

Hao Lei, Ming Ming Chiu, Feng Li, Xi Wang, and Ya jing Geng. Computational thinking and academic achievement: A meta-analysis among students. *Children and Youth Services Review*, 118:105439, 2020. ISSN 0190-7409. doi: https://doi.org/10.1016/j.childyouth.2020.105439. URL http://www.sciencedirect.com/science/article/pii/S0190740920311725.

Christopher McFadden. 9 powerful real-world applications of augmented reality (ar) today, Jun 2020. URL https://interestingengineering.com/9-powerful-real-world-applications-of-augmented-reality-ar-today.

Brett Molina. 'pokémon go' sets download record on apple's app store, Jul 2016. URL https://eu.usatoday.com/story/tech/gaming/2016/07/22/pokmon-go-sets-download-record-apples-app-store/87446714/.

Hilarie Nickerson, Catharine Brand, and Alexander Repenning. Grounding computational thinking skill acquisition through contextualized instruction. 08 2015. doi: 10.1145/2787622.2787720.

Adam C. Oei and Michael D. Patterson. Playing a puzzle video game with changing requirements improves executive functions. *Computers in Human Behavior*, 37:216 – 228, 2014. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2014.04.046. URL http://www.sciencedirect.com/science/article/pii/S0747563214002672.

Sin Nee Ong and Noradila Nordin. Etr-ar: Augmented reality intervention in a web-based card game of the emerging technology revolution concepts. *Multidisciplinary Applied Research and Innovation*, 2(3):107–111, 2021.

Rauno Pello. Design science research a summary, Nov 2019. URL https://medium.com/@pello/design-science-research-a-summary-bb538a40f669.

Xiuquan Qiao, Pei Ren, Schahram Dustdar, Ling Liu, Huadong Ma, and Chen Junliang. Web ar: A promising future for mobile augmented reality - state of the art, challenges, and insights. *Proceedings of the IEEE*, 107:1–16, 02 2019. doi: 10.1109/JPROC.2019.2895105.

Elay Romanov. Why is augmented reality taking such a long time to gain popularity?, Jul 2020. URL `https://arpost.co/2020/07/20/augmented-reality-long-time-popularity/`.

Francisco Saraiva, Lázaro VO Lima, Cristiana Araújo, Luis Gonzaga Magalhães, and Pedro Rangel Henriques. Shrews: A game with augmented reality for training computational thinking (short paper). In *Second International Computer Programming Education Conference (ICPEC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

Santiago Schez-Sobrino, David Vallejo, Carlos Glez-Morcillo, Miguel Redondo, and Jose Castro-Schez. Robotic: A serious game based on augmented reality for learning programming. *Multimedia Tools and Applications*, 79, 12 2020. doi: 10.1007/s11042-020-09202-z.

H. Tankovska. Global consumer vr/ar/mr market size 2018-2023, Aug 2020. URL `https://www.statista.com/statistics/936078/worldwide-consumer-immersive-technology-market-revenue/`.

S. Teixeira, R. V. Boas, F. Oliveira, C. Araújo, and P. R. Henriques. Ontojogo: An ontology for game classification. In *2020 IEEE 8th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–8, Aug 2020. doi: 10.1109/SeGAH49190.2020.9201876.

David Weintrop, Nathan Holbert, Michael Horn, and Uri Wilensky. Computational thinking in constructionist video games. *International Journal of Game-Based Learning*, 6:1–17, 01 2016. doi: 10.4018/IJGBL.2016010101.

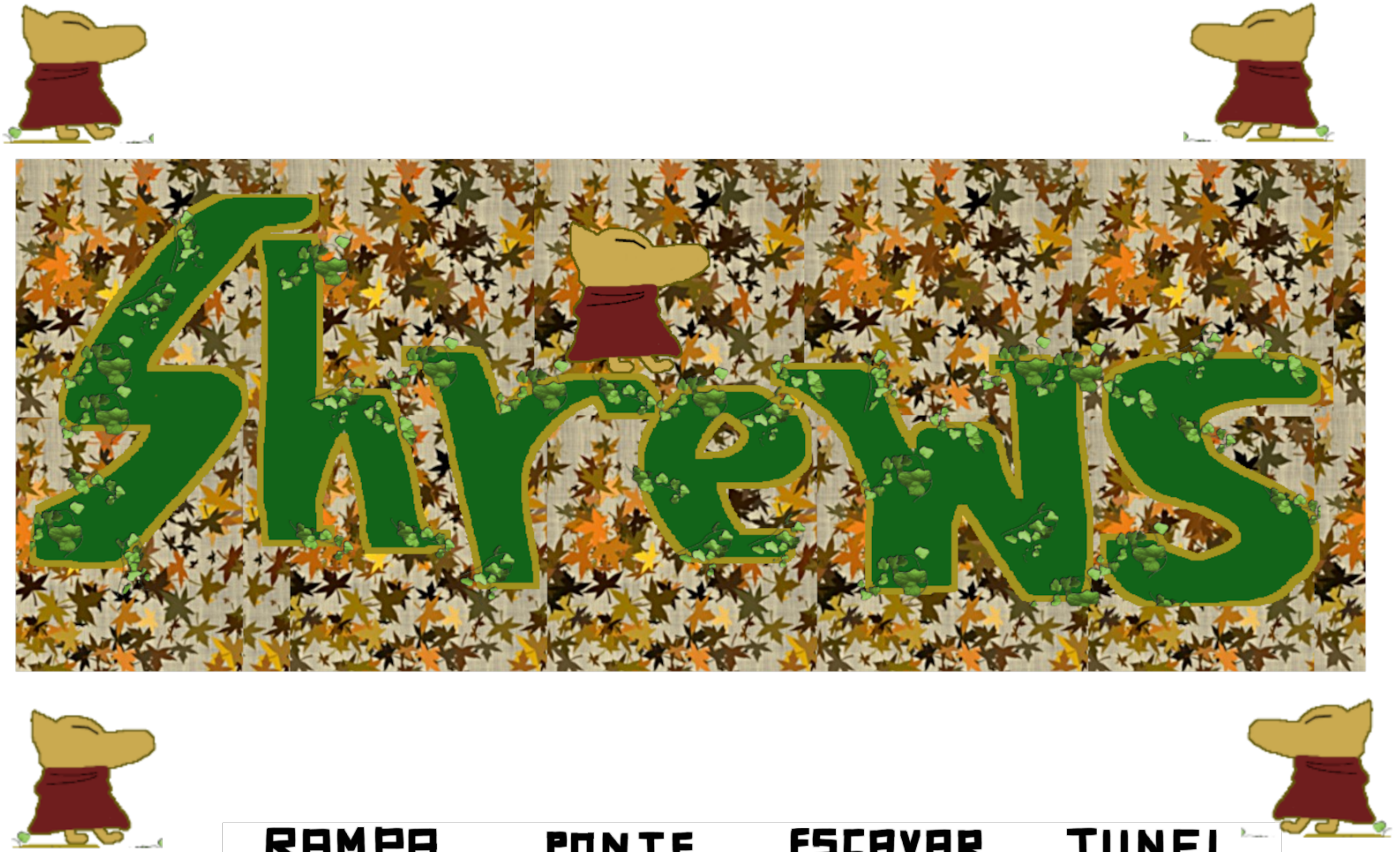
Jeannette M. Wing. Computational thinking benefits society, Jan 2014. URL `http://socialissues.cs.toronto.edu/index.html?p=279.html`.

Weinan Zhao and Valerie Shute. Can playing a video game foster computational thinking skills? *Computers and Education*, 141:103633, 07 2019. doi: 10.1016/j.compedu.2019.103633.

A

GAME BOARD

The following image is a full printable image of the Shrews game board.


Shrews


RAMPA
R
PONTE
P
ESCAVAR
E
TUNEL
T

B

LECTURE SURVEY

The following survey was used in the lecture given on Computational Thinking and Augmented Reality.

Survey

Este formulário irá nos ajudar a enteder quem são, quantos são e como usam o app. Nos ajude a entendender o Impacto da Realidade Aumetentada na construção de ferramentas adequadas para treinar o Pensamento Computacional

*Obrigatório

Scan this

1. Gênero *

Marcar apenas uma oval.

- ( ) Masculino
- ( ) Feminino
- ( ) Prefiro não dizer
- ( ) Outro: ______________________

2. qual a sua idade? *

Marcar apenas uma oval.

- menos de 17 anos
- 18 anos
- 20
- 21
- mais de 22 anos

3. Qual o seu curso? *

4. Ano do curso *

Marcar apenas uma oval.

- 1º ano
- 2º ano
- 3º ano
- 4º ano
- Outro:

Depois de Usar as apps, responda às seguintes questões

Pular para a pergunta 5

Pós-teste Make Your Hero

Uso da Realidade Virtual e Aumentada no Treino do Pensamento Computacional. Este formulário tem como objetivo avaliar o uso das ferramentas adequadas de Realidade Aumentada para treinar o Pensamento Computacional

5. Eu entendi o que é Realidade Aumentada.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	⬭	⬭	⬭	⬭	⬭	Concordo Totalmente

6. Eu prefiro usar ferramentas que usam Realidade Aumentada, em vez do que teclado e rato para aprender.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	⬭	⬭	⬭	⬭	⬭	Concordo Totalmente

7. Achei fácil a atividade. 1 para Não achei nada fácil e 5 para Muito fácil

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	⬭	⬭	⬭	⬭	⬭	Concordo Plenamente

8. Eu achei fácil como criar um personagens.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente						Concordo Plenamente

9. Seria fácil montar qualquer herói pedido.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente						Concordo Plenamente

10. Eu entendi como os cartões funcionam.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente						Concordo Totalmente

11. Eu entendi para que servem os botões.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente						Concordo Totalmente

12. Se eu tiver que aprender a programar no futuro, gostaria de usar este tipo de ferramenta para entender melhor os conceitos.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

13. Eu gostaria de usar este app no meu smartphone.

Marcar apenas uma oval.

	1	2	3	4	5	
Não quero	◯	◯	◯	◯	◯	Muito

14. Eu recomendaria este app aos meus amigos.

Marcar apenas uma oval.

	1	2	3	4	5	
Jamais	◯	◯	◯	◯	◯	Claro

Pós-teste Shrews

Uso da Realidade Virtual e Aumentada no Treino do Pensamento Computacional. Este formulário tem como objetivo avaliar o uso das ferramentas adequadas de Realidade Aumentada para treinar o Pensamento Computacional

15. Percebi como funcionam os marcadores/botões. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

16. Percebi como dar instruções ao Shrew. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

17. Percebi como funciona o movimento do Shrew face aos tipos de cubos. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

18. Percebi em que cubos posso construir com o Shrew. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

19. Tive dificuldade a concluir níveis. *

Marcar apenas uma oval.

- Sim
- Não

20. Se respondeu sim, indique a maior razão.

____

21. Tive dificuldade na usabilidade dos botões.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente						Concordo totalmente

22. Em geral, gostei do conceito do jogo Shrews.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente						Concordo totalmente

Sugestão para melhorar

23. Se gostaria de deixar alguma opinião ou feedback relativo aos projetos, escreva abaixo.

C

GENERAL SURVEY

The following survey was the general survey used to present to testers outside of the lecture.

Shrews questionário

*Obrigatório

1. Gênero *

Marcar apenas uma oval.

- Masculino
- Feminino
- Prefiro não dizer
- Outro: ______________________

2. Qual a sua idade? *

Marcar apenas uma oval.

- 6 a 9
- 9 a 15
- 15 a 18
- 18 a 23
- 23 ou mais

3. Já ouviu falar de Realidade Aumentada? *

Marcar apenas uma oval.

- Sim
- Não

4. Já ouviu falar de Pensamento Computacional? *

Marcar apenas uma oval.

- Sim
- Não

5. Já usou Realidade Aumentada de alguma forma? *

Marcar apenas uma oval.

- Sim
- Não

Pular para a pergunta 6

Pós Teste

Após apresentação e jogar o jogo, responda abaixo.

6. Eu entendi o que é Realidade Aumentada. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente						Concordo Totalmente

7. Eu entendi o que é o Pensamento Computacional. *

Marcar apenas uma oval.

- Sim
- Não

8. Percebi como funcionam os marcadores/botões. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

9. Percebi como dar instruções ao Shrew. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

10. Percebi como funciona o movimento do Shrew face aos tipos de cubos. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

11. Percebi em que cubos posso construir com o Shrew. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

12. Tive dificuldade a concluir níveis. *

Marcar apenas uma oval.

- Sim
- Não

13. Se respondeu sim, indique a maior razão.

____

14. Não tive dificuldade a mexer nos botões virtuais. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente						Concordo totalmente

15. Sinto utilizei o Pensamento Computacional para resolver os níveis com sucesso. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente						Concordo totalmente

16. Prefiro aprender com jogos de Realidade Aumentada face a outros jogos. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

17. Em geral, gostei do conceito do jogo Shrews. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	◯	◯	◯	◯	◯	Concordo totalmente

Sugestões

18. Se gostaria deixar alguma opinião e sugestão escreva abaixo.

D

WRITTEN FEEDBACK

The following feedback was written feedback collected from the surveys.

Feedback 1
Excelente ideia, muito trabalhado já feito, alguns bugs mas resoluveis, os botões às vezes demoram a funcionar.

Feedback 2
Gostei mas foi difícil quando era para clicar nos botões, foi fixe.

Feedback 3
Bom trabalho!

Feedback 4
Foi muito interativo e engraçado de se jogar, mas por vezes era complicado de clicar nos botões.

Feedback 5
A interface é bastante intuitiva, o projeto é bastante interessante. Há infinitas possibilidades para aplicação do projeto na sala de aula. Parabéns!