UNIVERSITY OF THESSALY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DESIGN AND DEVELOPMENT OF A DIGITAL LEARNING GAME UNITY

DIPLOMA THESIS

ATHANASOPOULOS GIORGOS

1809

SEPTEMBER 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΠΤΥΞΗ ΨΗΦΙΑΚΟΥ ΜΑΘΗΣΙΑΚΟΥ ΠΑΙΧΝΙΔΙΟΥ

ΜΕΣΩ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ UNITY

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΘΑΝΑΣΟΠΟΥΛΟΣ ΓΙΩΡΓΟΣ

1809

ΣΕΠΤΕΜΒΡΙΟΣ 2023

Approved from examination committee:

Supervisor: **Tsalapata Hariklia**

Electrical and computer engineer, University of Thessaly.

Member: **Evmorfopoulos Nestor**

Electrical and computer engineer, University of Thessaly.

Member: **Stamoulis Georgios**

Electrical and computer engineer, University of Thessaly.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The 11 points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant

(Signature)

Athanasopoulos Giorgos

16/10/2023

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας αυτής, αποτελούν αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλουν οποιασδήποτε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχουν έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

ΑΘΑΝΑΣΟΠΟΥΛΟΣ ΓΙΩΡΓΟΣ

16/10/223

SUMMARY

This diploma thesis is about the development of a video game made with the game engine called Unity. The game is a two-dimensional (2D) game in which the player controls a character trying to survive from hordes of enemies. In order to survive he has to kill the enemies which drop currencies that he can exchange in the shop for different items at the end of each wave. To win the wave the player has to survive the wave until the time runs out, whereas on wave 20 he wins the round by staying alive until the end of time or killing the strongest monster of the wave if he can (Boss). Finally, to win the overall run the player has to beat 20 waves and by doing that he gains a point which he can use to upgrade himself back in town.

ΠΕΡΙΛΗΨΗ

Η Διπλωματική εργασία αφορά την ανάπτυξη ηλεκτρονικού παιχνιδιού χρησιμοποιώντας το πρόγραμμα σχεδίασης παιχνιδιών Unity. Πρόκειται για ένα παιχνίδι δισδιάστατο όπου ο χρήστης ελέγχει έναν χαρακτήρα και προσπαθεί να επιβιώσει από ορδές αντιπάλων. Για να μπορέσει να επιβιώσει πρέπει να σκοτώνει τους αντιπάλους ώστε να παίρνει νομίσματα τα οποία μπορεί να τα ανταλλάξει με διάφορα αντικείμενα στο κατάστημα στο τέλος κάθε γύρου. Για να νικήσει τον γύρο πρεπει να επιβιώσει μέχρι να τελειώσει ο χρόνος, ενώ στον τελευταίο (γύρος 20) έχει τη επιλογή να νικήσει τον γύρο είτε μένοντας ζωντανός μέχρι να τελειώσει ο χρόνος είτε να νικήσει (αν μπορεί) τον δυνατότερο αντίπαλο (Boss). Τέλος για να κερδίσει συνολικά την προσπάθεια χρειάζεται να νικήσει και τους 20 γύρους κερδίζοντας έτσι και έναν πόντο για να χρησιμοποιήσει για αναβαθμίσεις.

TABLE OF CONTENT

SYMBOLS, KEY WORDS AND ABBREVIATIONS

For better comfort there are some words and abbreviations in the world of electronic games that we will use till the end.

1. RPG: Stands for role-playing game where the player has a character that he moves throughout the game and when that character dies the game usually ends.
2. Boss: It is usually a large enemy stronger than the others which has some special abilities.
3. Attack speed: It is the amount of times someone/something attacks per second.
4. UI: Stands for user interface and it's in general where the interaction between the humans and machines occur.
5. Stat: with stats in the rest of the document we will refer to some characteristics that the player has like health points, damage, movement speed, attack speed and so on.
6. FPS: frames per second. It is the refresh rate of the screen per second. For instance, 60 fps means that the screen will refresh the image of it 60 times in 1 second. The more fps the better and smoother the game runs.
7. Field of a Class: Let's say that we have a class called Students and each student has a name and age, then we say that the class Students has 2 field called name and age.
8. SO: Scriptable Object.
9. HP: Hit points or Health points (it's the same).

1. INTRODUCTION

Everyday life is full of challenges and difficulties that we all face at some point. Some of the common problems that people face in their everyday lives include health crises, workplace issues, friendship issues, financial problems and making the right decisions. The need to help people with problems or taking decisions which, he/she would not regret later on, lead to this particular diploma thesis, making people overcome difficult situations in some point of their lives via a video game.

1.1 The problem

We live in a world where a fast-paced life habit is required in order for a person to survive. This life habit can have several negative effects on the person's health both mentally and physically. Some of these negative effects that impact a person's mind and body according to Dr. Libby are the following:

- Feeling tired but wired or generally exhausted
- Quick to overreact
- Poor short-term memory
- Easily panicked
- Difficulty relaxing
- Bad and hurried decision making
- Poor sleep patterns or waking up just as tired as when you went to bed and many more

It is important to note that these symptoms are common problems of the fast-paced life habit but not normal in any way and need to be dealt with as soon as possible.

1.2 Purpose

The purpose of this project is to develop a game using the game Engine technology called Unity, which helps people to overcome some of the above problems from everyday life routine, helping them boost the quality of life. It is not possible to cover and deal with all the problems but a handful of those.

The game lets the person forget for a short period of time about reality, removing this way the stress and anxiety he developed throughout the day, which also helps him have better sleeping habits, since stress and sleeping habits are connected according to clinical psychologist and sleep medicine expert Dr. Michael Breus. Moreover, the progression of the game is linear, which means that to achieve the game's goal, surviving the enemy hordes, he has to fight his way through from beginning to the end. This process gives the person enough time to think each time what to do, sometimes more and some other less, raising this way the person's reaction time to press the keyboard's buttons so that he survives. Due to an in-game currency system and a variety of items to choose from the game can drastically raise a person's accurate decision-making skill, since he needs to select and buy items depending on the situation each time, which varies every time he starts a new run. Lastly, it offers a fun and relaxing experience helping the person's mental condition to prepare appropriately for the following day at work. A survey indicates that nearly 80% of gamers say video games provide them with mental stimulation, relaxation and stress relief (Hansa Bhargava, 2021), while on the other side, approximately 20% is affected negatively. According to WHO (World Health Organization, 2022), anxiety and depression increased by 25% after the beginning of the pandemic covid-19 and many countries included mental health and psychosocial support in their covid-19 response plans.

Having good mental health improves our overall well-being. Mental health affects how we think, act, and feel. It contributes to how we cope with stress, how we conduct our relationship, and the level of risk we assume with the choices we make. Good mental health involves being able to prioritize your own well-being and that of others, and being able alter your behavior when a situation or relationship isn't working for you, says Lela Moore (2023).

Another purpose of this project is to help, me as an individual, acquire or expand the knowledge of programming and game development even further, as well as get more accustomed to the game engine called Unity and its environment.

2. VIDEO GAMES

Over the past few decades video games have done a tremendous leap forward in every aspect and also have millions of people following and supporting this trend. But what exactly is a video game? According to Wikipedia Video games are interactive electronic entertainment that use visual displays and input devices to create simulated worlds and challenges for players. Games can be as simple as finding a specific word to exploring a whole new universe with new unimagined flora and fauna.

2.1 History of video games

The history of video games began in the 1950s and 1960sas computer scientists began designing simple games and simulations on minicomputers and mainframes. One of the first such games was Spacewar!, developed by Steve Russel, Martin Graetz and Wayne Wiitanen, MIT student hobbyists in 1962.

The first generation of video game consoles emerged in the 1970s, using dedicated hardware and cartridges to play games on television screens, allowing players to swap games too. Some of the popular consoles of this era were the Magnavox Odyssey, the Atari 2600, and the Fairchild Channel F. During this decade, Space Invaders led off what is considered to be the ***Golden Age*** of arcade games which lasted from 1978 to 1982. Several influenced and best-selling games arcade games were released during this period from Atari, Namco, Taito, Williams and Nintendo, including Asteroids (1979), Galaxian (1979), Defender (1980), Missile Command (1980), Tempest (1981) and Galaga (1981). The second generation of consoles introduced more advanced graphics, sound, and gameplay, as well as iconic titles like Pac-Man, Donkey Kong, and Super Mario Bros

The video game industry faced a major crisis in 1983, due to market saturation, low-quality games, and competition from personal computers. Many companies went bankrupt or left the industry, but some survived and innovated. The third generation of consoles, led by the Nintendo Entertainment System (NES), revived the industry with improved technology, quality control, and marketing. The fourth generation of consoles, such as the Sega Genesis and the Super Nintendo Entertainment System (SNES), introduced 16-bit graphics, digital sound, and online capabilities.

The fifth generation of consoles marked the transition to 3D graphics, optical discs, and 32-bit or 64-bit processors. Some of the influential consoles of this era were the Sony PlayStation, the Nintendo 64, and the Sega Saturn. The sixth generation of consoles continued to improve the graphics, sound, and storage capacity of video games, as well as introducing new features like motion control, wireless controllers, and online multiplayer. Some of the popular consoles of this era were the Sony PlayStation 2, the Nintendo GameCube, and the Microsoft Xbox.

The seventh generation of consoles brought high-definition graphics, online services, digital distribution, and social networking to video games. Some of the successful consoles of this era were the Nintendo Wii, the Sony PlayStation 3, and the Microsoft Xbox 360. The eighth generation of consoles, which began in 2012, introduced more powerful hardware, cloud computing, virtual reality, and streaming services to video games. Some of the current consoles are the Nintendo Switch, the Sony PlayStation 4, and the Microsoft Xbox One.

The ninth generation of consoles, which started in 2020, is expected to offer even more realistic graphics, immersive sound, faster loading times, and backward compatibility to video games. Some of the new consoles are the Sony PlayStation 5 and the Microsoft Xbox Series X/S (Wikipedia history of video games).

2.2 Advantages and disadvantages of video games

Despite the constant changes in complexity, educational and health solutions, simulations and other things that video games can provide, a lot of adults still to this day believe that video games do not offer anything and that they are a waste of human’s time. This way of thinking has to change, since any game with any genre can help a person deal with some of his problems. In fact, there are tons of genres and different games like simulating everyday life, doing complex calculations or even simulating an airplane’s cockpit gear letting the person fly an airplane (Microsoft Flight Simulator). Some of the advantages and disadvantages are listed below:

2.2.1 Advantages of video games

- They can help players obtain skills that someone wouldn’t normally get the chance to obtain like flying an airplane, be a car mechanic, space engineer and so on

- They can improve problem solving, decision making and critical thinking skills, as they often require players to face difficult challenges, plan strategies and adapt to changing situations
- There are many educational games or also known as serious games that can help children and adults learn math, spelling, manage resources, understand problems of the world, learn about animals, countries, etc. Serious games are the games that try to educate the person above everything else according to David Michael and Sandie Chen (2005)
- They can improve physical and mental skills, such as reaction time, memory, attention, creativity, spatial awareness
- They can provide entertainment, fun, relaxing moments for all players regardless of age
- They can help players build friendships, foster communication, collaboration and connect people from all around the world to achieve the same goal through the online multiplayer and/or cooperative modes that some games have or with the chat system that multiplayer games have.
- They can help people with stress, anxiety, depression or boredom by providing a distraction.

2.2.2 Disadvantages of video games

- They can become addictive and harmful for people who spend too many hours with them and can worsen their physical health if exercise is replaced with gaming.
- They can interfere with social relationships
- They can be very expensive and lead to a loss of fortune especially the mobile phone games which have a feature called In-app Purchases and make the people buy in-game items with real money in order to be stronger.
- They can cause physical problems such as headaches, neck pain, back pain, eye strain.
- They can make people less aware of their surroundings or reality.

Video games have evolved from simple experiments to complex forms of art and entertainment that appeal to millions of people around the world. They have influenced culture, society, education, and technology in various ways. Video games are constantly changing and adapting to new trends, challenges, and opportunities. Video games aren't bad for you as long as you aren't overdoing (Ila Adhikari, Filip Poutintsev, 2021).

3. SIMILAR VIDEO GAMES

In gaming platforms and in the market in general, there are already similar video games like the video game developed in this project. The main genres that these games have in common are Action, Rogue-like and/or Rogue-lite, Bullet-Hell, Survival and RPG. Co-founder of Black Shell Media Daniel Doan states that rogue-like games despite the many years that they are in the market, they got extremely popular the past few years because the games borrow mechanics from other genres and make them feel more familiar. Another factor that made this genre so popular is the slight changes made so that with every run the player has a small progress for future runs. This progression serves to enhance the games experience and helps players feel like they're making actual progress even through numerous deaths, so that they don't quit due to sheer frustration (article of why are modern roguelike games so popular 2017). Some similar video games with these genres are mentioned below:

Nomad Survival: It is a 2D video game released in 2022 developed and published by "The Fox Knocks" team and the main purpose of the game is to survive a horde of enemies for a specific amount of time. The difficulty of the game scales with every minute passing by and to counter this feature the player gets to choose a passive or active effect to add to his pool each time he levels up.

Vampire Survivors: This is another 2D video game released in 2022 developed and published by "poncle". Vampire survivors is a gothic horror casual game with rogue-lite elements, where the player's choices can allow him to quickly overpower the hundreds of monsters that get thrown at him or get overwhelmed by their number and strength. The gameplay of this game is to survive from thousands of night creatures for a fixed amount of time gaining more abilities and power on leveling up. In a recent post by Ken Allsop published in 2023 Vampire Survivors remained the best indie game of 2022 with over 200.000 reviews on pc and a positivity of 98%.

Halls of Torment: It is an isometric video game released in 2023 developed and published by “Chasing Carrots”. The player has to survive as long as he can from enemy hordes collecting as much gold as he can, so that he can boost some stats to make future run easier. It is another level-up based game to power up the character.

Brotato: Brotato is a 2D top-down arena shooter video game released in 2023 developed and published by “Blobfish”. Brotato offers a variety of characters with different and unique effects for the player to choose from, where he has to survive certain number of waves where enemies attack him. To get stronger, the player can choose from a variety of traits and items to create unique builds and survive until the end.

Enter The Gungeon: This is a 2D action bullet hell dungeon crawler video game released in 2016 developed by “Dodge Roll” and published by “Devolver Digital”. The player has to finish and conquer stages (maps) that consist of different sections, where in each section enemies appear to hinder the player of advancing. After defeating the enemies of a section, he can proceed exploring to the next area of the map, until he reaches the legendary Gungeon’s ultimate treasure: the gun that can kill the past.

Soulstone Survivors: It is a 2.5D action video game released in 2022 developed and published by “Game Smithing Limited”. Being a 2.5-dimensional video game means that the environment is in 3D but the gameplay and camera perspective are locked in 2 dimensions, ignoring the 3rd dimension (usually z-axis is ignored on a x, y, z axis system where x is width, y is length and z is height). Here the player is challenged to face the Lords of the void and their endless minions collect their Soulstones and become a god. It has a massive skill tree to unlock and power up the character, crafting system to craft new weapons and unlock multiple characters.

The Binding of Isaac, Rebirth: This is another 2D video game released in 2014 developed by “Nicalis. Inc, Edmund McMillen” and published by “Nicalis. Inc”. The Binding of Isaac is a randomly generated action RPG shooter with heavy Rogue-like elements. Players will find bizarre treasures that change Isaac's form giving him super human abilities and enabling him to fight off droves of mysterious creatures, discover secrets and fight his way to safety.

The above video games are the most known and popular video games for these specific genres. Of course, there are many more video games with a similar playstyle and challenges already on the market and they keep coming. It is a very popular playing experience for many gamers worldwide, if we consider that many of the above games have thousands of copies sold, even if they were recently released. The information provided above for each video game were taken from a website called Steamdb.

4. EXTERNAL TOOLS

In order to make a 2-dimensional game or a game in general game assets are needed. Assets in Unity can be 2D and 3D models, add-ons, sounds, tools, shaders and more. To make a 2D asset, in our case a 2D enemy or player, images are required or more precisely sprites. Sprite is a computer graphic which can be moved on-screen and be manipulated as a single entity. To make sprites or 2D images the need of external programs is required that allow the creation of images.

4.1 Photoshop

According to Wikipedia, Adobe Photoshop is a raster graphics editor developed and published by Adobe Inc for Windows and macOS. In computer graphics and digital photography, a raster graphics represents a two-dimensional picture as a rectangular matrix or grid of square pixels, viewable via a computer display, paper, or other display medium. It was originally created by Thomas and John Knoll in 1987 and since then the software has become the most used tool for professional digital art, especially in raster graphics editing. Photoshop can edit and compose raster images in multiple layers and support masks, alpha compositing and several color models. Also, it has limited abilities to edit or render text, as well as 3D graphics and video. This program was mainly used to create and/or edit images, to be inserted in Unity's game development environment.

4.2 Gimp

Like in Photoshop, the use of GIMP is identical of that in Photoshop, since these two programs are very similar. GIMP (GNU Image Manipulation Program) is a free and open-source raster graphics editor used for image manipulation and image editing, free-form drawing, transcoding between different image file formats and more specialized tasks. GIMP is available for Linux, macOS and Microsoft Windows and was released under the GPL-3.0 or later license. The GPL or General Public License is a series of widely used free software licenses that guarantee end users the four freedoms to run, study, modify and share the software. Its development began in 1995 from Spencer Kimball and Peter Mattis, originally named General Image Manipulation Program, as a semester-long project at the University of California, Berkley. After some years, they met Richard Stallman of the GNU Project and asked him to change the General of the application's name to GNU and he approved. GNU Project is a free software and its goal is to give computer users the freedom and control in their use if their computers and computing devices by collaboratively developing and publishing software that gives everyone the rights to freely run the software, copy and distribute it, study it and modify it. (Source: Wikipedia)

4.3 Microsoft visual studio

Microsoft Visual Studio is an IDE, integrated development environment, a software application for software development. Visual Studio is a creative launching pad that you can use to edit, debug, build code and then publish an app. Apart from the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers and many more features to enhance the software development process. In Microsoft Visual Studio the creation of any script and code was created that was needed for the game's mechanics and features.

4.4 Bing ai

Bing AI is a new AI-powered chat mode of Microsoft Bing that can help anyone get summarized answers and creative inspiration. with artificial intelligence developed by OpenAI. Bing Ai has customizations built on the language model, such as different input methods and interface features. Bing Ai moreover has integration with applications like Edge or Skype. Microsoft Bing uses features of DALL-E 2 too, an OpenAI-created large language model, to create original, realistic images and art from a text description by combining concepts, attributes and styles. This feature was used to create some images that were then imported in the above image editors to be polished, changed and cleaned of any noise generated in the images so that they are ready to be imported into Unity's environment.

4.5 Blender

Blender is a free and open-source 3D creation software developed by Ton Roosendaal, that supports the entire 3D pipeline, from modeling, sculpting, rigging, animation, simulation, rendering, compositing, motion tracking, video editing, and game creation. Blender is available for Windows, Mac OS, Linux, and other platforms. Blender has a large and active community of users and developers who contribute to its features, bug fixes, documentation, and tutorials. Blender is also used by professional studios and artists for creating stunning 3D content, such as the short film Sprite Fright. Blender is one of the most powerful and versatile 3D software in the world. It also supports all the above features in a 2D space making it an extremely powerful tool for both 2D and 3D projects. The use of this program was limited on this project despite the many features it provides and the main use was to animate 2D images by adding an armature on the image. Blender could also be used to make detailed and high-quality images for intermediate users by creating first a 3D model with the desired details, and then export an image of that object. The image can be further enhanced with the image processing features Blender provides, for example adding lighting or changing the RGB values of the image. This process is very time consuming since it depends on creating a 3D model first and then image processing so in this case, Blender was not used this way.

5. INFORMATION OF THE VIDEO GAME

The video game developed in this project is a two-dimensional (2D) action survival shoot them up arena game with rogue-like and rogue-lite elements. It was inspired by the game Brotato and has many similar features and mechanics as that game. However, this game combines the gameplay of Brotato with the gameplay of other games by adding more mechanics to it like a Skill tree and Upgrade points. The definitions of the above genres will be explained one by one below:

- Action: It emphasizes physical challenges and needs good coordination of eye – hand combination
- Rogue-like: It doesn't have the aspect of randomly generated maps like in this genre but the player has to start again from wave 1 after dying losing any progress he made
- Rogue-lite: By winning a run, there is a meta progression system which allows the player to upgrade some stats making future runs easier
- Shoot them up: The protagonist has to face a large number of enemies by shooting at them while dodging their damage sources. Shooting refers mainly to guns but in this game the player can equip melee weapons too
- Arena: Usually, a small map in which both the player and all the enemies exist simultaneously battling till the end.
- Survival: The player has to survive until the end of the game or else he has to try again from the beginning
- RPG: Standing for role-playing game. These games have a character which the player can control from start to finish

The goal for the player of this video game is to survive a set number of waves from the enemy for a specific time in each run. If the player survived a wave the next one starts increasing the time he has to survive linearly for a maximum of 20 waves. The survival of the player can be further made easier by using the upgrade system of the player's stats values which are acquired by winning a run. The player can wield a maximum of 6 different weapons, each of them targeting the closest enemy of the weapon's current position. This means that a total of 6 enemies can be targeted at the same time if they are scattered on the map and each weapon has a different target. The player controls a character who has a simple WASD movement script (WASD is a shortcut for horizontal and vertical movement system) as well as a skill which allows him to teleport a few units away in the direction he is walking by pressing the spacebar button. (Player sprite is taken from a website called webart2d and is a free to use asset)

5.1 Main menu

The first thing the player will see after launching the game, is the Main Menu interface. The menu interface is the first scene of the project and consists of mainly buttons and texts. There are 4 buttons Play, Database, Options and Exit (Image 1).

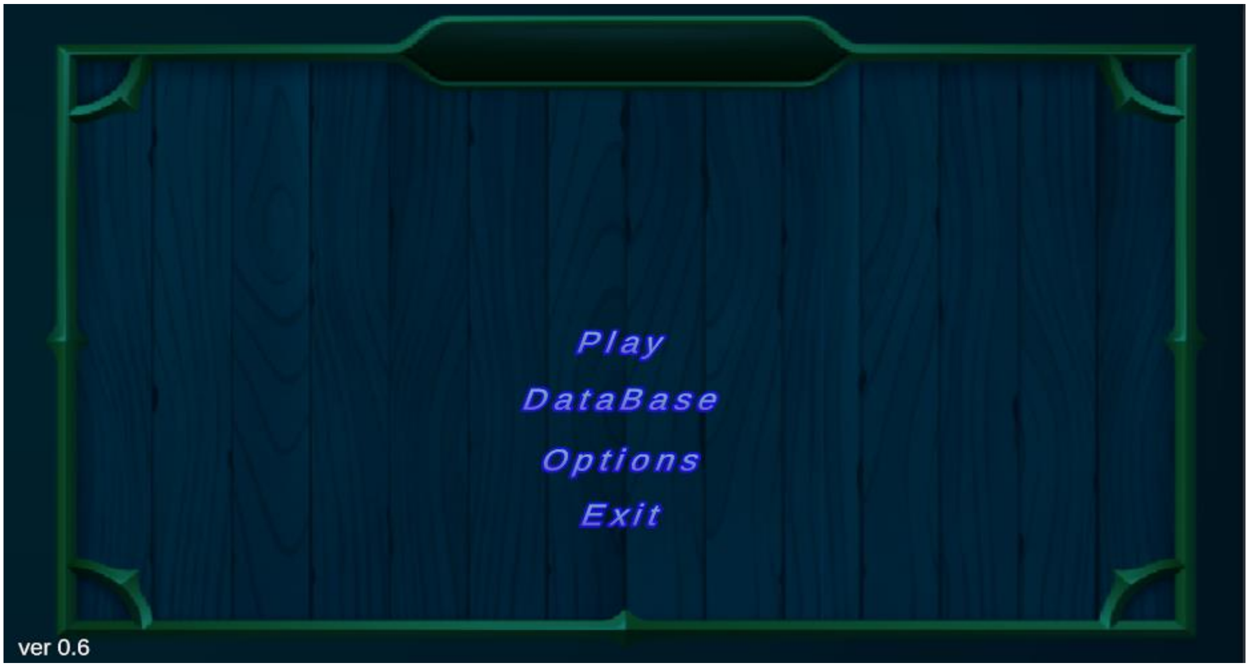


Image 1. Main menu

5.1.1 Play & exit buttons

Play and Exit buttons are very simple in understanding of what they are doing. Pressing Play loads the next scene where a part of the gameplay begins. On the other hand, pressing the Exit button closes the application and all background services that are needed to be running for a smooth gaming experience while the application is opened.

5.1.2 Options

Pressing the Options button, a new window appears with a list of game options like in all other games while the main menu attributes are hidden (Image 2). In an Options panel, the player can customize a variety of game assets or change the application’s window properties. Only the basic options are implemented in this video game which are explained below:

Full Screen: This toggles the application's window to be full screen or not allowing it to be dragged when it is turned off.

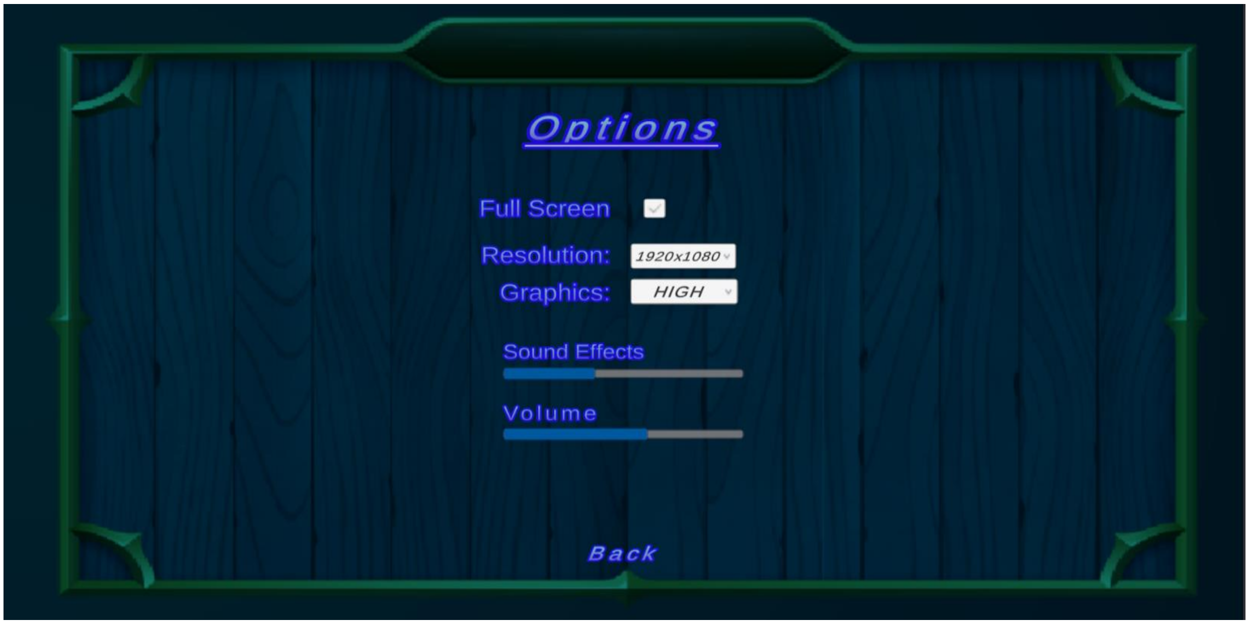
Resolution: Changing the resolution changes the number of displayed pixels horizontally and vertically (width and height). Lowering the resolution can increase performance but reduce the overall quality too. Additionally, combining low resolution and full screen (while it is toggled off) resizes and shrinks the application's window size allowing thus multitasking on same screen.

Graphics: Graphics provides three options for the player High (Default value), Medium and Low. Changing graphics from high to something else lowers the quality of the assets and game objects boosting performance. The impact of this feature is not visible in our case since everything is in 2D and almost everything are images. This would be noticeable in a 3D game where Unity changes the number of quads each 3D Object has, the reflection rate of lights on the surface and/or the shadow effects lowering its details.

Sound Effects: Sound effect is a sound that was created or edited for specific content in a video game making it more exciting and real. Currently the game has sound effects only in the first scene when interacting with buttons. The sound effects used in the project are created by Leohpaz and are available in Unity's Asset Store.

Volume: Volume is a slider scaling from 0 to 1 setting the game's music loudness. If the player doesn't want to listen to the game's music, he simply slides the slider all the way to the left and disables it. The music that is playing in the background is called "Sunny Day" created by Rizwan Ashraf and is also available in Unity's Asset Store.

Back: A simple button to close Options panel and return to previous menu which is the Main Menu.



Full Screen ☑

Resolution: 1920x1080

Graphics: HIGH

Image 2. Options

5.1.3 Data base

Data Base is the last interactable buttons of the Main Menu. Upon pressing it, a new menu will open having 3 main Tabs Enemies, Achievements and Items. Clicking on either of those new tabs will open a specific menu giving some information accordingly. Pressing “Enemies” will display all enemies that exist currently in the game to the left side and to the right side there is a container to display the stats and Information of the selected enemy (Image 3 and 4).

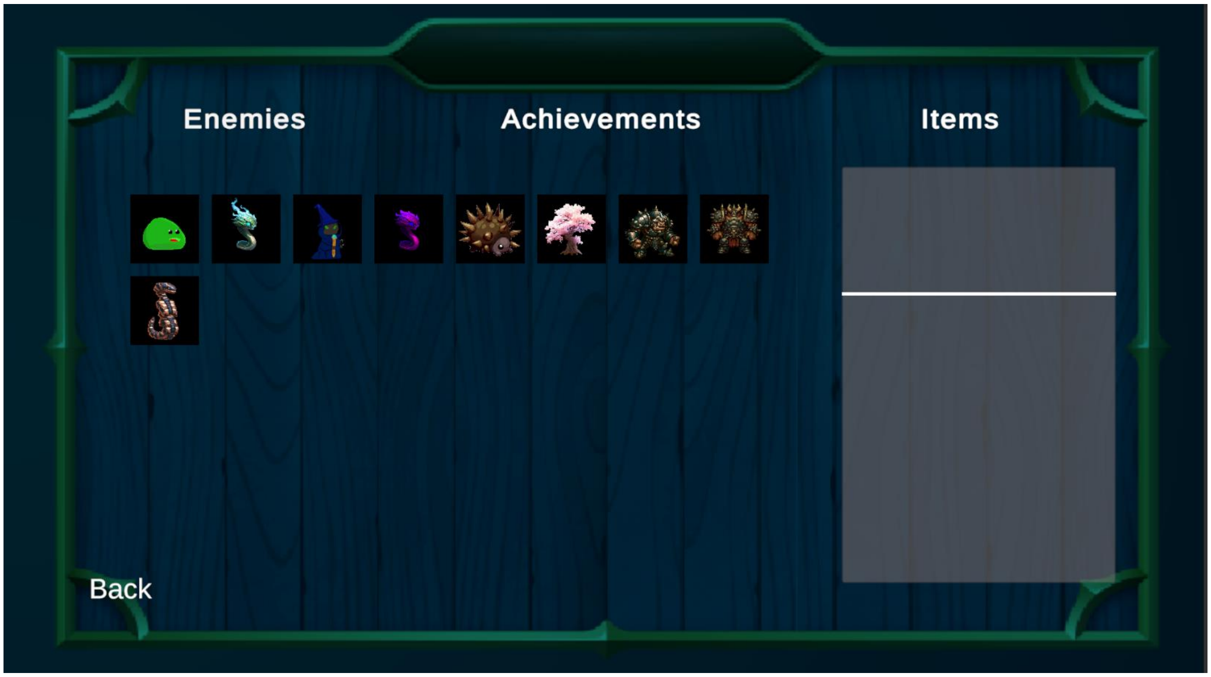
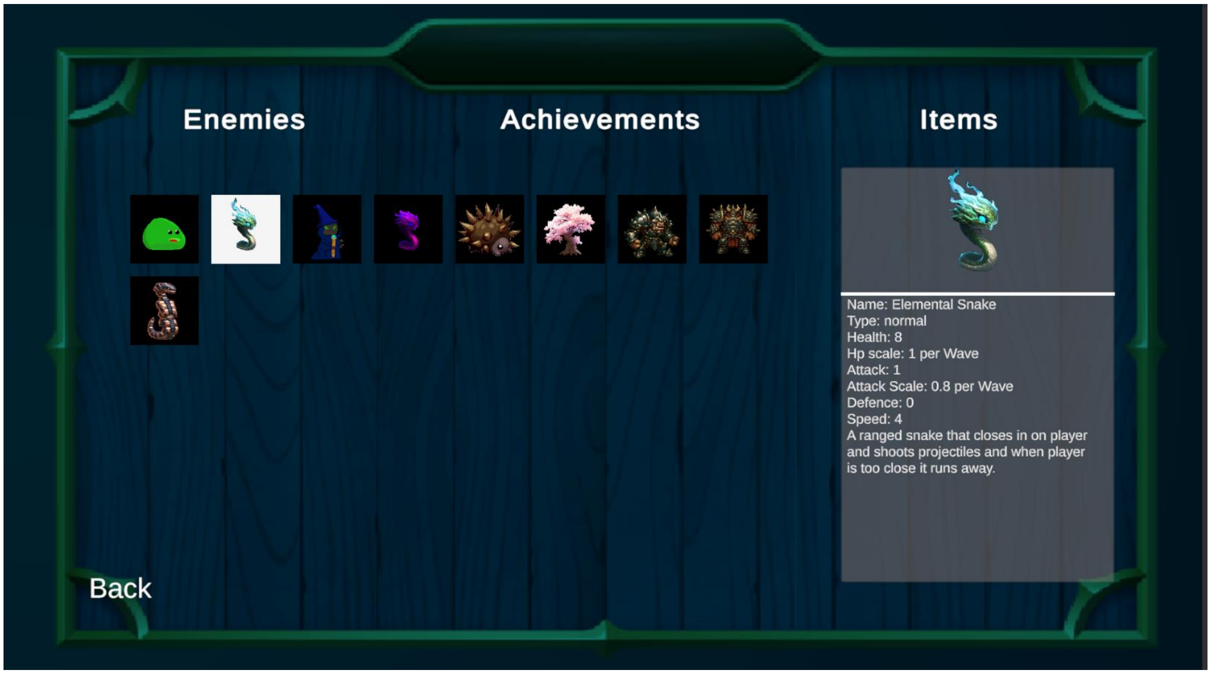


Image 3. Enemies



Name: Elemental Snake
Type: normal
Health: 8
Hp scale: 1 per Wave
Attack: 1
Attack Scale: 0.8 per Wave
Defence: 0
Speed: 4
A ranged snake that closes in on player and shoots projectiles and when player is too close it runs away.

Image 4. Enemies with selection

From the above image it is visible that there are 9 total enemies, 7 of which are normal enemies while the other 2 are bosses. Each enemy has his own stats and information to display. This was made with the use of Scriptable Objects. A Scriptable Object is a data container that you can use to save large amounts of data, independent of class instances. One of the main use cases for Scriptable Objects is to reduce a Project's memory usage by avoiding copies of values. With this method for instance each elemental snake enemy will copy the data of the same Scriptable Object saving thus memory since we don't save for each one of them the values in a class's field boosting performance, but also organizing and making it easier to handle all the data as a developer. In the diagram below we can see how the Slime monsters are connected to the same Scriptable Object for the Slime enemy (Image 5).

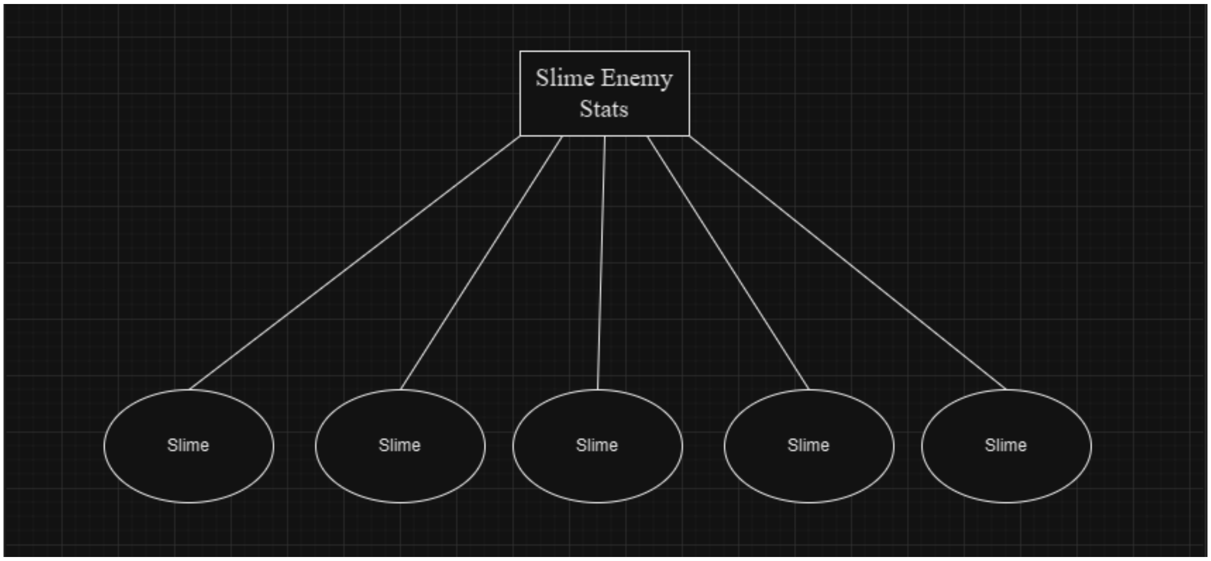
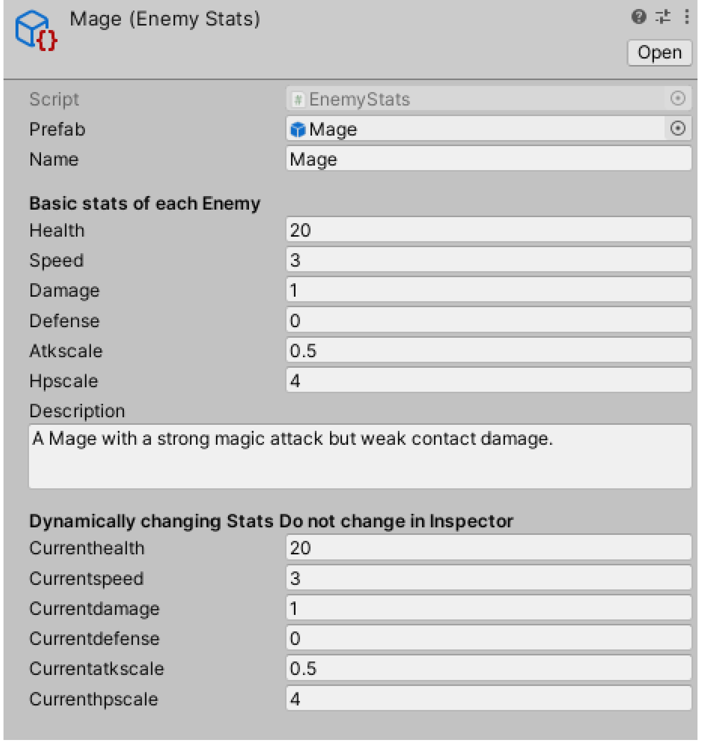


Image 5. Scriptable object example

From the 8 fields that the Scriptable Object has the 7 are connected to all enemies with the same name (Slime SO -> slime enemies, Mage SO -> mage enemies and so on). The only variable that has to be as a field in a script of the actual prefab is obviously Health because we want to have a unique number or the damage each enemy can take. If the Health variable was connected to the SO it would result in all enemies dying simultaneously of the same kind. However, the initialization of the variable will be done with the Health value of the SO, so that they all have the same amount of max Health points. If we didn't have SO then every enemy would have 8 fields that will need some space in memory slowing the game down unnecessarily. Now we save memory space equal to $7 * number\ of\ same\ enemy\ kind$. This probably seems small but imaging having 100 of those and 5 different kinds how much memory space it costs.

Image 6 below shows how the data is saved in a Scriptable Object called <Enemy Stats> and what fields it contains.



Mage (Enemy Stats)

Script	EnemyStats
Prefab	Mage
Name	Mage

Basic stats of each Enemy

Health	20
Speed	3
Damage	1
Defense	0
Atkscale	0.5
Hpscale	4

Description

A Mage with a strong magic attack but weak contact damage.

Dynamically changing Stats Do not change in Inspector

Currenthealth	20
Currentspeed	3
Currentdamage	1
Currentdefense	0
Currentatkscale	0.5
Currenthpscale	4

Image 6. Mage's scriptable object

As we can see it consists of several fields. A prefab, name, some stats values, a description and some more current stats. Obviously, the name is the name of the enemy and the stats are some values like health (how much damage the enemy can take before dying), Speed is how fast it moves, damage how much it damages the player, defense is the amount the damage received is reduced, attack and hp scale is the amount this enemy gets in those stats after each wave, description is for displaying some info in the Database menu and all the current stats are there to change throughout the game.

The need of current stats is necessary since the values changed in a Scriptable Object are until the player closes the application, but we need to reset them after each run the player does. This problem exists with Scriptable Objects because they are meant to be read only data but by changing the current Stats, we overpass that problem resetting them after each run again. The prefab field is filled with the final Game Object along with all of its components, property values and child game objects and will be Instantiated (created) at runtime. The use of Prefabs is to have game ready assets that can be simply dragged into the scene from the saved folder of the Project or be Instantiated at runtime through some code.

The prefab of the Mage enemy as an example consists of 3 Game Objects. The main Game Object has almost all components needed for the game. The components are an Animator which has all the animations of that specific enemy, a Sprite Renderer in order to render the image and make it visible in the scene, a Box Collider which determines if the enemy contacted the player to damage it, a Rigid Body that is responsible for all the physics on the Game Object and custom scripts for the gameplay. The main Game Object has 2 simple Children Game Objects one is an Empty Game Object which is just a point from where the projectiles the mage fires come from and the second child is another Game Object with a Collider to make the enemies collide with each other. Prefab of Mage shown below (Image 7).

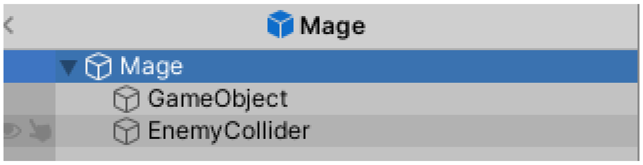
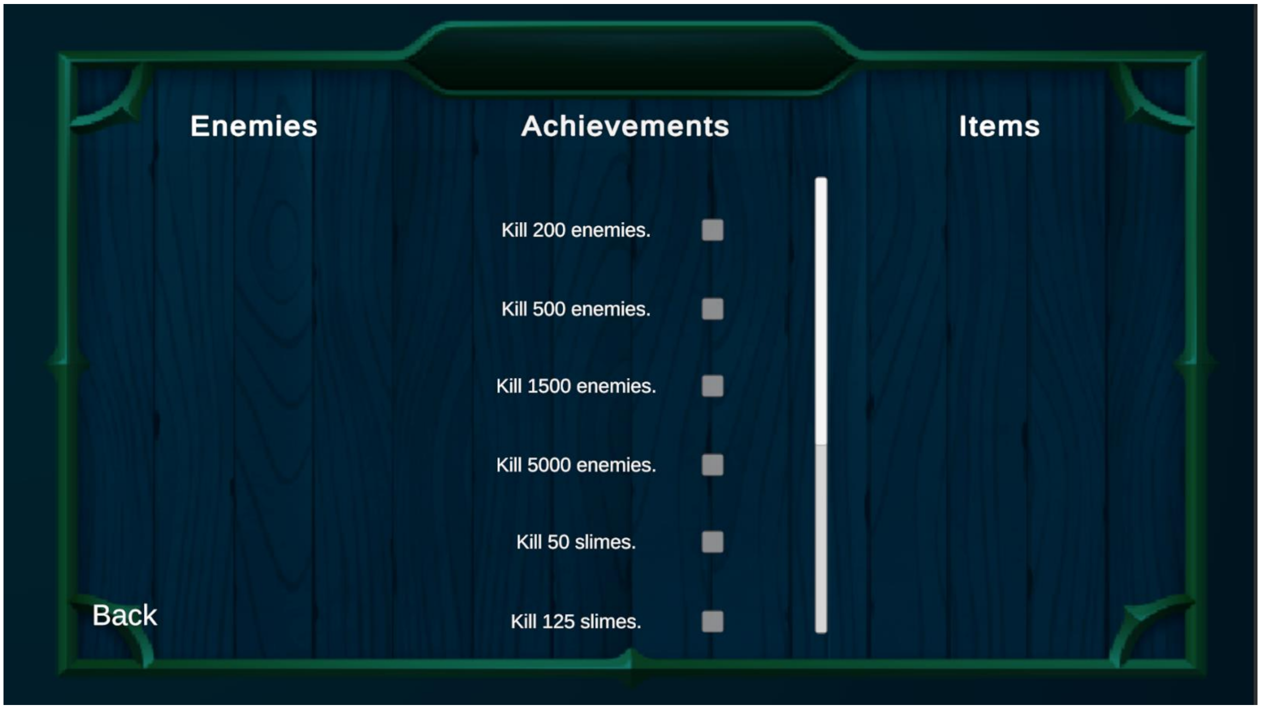


Image 7. Mage’s prefab

In the Achievements tab there are some simple challenges that the player can do while playing the game normally. According to Wikipedia in video gaming, an achievement (or a trophy) is a meta-goal defined outside a game's parameters, a digital reward that signifies a player’s mastery of a specific task or challenge within a video game. Unlike the in-game systems of quests, tasks, and/or levels that usually define the goals of a video game and have a direct effect on further gameplay. Achievements in games in general can provide many benefits, like extending the title’s longevity, completing goals lead to satisfaction, they create goal commitment, they provide players with the impetus to do more than simply complete the game but to also find all of its secrets and complete all of its challenges. Peter Finn noted the importance of including achievements in a game with a simple example. If a storytelling game has multiple endings the player after finishing the game wouldn’t care or know about the other endings and this adds one more reason to see all the endings and see the whole story of the game.

On pressing the Achievements tab, a scroll panel appears listing all available achievements with a mark to the right if it is unlocked or not as shown below (Image 8).



Kill 200 enemies. □

Kill 500 enemies. □

Kill 1500 enemies. □

Kill 5000 enemies. □

Kill 50 slimes. □

Kill 125 slimes. □

Image 8. Achievements interface

Currently there are 10 achievements total in the game, each listed below:

- Kill 200 enemies
- Kill 500 enemies
- Kill 1500 enemies
- Kill 5000 enemies
- Kill 50 slimes (specific enemy)
- Kill 125 slimes
- Kill 550 slimes
- Collect 1000 gold
- Collect 5000 gold
- Collect 20000 gold

The main reason why there are only a few of them is to demonstrate the main idea of the achievement system and to show its functionality. After playing for a little and revisiting the achievement tab in the main menu the player will notice that some achievements will be marked as done if he has fulfilled the requirements.

The Items tab displays all the purchasable items and weapons in the shop. They are divided in 4 groups according to their quality (rank) which can be from 1 to 4. Each item has a fixed rank and it won't change by any means while playing, however the player can find any rank for a weapon (from 1 to 4) in the shop which is calculated based on probabilities. A rank 1 weapon is weaker compared to the same weapon with 1 rank higher (rank 2). The probability calculation is dynamic which means that first the rarest rank is calculated and then the other ranks are calculated based on that value. For instance, let's say that we want to calculate the probabilities of finding a rank 1, 2, 3 or rank 4 Item/weapon on wave 9. To make it simpler and easier to read the letter P will mean Probability so for the probability of wave 9 we can type P9. If we want to mention which rank's probability it is, we type P9(1). This translates to probability of rank 1 Items/weapons on wave 9. Below is the formula for these calculations.

Rank 4: P9 = P9(4) linear equation increasing a fixed amount each wave so that it is close to 5% on last wave.

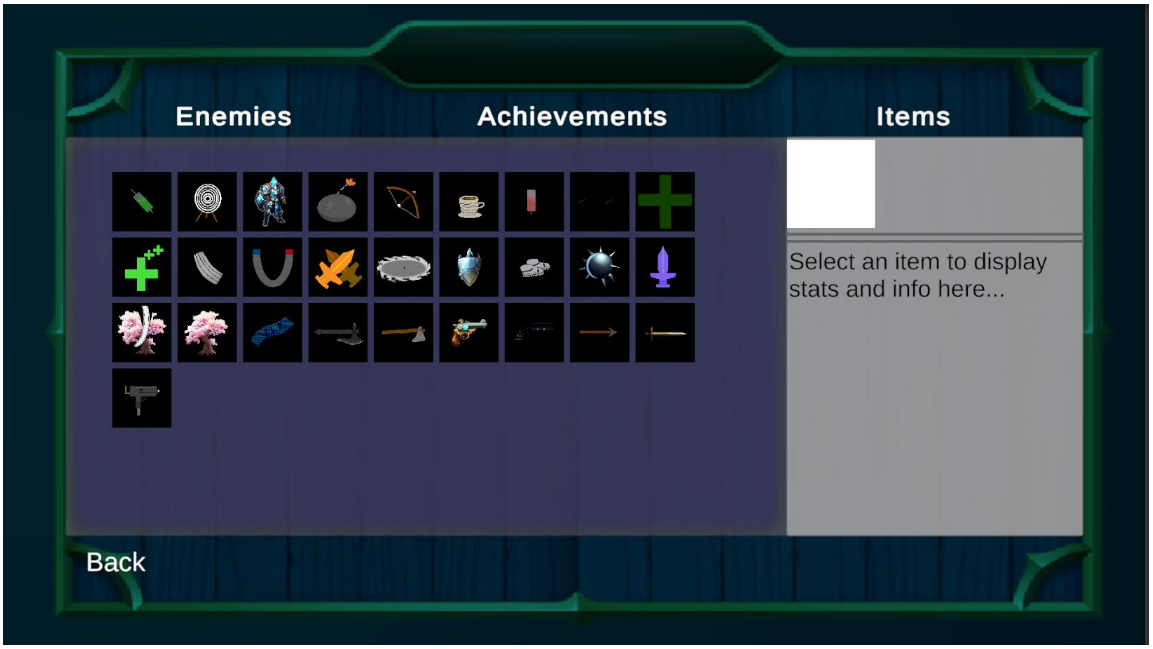
Rank 3: P9 = P9(3), also a linear scaling so that the final value is around 30%.

Rank 2: P9 = y – P9(3) – P9(4) with a max limit of 45%. If this equation's outcome is above 45% it is set to 45%. Y is a scaling factor based on the wave.

Rank 1: P9 = 1 – P9(2) – P9(3) – P9(4). The remaining percentage is the probability of a rank 1 Item or weapon.

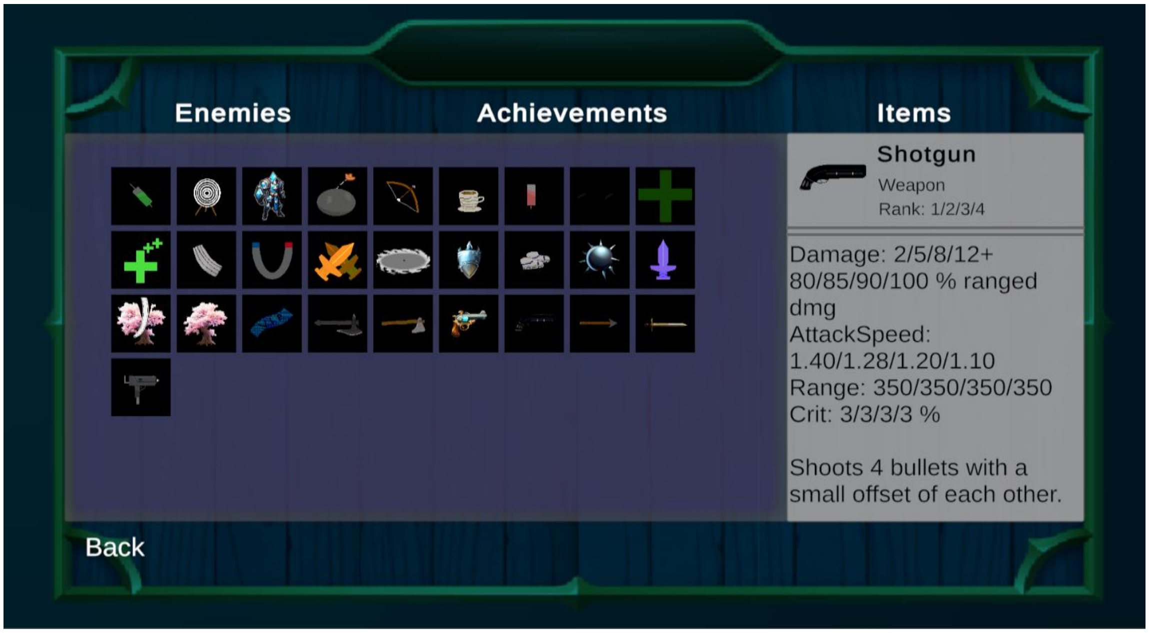
It is worth noting that the probabilities of the higher ranks may be 0 in some waves. The idea behind this is to limit the immediate growth of the player because it wouldn't be a challenge. The difficulty on wave 6 changes so Rank 3 Items and weapons are unlocked on wave 5 and onward so that the player has a chance of finding them and get prepared.

On the right side of the new panel just like in the enemy's tab there is a container to display the currently selected item with all the information it needs to provide to the player. (Image 9 - 10)



Select an item to display stats and info here...

Image 9. Items Tab



Shotgun
Weapon
Rank: 1/2/3/4

Damage: 2/5/8/12+ 80/85/90/100 % ranged dmg
AttackSpeed: 1.40/1.28/1.20/1.10
Range: 350/350/350/350
Crit: 3/3/3/3 %

Shoots 4 bullets with a small offset of each other.

Image 10. Items tab with information

5.2 Town – village (or portal map)

Like mentioned above this scene is loaded after pressing ‘Play’ button in the Main Menu. This is where the player gets to control the character for the first time. In the town there is a forger in which the player can boost his stats by upgrading an equipment spending upgrade tokens which are acquired by winning a run on any difficulty (1 run won equals 1 upgrade token). There is also a Portal to start the battle and load the next Scene which is the battle map. All interactions in the town with the Game Objects is done by pressing the key button “f” while near the game object. For the current being it is hardcoded and can’t be changed but it will be flexible in the future. Below is the image of the Town (Image 11).

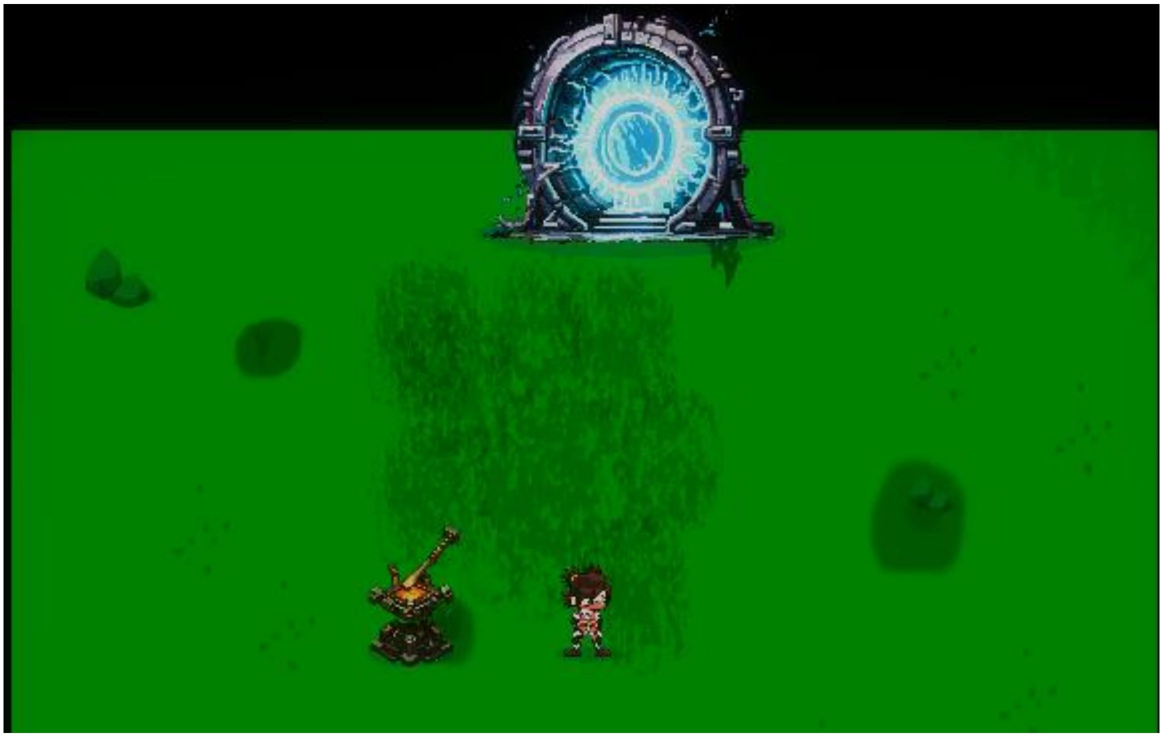
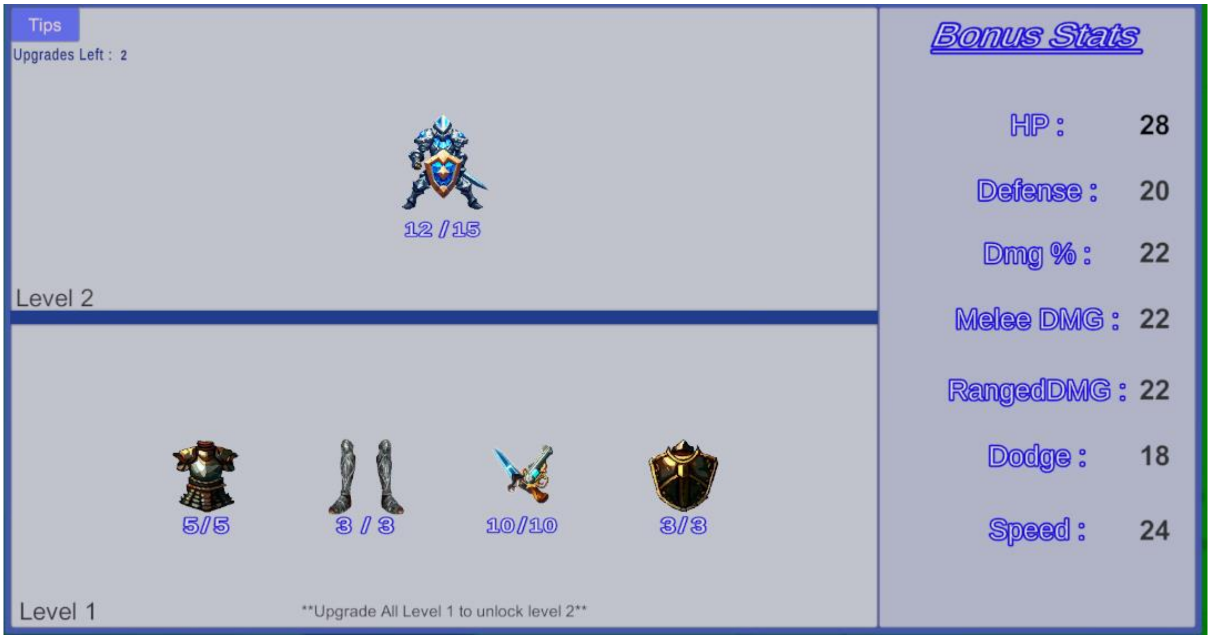


Image 11. Portal map

5.2.1 The forger

Interacting with this Game Object opens a new UI that displays the equipments that the player can upgrade, the number of available upgrade tokens and the bonus stats obtained by forging and upgrading the equipment. The equipment is divided into Level 1 and Level 2. There are 4 level 1 equipments which can be upgraded immediately and 1 level 2 equipment which its upgrade availability is unlocked after upgrading all Level 1 equipments to their limit. Hovering over with the mouse of an equipment makes a new window pop up showing what upgrades the equipment gives. The available stats that can be boosted with the equipments are Health Points, Defense, Damage Percentage, Melee Damage, Ranged Damage, Dodge and Speed. (Image 12)



HP : 28

Defense : 20

Dmg % : 22

Melee DMG : 22

RangedDMG : 22

Dodge : 18

Speed : 24

Image 12. Equipment ui

The Level 1 equipments are:

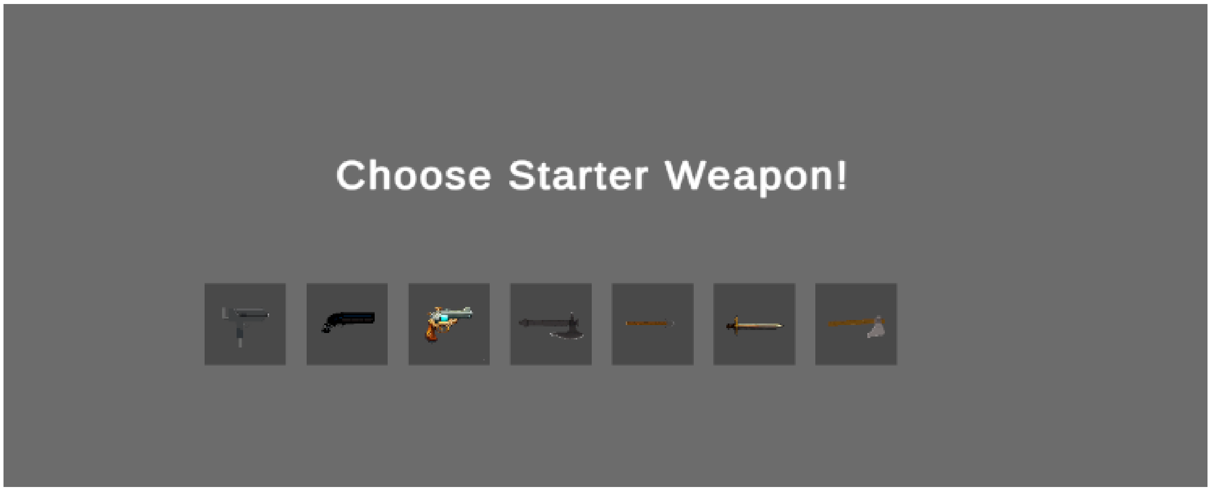
- Chest Plate: Max upgrades -> 5. With each upgrade it gives 2 Health points and 1 armor to the player
- Boots: Max upgrades -> 3. Each upgrade gives 1 Health point, 1% Dodge and 4% Movement speed
- Weaponry: Max upgrades -> 10. Gives 1% Damage, 1 melee damage and 1 ranged damage per upgrade
- Shield: Max upgrades -> 3. Gives 1 Health, 1% Dodge and 1 Armor per upgrade

The Level 2 equipment is:

- Set Upgrade: Max upgrades -> 15. Gives 1 Health, 1 Armor, 1% Damage, 1 Melee and Ranged Damage, 1% Dodge and 1% Speed per upgrade

5.2.2 Portal

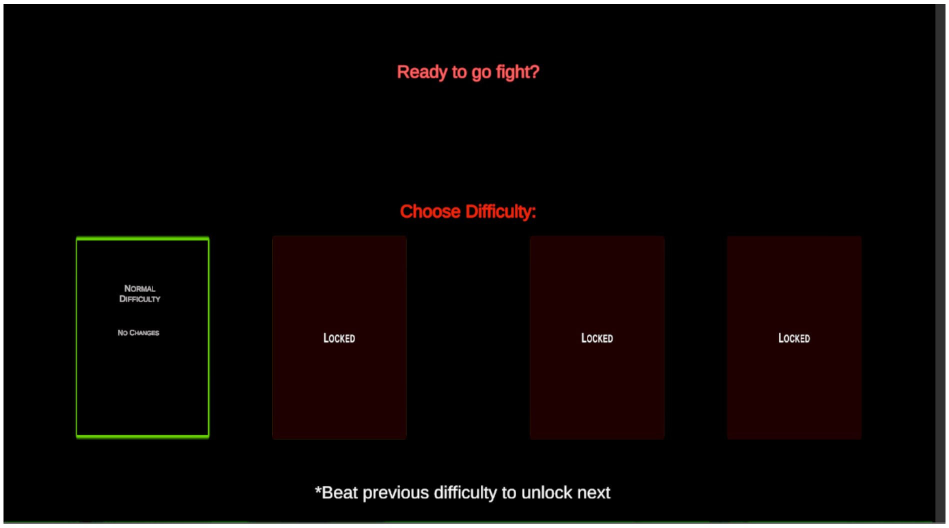
The Portal is another interactable Object in this Scene which will open a new window by pressing the “F” button while near it. When the interaction button is pressed while near it a window pops up letting the player choose his starter weapon -1 weapon from the list - (Image 13).



Choose Starter Weapon!

Image 13. Weapon selection

After choosing the starter weapon the player has to choose the difficulty. There are a total of 4 Difficulties the player can choose from, Normal, Hard, Very Hard and No Tomorrow. If the player has no progress, then the only Difficulty he can pick is Normal since the other are locked (Image 14).



Ready to go fight?

Choose Difficulty:

*Beat previous difficulty to unlock next

Image 14. Difficulty selection

The next Difficulty can be unlocked by beating the previous one. The new Difficulties come with extras too. Normal mode is the default difficulty with the main gameplay. On Hard difficulty the enemy spawn rate increases making it more difficult. On very hard mode Boss enemies start to spawn at wave 10 too instead of only in wave 20 which is the last wave. On No Tomorrow Difficulty environmental events start to occur making the survival of the player even more difficult. Such environmental events are fireballs flying horizontally or vertically in a sine wave movement that damage the player on contact. It is important to note that every effect on the Difficulties is additive which means that the last Difficulty has all the effects of the previous ones too apart from the new effect. (Image 15)

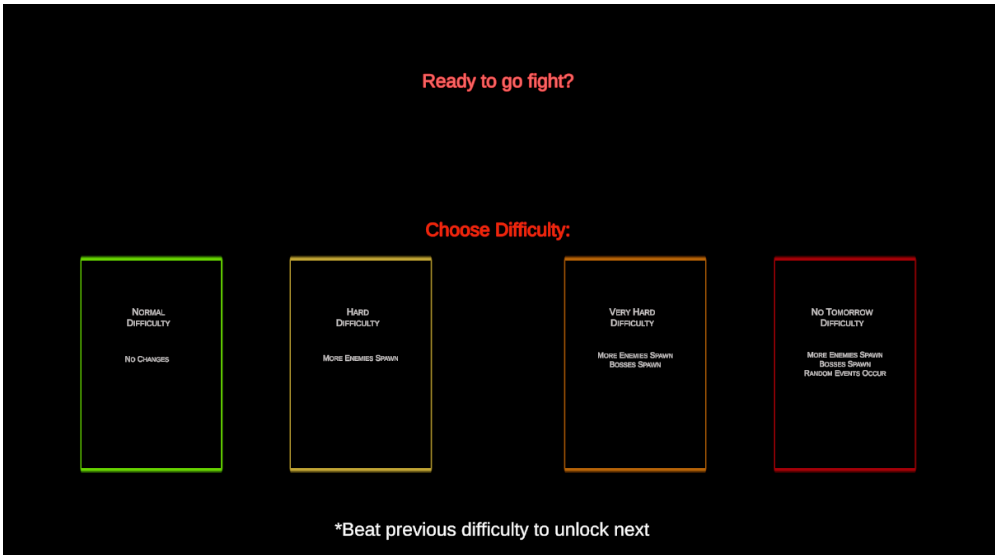
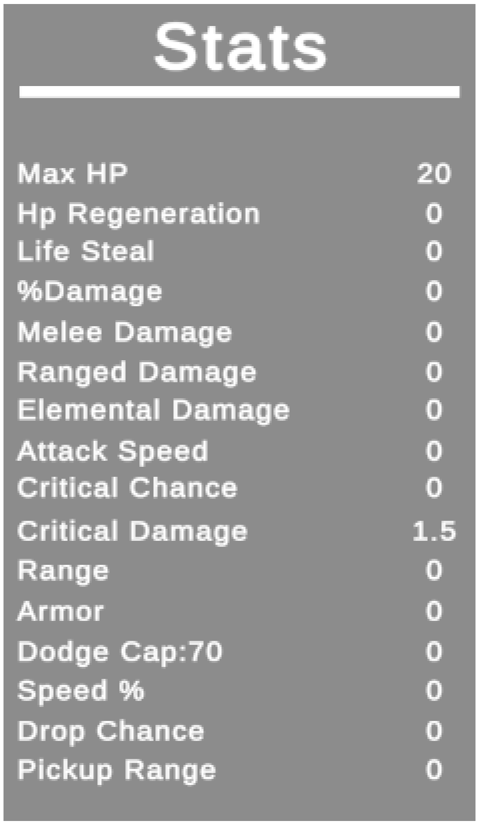


Image 15. Difficulty selection (all unlocked)

5.2.3 Character

The Character is the most significant Game Object in most video games. The Sprite of this character is a free online asset which anyone can download in gamearts2D webpage (link provided in the end of the document). Its components are very similar to the components of the enemies with some differences in the scripts plus it has additionally an input controller for detecting the keyboard, mouse and Gaming Controller inputs (Play Station 4 Controller). The support with a PS4 controller is partially which means that it is not supported at full extend. It is possible to control the character with the PS4 Controller but navigating in the UI is not supported yet. The navigation in the UI has to be done with the mouse. The player can use his skill which is a small teleportation in the direction of the input by pressing the “space” button or the “x” button on the controller.

The player has several stats too, some of which we have already seen in the equipment upgrade section. These stats are saved and change at runtime in a SO like all the enemies. In the image (Image 16) below we see all the changeable stats while playing.



Stats

Max HP	20
Hp Regeneration	0
Life Steal	0
%Damage	0
Melee Damage	0
Ranged Damage	0
Elemental Damage	0
Attack Speed	0
Critical Chance	0
Critical Damage	1.5
Range	0
Armor	0
Dodge Cap:70	0
Speed %	0
Drop Chance	0
Pickup Range	0

Image 16. Stats

- Max HP: Current max health points
- Hp Regeneration: Regenerates 1 health point every x seconds. Equation $x = \frac{5}{1+(Regen-1)/3.25}$ seconds
- Life Steal: Heals for 1 Health point each time it is triggered. 1 life steal = 1% chance.
- % Damage: Overall damage amplifier in percentage

- Melee Damage: Boost in Damage for all melee weapons and items based on their amplifier
- Ranged Damage: Boost in Damage for all ranged weapons and items based on their amplifier
- Elemental Damage: Boost in Damage for all elemental weapons and items based on their amplifier
- Attack Speed: How fast the weapon is ready to attack again in seconds. Equation $Attack\ speed = \frac{weapon\ attack\ speed}{1+attack\ speed*0.01}$ seconds. If attack speed is 100 it means that the weapon attacks twice as fast. For instance, if the base attack speed is 1.5s with 100 attack speed it will attack every $\frac{1.5}{1+100+0.01} = 0.75$ seconds
- Critical Chance: The % probability to deal more damage
- Critical Damage: Default value is 1.5. The damage amplifier when critical hit occurs
- Range: From how far the player can attack. Each weapon adds its range to this value for the new range value. Melee weapons range is changed by 25% of range value
- Armor: Flat damage reduction. Damage received is at least 1
- Dodge: Chance to evade completely an attack without receiving any damage. Percentage value. 1 Dodge = 1% probability with a max of 70%
- Drop Chance: It is the probability bonus of enemies dropping consumables (Steak item healing player for 3 HP)
- Pickup Range: The range of which the consumables and coins start to be attracted by the player. This is a percentage bonus too which means that the default pick-up range value is increased by this value

6. DEVELOPMENT OF THE VIDEO GAME

A video game has a lot of scripts are running simultaneously and/or individually every frame or when they are accessed. To write such scripts the developer needs to decide on an existing programming language and write code based on that programming language’s standards. Unity supports a variety of programming languages like UnityScript which is similar to JavaScript, Boo which is similar to Python, Rust, C/C++, IronPython which is a Python implementation that can run on .Net framework, Lua which is a lightweight scripting language that can be integrated with Unity using third-party tools and C# which is the most widely used and recommended language for Unity development.

6.1 Programming language

From the above programming languages, the language that was used to create and write all the scripts for this project is C# developed by Anders Hejlsberg from Microsoft in 2000 which was later approved as an international standard by Ecma in 2002 and ISO/IEC (ISO/IEC 23270) in 2003. C# (sharp) is a modern**,** open-source**,** cross-platform, object-oriented programming language for the .NET developer platform. C# is used to develop web apps, desktop apps, mobile apps, games, and much more. C# has its roots in the C family of languages and will be immediately familiar to C, C++, Java, and JavaScript programmers. C# offers many features and benefits, such as strong typing, generics, delegates, lambda expressions, LINQ, async/await, and more. C# is one of the most popular and versatile programming languages in the world.

6.2 Spawn system

A script is responsible for summoning the enemies. The summoning is every fixed amount of time and it changes depending on the current wave and difficulty. There are 3 ways for enemies to be summoned (Instantiated on the map).

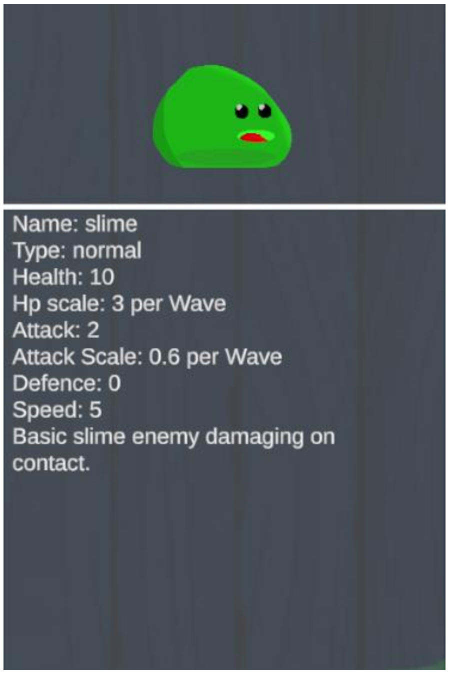
The first is by selecting a random enemy from the enemy pool and spawn it randomly inside the map borders, the second one is to summon a specific type of enemy, which also spawns somewhere in the map randomly and the last one is to massive spawn an enemy type each close to each other like a small pack. The number of the massive spawn is calculated based on difficulty and current wave (the later in the run the more are spawned). The last spawn method randomly picks an enemy and spawns that many copies as the number calculated in the script.

In early waves the probability of getting weaker enemies is higher than getting stronger enemies (enemies with higher stats). At Boss waves the number of enemies that spawn is reduced to let the player focus and battle the Boss. When an enemy dies (Health points are reduced to 0), it drops a coin which can be picked up by the player to increase his budget and also have a chance to drop a Steak which is a consumable that heals the player for a fixed amount when picked up.

If the player successfully survives a wave a small and simple animation is played noting that the player won the wave and after that the game pauses and the shop activates. This happens after every wave except wave 20 which is the winning condition. The transitions and alerting that the player won the wave in the scripts is done with Coroutines and Events/Delegates. So below we will see and explain what these are.

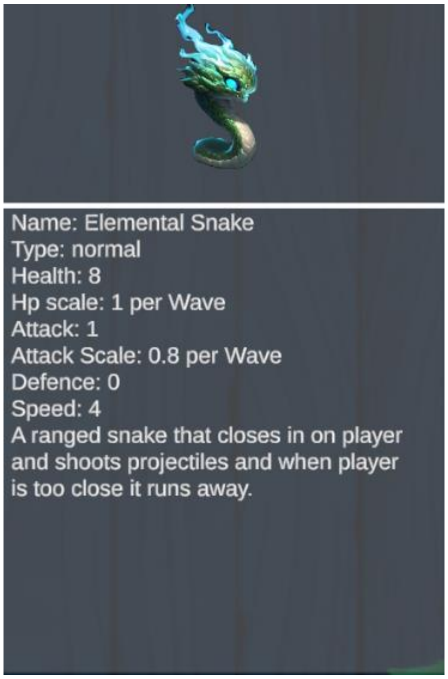
6.2.1 Enemies

There are currently 9 different enemies and are listed below (Images 17 to 25)



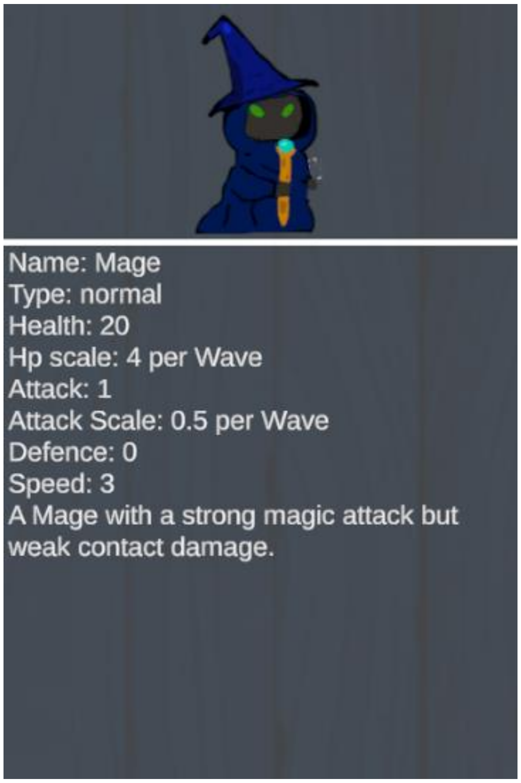
Name: slime
Type: normal
Health: 10
Hp scale: 3 per Wave
Attack: 2
Attack Scale: 0.6 per Wave
Defence: 0
Speed: 5
Basic slime enemy damaging on contact.

Image 17. Slime



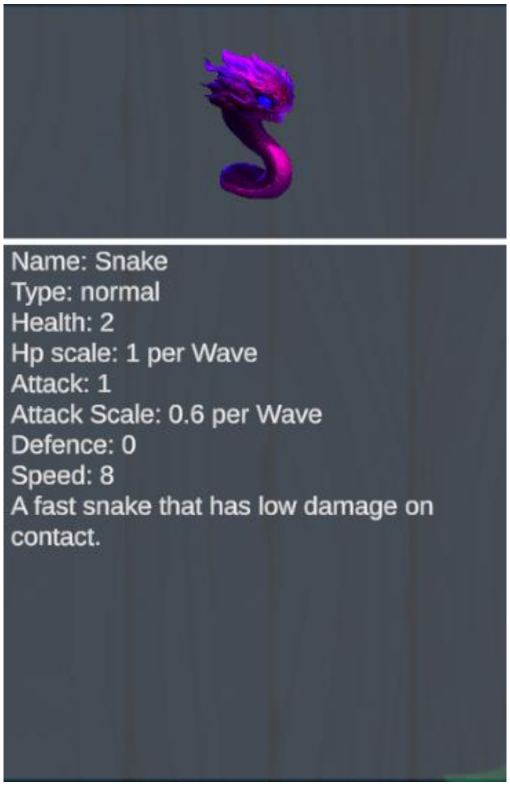
Name: Elemental Snake
Type: normal
Health: 8
Hp scale: 1 per Wave
Attack: 1
Attack Scale: 0.8 per Wave
Defence: 0
Speed: 4
A ranged snake that closes in on player and shoots projectiles and when player is too close it runs away.

Image 18. Elemental Snake



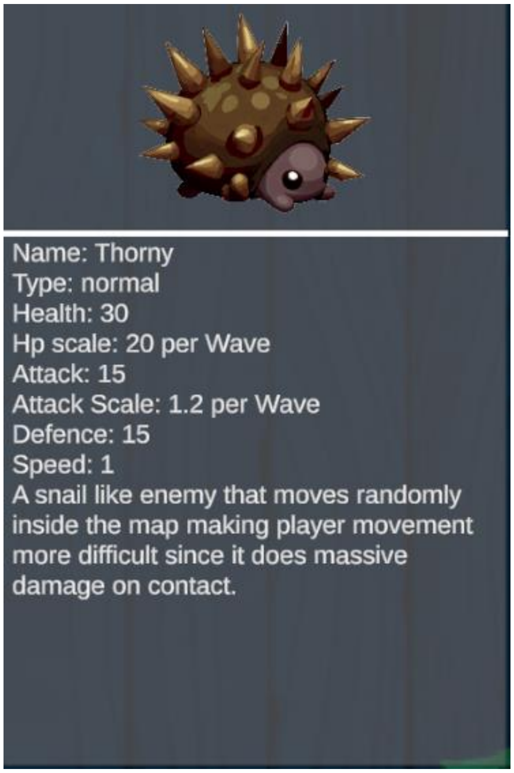
Name: Mage
Type: normal
Health: 20
Hp scale: 4 per Wave
Attack: 1
Attack Scale: 0.5 per Wave
Defence: 0
Speed: 3
A Mage with a strong magic attack but weak contact damage.

Image 19. Mage



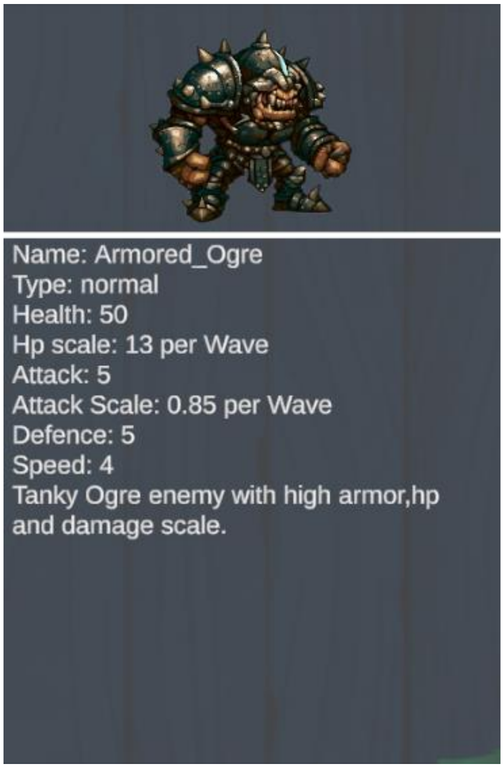
Name: Snake
Type: normal
Health: 2
Hp scale: 1 per Wave
Attack: 1
Attack Scale: 0.6 per Wave
Defence: 0
Speed: 8
A fast snake that has low damage on contact.

Image 20. Melee Snake



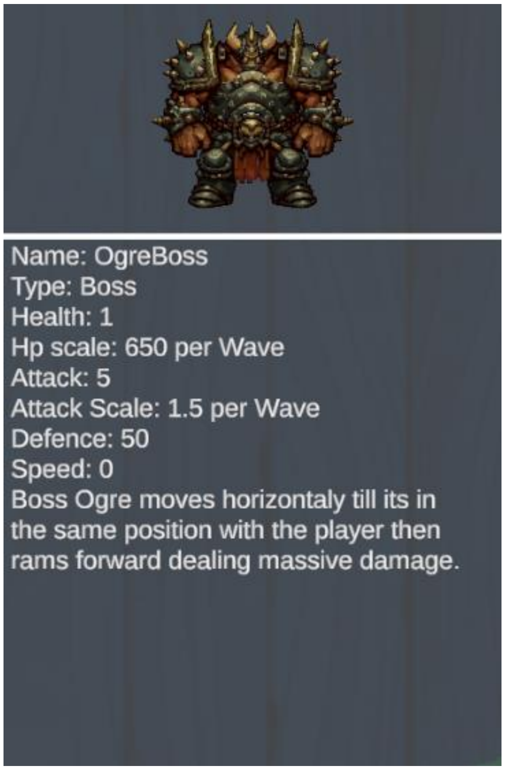
Name: Thorny
Type: normal
Health: 30
Hp scale: 20 per Wave
Attack: 15
Attack Scale: 1.2 per Wave
Defence: 15
Speed: 1
A snail like enemy that moves randomly inside the map making player movement more difficult since it does massive damage on contact.

Image 21. Thorny



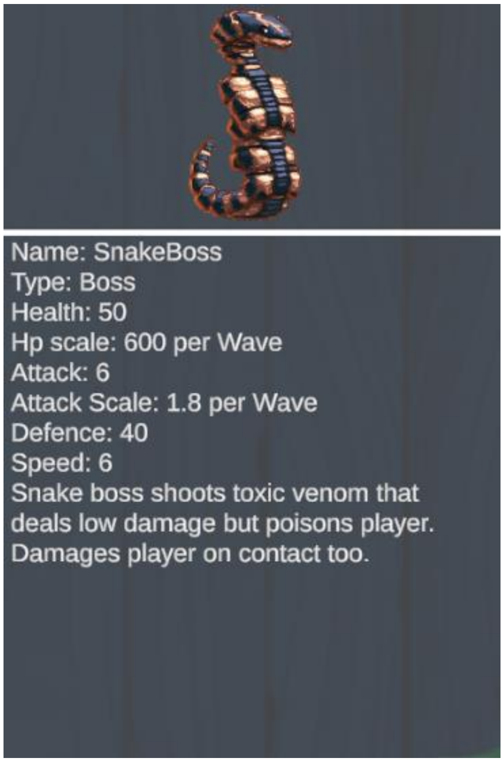
Name: Armored_Ogre
Type: normal
Health: 50
Hp scale: 13 per Wave
Attack: 5
Attack Scale: 0.85 per Wave
Defence: 5
Speed: 4
Tanky Ogre enemy with high armor,hp and damage scale.

Image 22. Armored Ogre



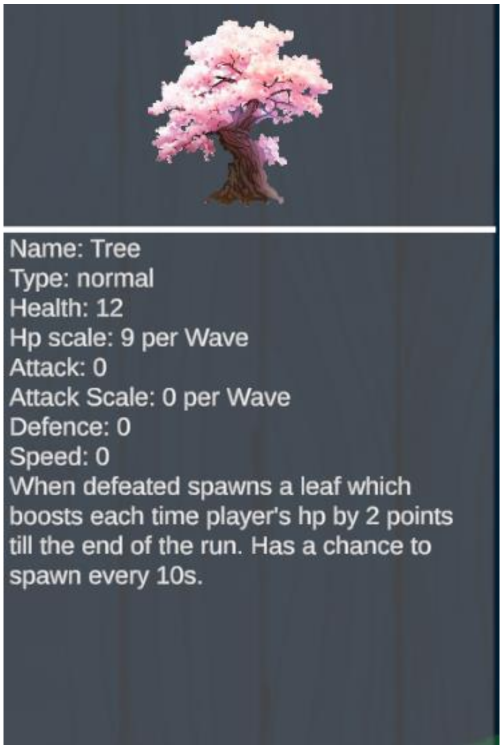
Name: OgreBoss
Type: Boss
Health: 1
Hp scale: 650 per Wave
Attack: 5
Attack Scale: 1.5 per Wave
Defence: 50
Speed: 0
Boss Ogre moves horizontaly till its in the same position with the player then rams forward dealing massive damage.

Image 23. Ogre Boss



Name: SnakeBoss
Type: Boss
Health: 50
Hp scale: 600 per Wave
Attack: 6
Attack Scale: 1.8 per Wave
Defence: 40
Speed: 6
Snake boss shoots toxic venom that deals low damage but poisons player. Damages player on contact too.

Image 24. Snake Boss



Name: Tree
Type: normal
Health: 12
Hp scale: 9 per Wave
Attack: 0
Attack Scale: 0 per Wave
Defence: 0
Speed: 0
When defeated spawns a leaf which boosts each time player's hp by 2 points till the end of the run. Has a chance to spawn every 10s.

Image 25. Tree

All it takes to create more enemy stats Scriptable Objects are 3 steps. Right click inside the project in any folder choose Create => ScriptableObjects => EnemyStats, a new Scriptable object is created, change the values and then link the data to a Prefab (Image 26). The link between the prefab and the scriptable object is done through code so I'll skip that part.

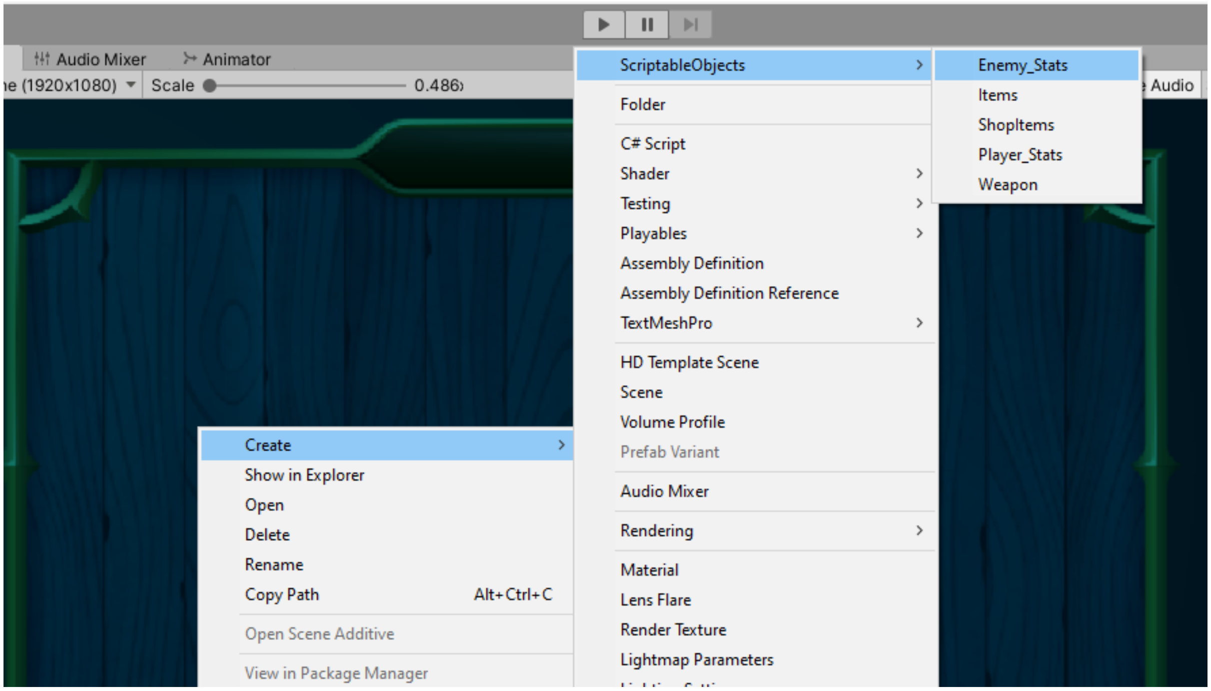


Image 26. New enemy data

6.3 Delegates or events

A Delegate is an object which refers to a method or it can be called a reference type variable that can hold a reference to the methods. Delegates in C# have a similar use like pointers in C or C++. It provides a way to tell us which method to be called when an event is triggered.

Events on the other hand are special kind of delegates that can be invoked only inside the class or derived classes where they are declared. The class that has the event inside is called the publisher and all other classes that have subscribed to the event are listening to it and will call their method when the event is fired. The methods called can be different on each listening class.

Either having a delegate or having an event a publisher and subscribers are required. If the publisher has more than one subscriber then all the subscribers are triggered executing their code.

The main reason to use delegates/events is that we simplify our code and that we don't care or know who is listening and what their actions will be. An example is shown below with 1 publisher that fires a Print Event and 3 subscribers. The subscriber 1 and 2 have a print method while the subscriber 3 has none. After the event is fired all 3 subscribers get alerted and the Print Methods are called so from Subscriber 1 we get "SUB 1", from Subscriber 2 we get "SUB 2" and from Subscriber 3 we don't get anything because it doesn't print anything or it has another method called when the event is triggered. The method called on Subscriber 3 could be something completely different like adding 2 numbers or finding the max number of an array. (Image 27)

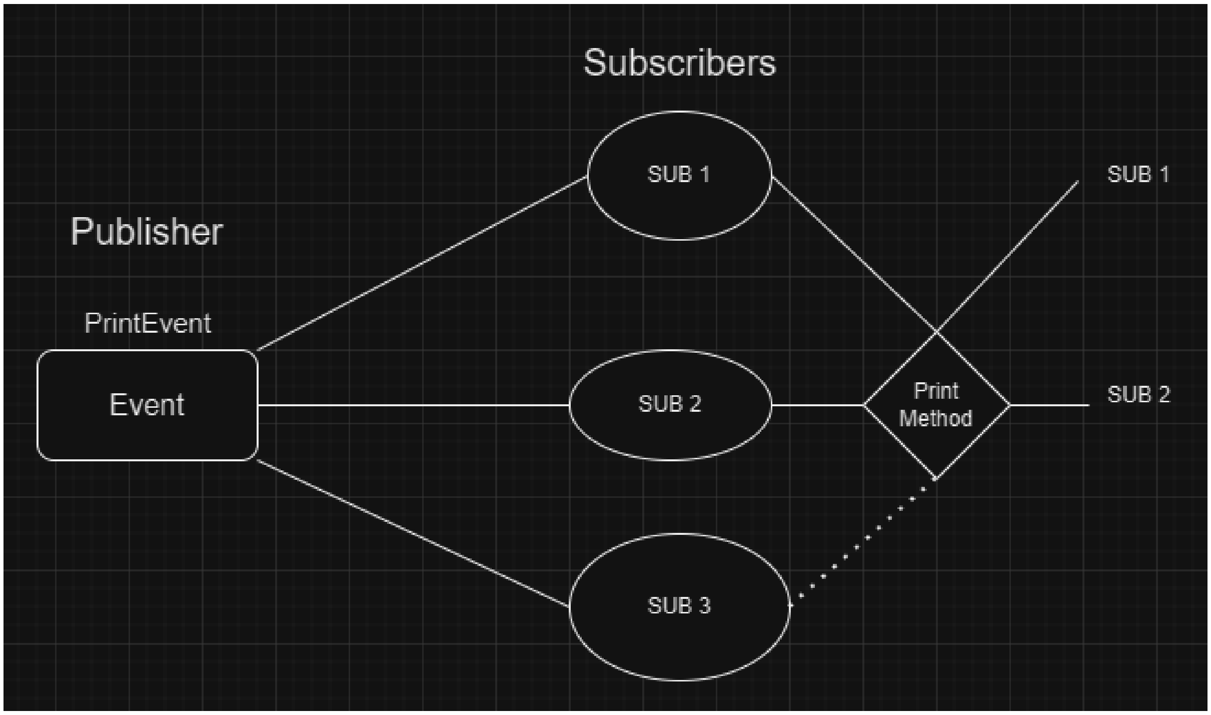


Image 27. Event example

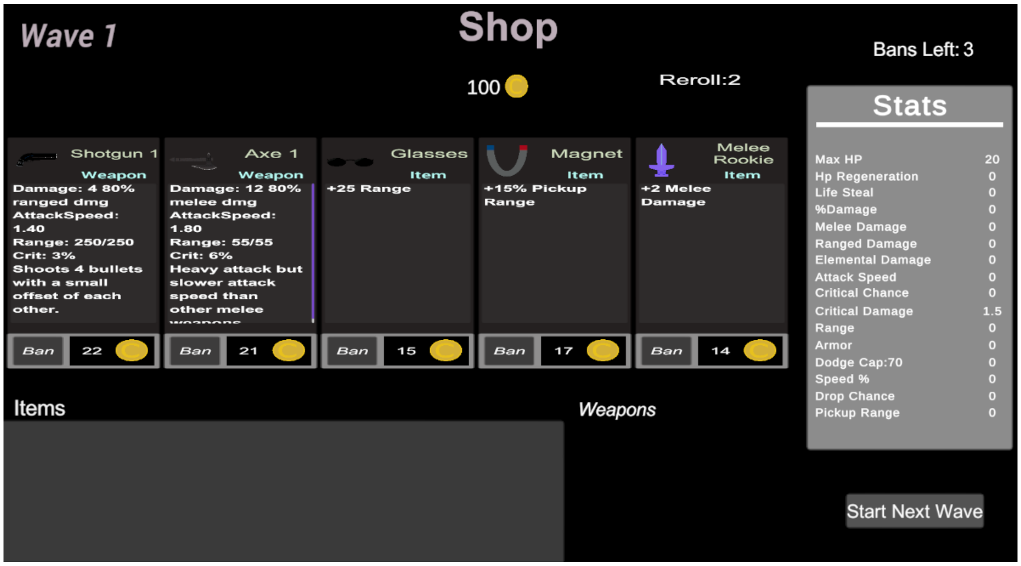
6.4 Coroutines

In Unity, a coroutine is a method that can pause execution and return continuing where it left off on the following frame. A coroutine allows us to spread a task over several tasks and that's why it is best suited for animations or a sequence of events over time. Coroutines aren't threads which means that all coroutines run on the main thread which can significantly increase the time spent on the main thread of the CPU.

In our project we use coroutines to change the alpha channel of the background image from transparent to full value (0 -> 1) over several frames. It is used also to find and destroy every enemy and their projectiles after finishing a wave resetting thus the map preparing for the next wave.

6.5 Shop

The shop is activated after each wave the player has survived. When the shop opens the player can buy as many items as he can afford which are dynamically and semi - randomly added to the pool to choose from with a total of 5 weapons/items. The distribution of the shop items is semi-randomly because the probability might be altered based on the item tags or the weapons currently equipped. There is a 10% bonus chance to have a weapon in shop that the player has equipped. If it fails to show up it can still be included in shop if it is chosen from the main item/weapon list since it isn't removed from there. In the first 3 waves the shop has 2 out of 5 slots with a weapon to help the player grow strong early and maxing out the weapons he can hold while the other 3 are guaranteed items.   From wave 3 till wave 7 the first slot is guaranteed to be a weapon while the other 4 slots are 65% to be an item and 35% to be a weapon. From there on, the 65/35 percent ratio continues until the end. Shop preview in the image below. (Image 28)



Shotgun 1
Weapon
Damage: 4 80% ranged dmg
AttackSpeed: 1.40
Range: 250/250
Crit: 3%
Shoots 4 bullets with a small offset of each other.

Axe 1
Weapon
Damage: 12 80% melee dmg
AttackSpeed: 1.80
Range: 55/55
Crit: 6%
Heavy attack but slower attack speed than other melee weapons

Glasses
Item
+25 Range

Magnet
Item
+15% Pickup Range

Melee Rookie
Item
+2 Melee Damage

Stats

Max HP	20
Hp Regeneration	0
Life Steal	0
%Damage	0
Melee Damage	0
Ranged Damage	0
Elemental Damage	0
Attack Speed	0
Critical Chance	0
Critical Damage	1.5
Range	0
Armor	0
Dodge Cap:70	0
Speed %	0
Drop Chance	0
Pickup Range	0

Image 28. Shop

Upon buying an item or weapon it is added to the player's loot. If it is new, it is added to his inventory and if it already exists which means that he has bought it previously it adds up increasing the counter of that item. If it is a weapon it is added to the Weapon Holder/Inventory. Now, if the player hasn't all the slots occupied with a weapon it is just added in an unoccupied slot. If the player has 6 weapons which is the max amount he can wield, then a method is called which checks if that specific item already exists in his weapon inventory and if it exists it is upgraded into the same weapon but 1 rank higher. In order to get this kind of upgrade the weapon rank bought has to be the same with the rank and name of one of the player's weapons, else he fails to buy it. All stats displayed in the shop are updated with the current values based on the player's stats if they have scalable values of course.

6.5.1 Rerolling and pricing

If the player dislikes the content of the shop and doesn't want to buy anything he has the option of rerolling it by spending money. There is no limit on how much he can reroll however, the price of rerolling increases each time it is used resetting to its default value of the wave after each wave. The default values and increase after each use are fixed and can't be changed at runtime as shown in the table below (Table 1).

Reroll prices

Wave	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Price	2	3	4	5	6	7	8	9	10	11	13	15	18	19	20	22	24	26	34
Increase	1	1	2	2	2	3	3	3	4	4	4	5	5	6	6	7	7	8	8

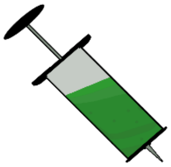
Table 1

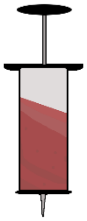
Moreover, there is a ban system where the player can erase a weapon or item he doesn’t like from the list until the rest of the run. Because there aren’t many items/weapons in the game he has only 3 bans and there isn’t a way to increase them. In case there are no more items with a specific rank, because the player has bought all available and/or has banished them, then a random item with 1 rank lower is chosen.

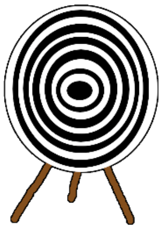
6.5.2 Weapons and items

There are 21 different Items and 7 weapons totally with 6 different attack patterns. These Items and weapons are listed below:

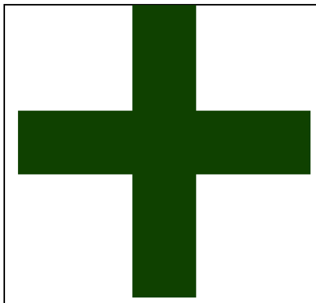
Items

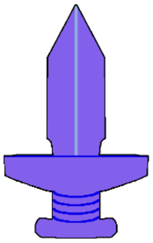
	Name: Antidote	
	Rank: 1	Type: Item
	Upon purchase boosts player’s Max Health points by 5 points, health-regen by 2 points and damage by -2%.	

	Name: Doping	
	Rank: 1	Type: Item
	Upon purchase boosts player’s damage by 8% but lowers max health and health regen by 1 point.	

	Name: Archery	
	Rank: 2	Type: Item
	Upon purchase boosts player’s critical chance by 5% and critical damage by 25%.	

	Name: Glasses	
	Rank: 1	Type: Item
	Upon purchase boosts player’s range by 25 points.	

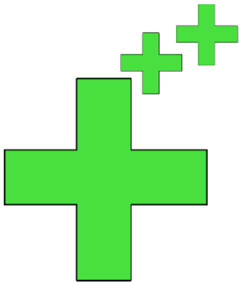
	Name: Health Boost	
	Rank: 1	Type: Item
	Upon purchase boosts player’s max health points by 1 point.	

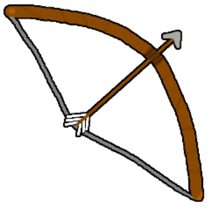
	Name: Melee Rookie	
	Rank: 1	Type: Item
	Upon purchase boosts player’s melee damage by 2 points.	

	Name: Coffee Break	
	Rank: 1	Type: Item
	Upon purchase boosts player’s attack speed by 10% but lowers his damage by 2%.	

	Name: VIP	
	Rank: 1	Type: Item
	Upon purchase reduces item and weapon cost in shop by 5% with each buy.	

	Name: Bomb	
	Rank: 1	Type: Item
	Upon purchase boosts player’s damage by 5% but reduces his melee and ranged damage by 1 point.	

	Name: Leukocytes	
	Rank: 1	Type: Item
	Upon purchase boosts player’s health-regen by 4 points but reduces max health by 2 points.	

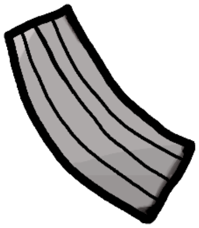
<table>
<tr><td rowspan="3"></td><td colspan="2">Name: Bow</td></tr>
<tr><td>Rank: 1</td><td>Type: Item</td></tr>
<tr><td colspan="2">Upon purchase boosts player’s ranged damage by 2 points.</td></tr>
</table>

<table>
<tr><td rowspan="3"> </td><td colspan="2">Name: Magnet</td></tr>
<tr><td>Rank: 1</td><td>Type: Item</td></tr>
<tr><td colspan="2">Upon purchase boosts player’s pick-up range by 15%.</td></tr>
</table>

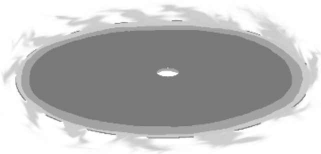
<table>
<tr><td rowspan="3"> </td><td colspan="2">Name: Tree</td></tr>
<tr><td>Rank: 1</td><td>Type: Item</td></tr>
<tr><td colspan="2">Upon purchase reduces the cooldown of tree spawn by 0.5 seconds for each stack of tree. Default cooldown is 10 seconds and it has a fixed chance to spawn.</td></tr>
</table>

<table>
<tr><td rowspan="3"> </td><td colspan="2">Name: Tree Chopper</td></tr>
<tr><td>Rank: 1</td><td>Type: Item</td></tr>
<tr><td colspan="2">Upon purchase sets Trees to 1 Health point to die in 1 hit. Unique item. After purchase it is removed from the list in shop for the rest of the run.</td></tr>
</table>

	Name: Shoes	
	Rank: 2	Type: Item
	Upon purchase boosts player’s speed by 3%, dodge by 8% but reduces armor by 1 point.	

	Name: Magazine Size +	
	Rank: 2	Type: Item
	Upon purchase boosts player’s attack speed by 12% ranged damage by 2 points and reduces damage by 3%.	

	Name: Melee Mastery	
	Rank: 2	Type: Item
	Upon purchase boosts player’s melee damage by 7 points and reduces his ranged damage by 3 points.	

	Name: Razor Disk	
	Rank: 2	Type: Item
	Upon purchase spawns a razor disk that orbits the player dealing 3 plus 60% of melee damage to enemies it collides with. Max purchasable razor disks is capped at 8 stacks.	

	Name: Platinum Shield	
	Rank: 3	Type: Item
	Upon purchase boosts player’s defense by 5 points and reduces his movement speed by 4%.	

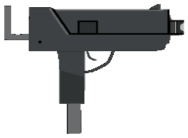
	Name: Spiked Ball	
	Rank: 3	Type: Item
	Upon purchase, the player has a chance to spawn a spiked ball at his current location every fixed amount of time that does a fixed amount of damage to enemies that touch it. Damage to enemies is 15 and can damage enemies a total of 20 times. After that the spiked ball is destroyed.	

	Name: Body Suit	
	Rank: 4	Type: Item
	Upon purchase boosts player’s max health by 2 points, health-regen by 1 point, life steal by 1, damage by 10%, melee and ranged damage by 2, elemental damage by 4, attack speed by 20%, speed by 3% but reduces range by 10 points and armor by 3 points.	

Weapons

	Name: Sword	Ranks: 1/2/3/4
	Damage: 5/12/25/42 with 50/70/90/105% melee damage amplifier	
	Attack Speed: 1.5/1.3/1.18/1s	
Range: 60/65/85/95 units.		
Critical Chance: 3/3/3/3%		
Description: Throws a spear towards the nearest enemy hitting and damaging all in its path. Can do damage while returning to player too.		

	Name: Pistol	Ranks: 1/2/3/4
	Damage: 10/20/30/50 with 100/100/100/100% ranged damage amplifier	
	Attack Speed: 1.2/1.15/1.05/0.8s	
Range: 400/400/400/400 units.		
Critical Chance: 3/3/3/3%		
Description: A fast and well scaling pistol.		

<table>
<tr><td rowspan="3"></td><td>Name: Uzi</td><td>Ranks: 1/2/3/4</td></tr>
<tr><td colspan="2">Damage: 3/3/6/6 with 50/65/80/90% ranged damage amplifier</td></tr>
<tr><td colspan="2">Attack Speed: 0.17/0.15/0.1/0.1s</td></tr>
<tr><td colspan="3">Range: 300/350/400/400 units.</td></tr>
<tr><td colspan="3">Critical Chance: 1/1/2/4%</td></tr>
<tr><td colspan="3">Description: A very fast weapon with low base damage and scale.</td></tr>
</table>

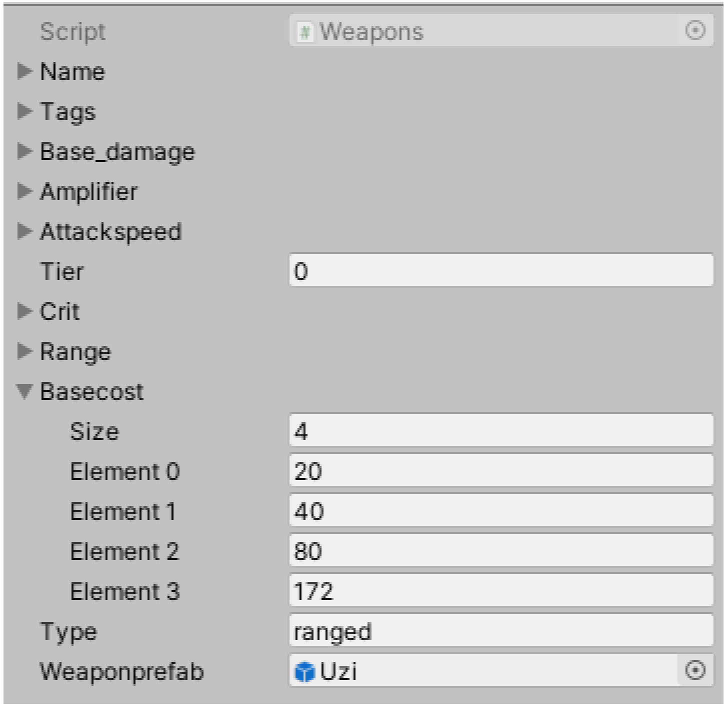
<table>
<tr><td rowspan="3"> </td><td>Name: Shotgun</td><td>Ranks: 1/2/3/4</td></tr>
<tr><td colspan="2">Damage: 4/6/12/18 with 80/85/90/100% ranged damage amplifier</td></tr>
<tr><td colspan="2">Attack Speed: 1.4/1.28/1.2/1.1s</td></tr>
<tr><td colspan="3">Range: 250/250/300/350 units.</td></tr>
<tr><td colspan="3">Critical Chance: 3/3/3/3%</td></tr>
<tr><td colspan="3">Description: Shoots 4 bullets in a cone in front with a slight angle each bullet.</td></tr>
</table>

<table>
<tr><td rowspan="3"> </td><td>Name: Spear</td><td>Ranks: 1/2/3/4</td></tr>
<tr><td colspan="2">Damage: 10/20/35/48 with 80/90/100/110% melee damage amplifier</td></tr>
<tr><td colspan="2">Attack Speed: 1.4/1.26/1.15/0.92s</td></tr>
<tr><td colspan="3">Range: 200/250/300/300 units.</td></tr>
<tr><td colspan="3">Critical Chance: 3/3/6/6%</td></tr>
<tr><td colspan="3">Description: Throws a spear towards the nearest enemy hitting and damaging all in its path. Can do damage while returning to player too.</td></tr>
</table>

<table>
<tr><td rowspan="3"> </td><td>Name: Axe</td><td>Ranks: 1/2/3/4</td></tr>
<tr><td colspan="2">Damage: 12/23/33/50 with 80/90/100/120% melee damage amplifier</td></tr>
<tr><td colspan="2">Attack Speed: 1.8/1.6/1.43/1.22s</td></tr>
<tr><td colspan="3">Range: 55/60/65/100 units.</td></tr>
<tr><td colspan="3">Critical Chance: 6/6/10/10%</td></tr>
<tr><td colspan="3">Description: Melee weapon that swings on the nearest enemy damaging all enemies it hits. Strong and high damage but low attack speed</td></tr>
</table>

<table>
<tr><td rowspan="3"> </td><td>Name: Hatchet</td><td>Ranks: 1/2/3/4</td></tr>
<tr><td colspan="2">Damage: 7/13/22/33 with 60/70/80/100% melee damage amplifier</td></tr>
<tr><td colspan="2">Attack Speed: 1.4/4.32/1.14/0.95s</td></tr>
<tr><td colspan="3">Range: 60/60/80/80 units.</td></tr>
<tr><td colspan="3">Critical Chance: 4/4/4/4%</td></tr>
<tr><td colspan="3">Description: Swings on nearest enemy damaging all it hits. Has medium damage and scaling but its attack speed is pretty low.</td></tr>
</table>

The data needed for every one of them is stored in a Scriptable Object just like in enemies. Again, we need to have the same variables for every same copy and here we don’t even have a Health variable that needs to change so every field is shared. First, let’s see the fields of the Weapons Scriptable Object which is included in the main Item SO which is called “ShopItems” as it is used to initialize the shop, and explain what they are needed for (Image 29).



Script	Weapons
▶ Name	
▶ Tags	
▶ Base_damage	
▶ Amplifier	
▶ Attackspeed	
Tier	0
▶ Crit	
▶ Range	
▼ Basecost	
Size	4
Element 0	20
Element 1	40
Element 2	80
Element 3	172
Type	ranged
Weaponprefab	Uzi

Image 29. Weapon scriptable object

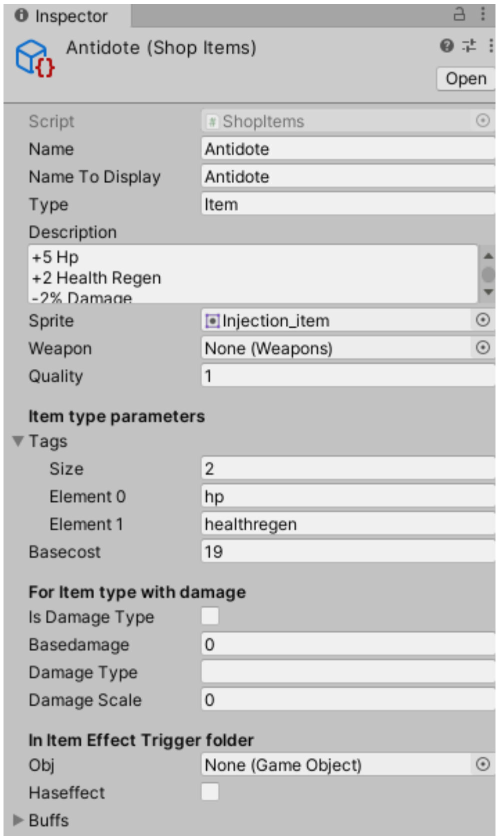
Weapons in shop can have 4 ranks with a probability chance that was calculated earlier and is calculated at. The ranks or Tiers of the items and weapons are represented with a color in background of the image. Black background means its rank is 1, blue background is rank 2, purple background is rank 3 and red-orange background is rank 4 which is the highest and rarest rank to find. In all RPGs with a rank system on items these are the standard colors that is why these specific colors were chosen here too. The arrow next to most fields is to insert an array size, any size is acceptable but, in our case, we need 4 (as many as our ranks). All values in the first slot of the array or as the above image shows all “Element 0” are for the rank 1 weapon, “Element 1” is for rank 2 and so on.

The fields are listed and explained below for the weapon SO (Image 29 above).

- Name: name of the weapon
- Tags: Each weapon and item have tags to increase the probability of finding items or weapons with same tags
- Base Damage: The starting damage player does with each weapon when he has no stats increase

- Amplifier: Percentage boost in damage from character stats
- Attack speed: Time in seconds till attacking
- Crit: Chance to Critically strike
- Tier: Default rank (rank 1)
- Range: how far the weapon becomes aware of enemies
- Base Cost: Price of weapon increasing with each wave past
- Type: If the weapon is melee, range or elemental based
- Weapon Prefab: Contains the Game Object of the weapon

Now for the Shop Items SO or "ShopItem", the number of fields it has are more than that of the weapon SO. Unlike the SO mentioned earlier in the document this SO has some empty fields known as "null" in programming languages because they are not essential for every item. Let's see the ShopItem SO of the image below and analyze its fields like with the weapon SO (Image 30). The image below is the SO of Antidote item.

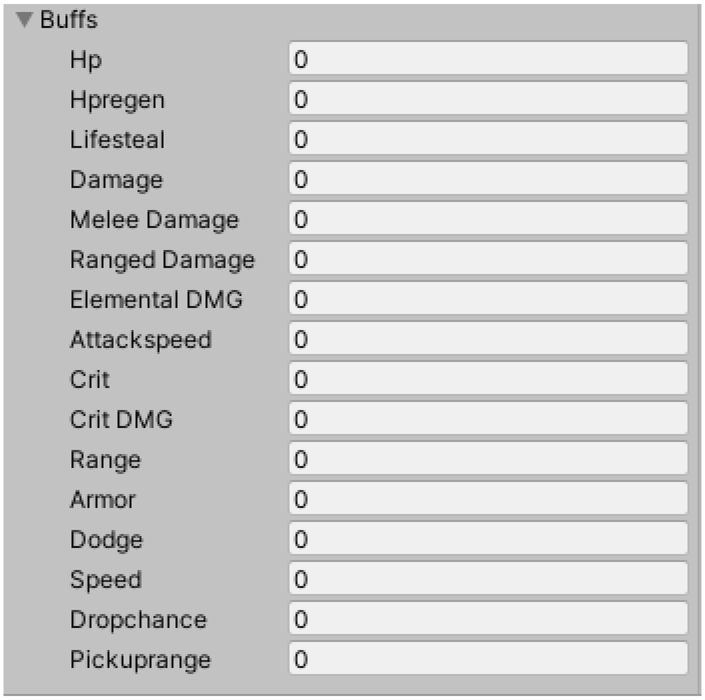


Antidote (Shop Items)

Script	ShopItems
Name	Antidote
Name To Display	Antidote
Type	Item
Description	+5 Hp +2 Health Regen -2% Damage
Sprite	Injection_item
Weapon	None (Weapons)
Quality	1
Item type parameters	
Tags	
Size	2
Element 0	hp
Element 1	healthregen
Basecost	19
For Item type with damage	
Is Damage Type	☐
Basedamage	0
Damage Type	
Damage Scale	0
In Item Effect Trigger folder	
Obj	None (Game Object)
Haseffect	☐
Buffs	

Image 30. Antidote shop item scriptable object

- Name: Item/weapon name
- Name to Display: Displayed in shop. It is needed because if in scripts all words are case sensitive
- Type: Can be Item or Weapon type
- Description: Details about the item. Displayed in shop too. We will see it later when we talk more about the shop
- Sprite: Image of the item to be displayed
- Weapon: This is a field only for Weapons. If it isn't a weapon this field is empty otherwise the data of the weapon SO goes here
- Quality: The rank of the item/weapon ranging from 0-3 instead of 1-4 because in programming languages the initial value of arrays and lists is 0
- Tags: As explained above it is needed to have a chance of finding same tagged items
- Base Cost: Initial price in the start of the game scaling up per wave
- Is Damage Type: This is a Boolean field which can be true or false (marked or unmarked) and is used to check if the item itself does damage or not
- Base Damage: For damaging types only. It is the base damage for hurting the enemies
- Damage Type: For damaging types only. Can be melee, ranged or elemental to boost the damage value according to player stats
- Damage Scale: For damaging types only. It is the percentage to boost the damage output of the item based on the damage type's stat
- Obj: Short for Object. Here comes any prefab that needs to be instantiated when the item is bought. For instance, Razor Disk has a prefab here
- Has Effect: It is another Boolean value (true or false) to check if an item gives only stats to the player or if it has a unique effect to be activated
- Buffs: These are the stats that can be changed through items when bought. Each item has its own changes in these stats. The stats that can be changed are displayed in the image below (Image 31)



▼ Buffs

Hp	0
Hpregen	0
Lifesteal	0
Damage	0
Melee Damage	0
Ranged Damage	0
Elemental DMG	0
Attackspeed	0
Crit	0
Crit DMG	0
Range	0
Armor	0
Dodge	0
Speed	0
Dropchance	0
Pickuprange	0

Image 31. Item's stats changes

If we want to add more stats, we simply add that field in the Buffs script and we are done already. In the script responsible for updating the player when an item/weapon is bought, we first check if the item has an effect to add it to the player and if the corresponding field is checked then it searches the item and applies the effect, else we just add or subtract the stats to the player.

As we already mentioned, the holding limit of weapons is 6 and an upgrade method is called whenever he tries to buy a weapon while at max weapons. Apart from this system, the player can use the left click of the mouse on a weapon in his weapon inventory to open a small window with 3 options. Option 1 is "Merge" which checks if the player can upgrade that weapon into 1 rank higher. To be able to upgrade a weapon he needs to have 2 same weapons and same rank including the clicked weapon. Option 2 is "Destroy" which removes the weapon from the player. Doing this doesn't give anything back to the player (no money or stats earned). Option 3 is "Cancel" which closes the window that opened. It is used when the player doesn't want to take any actions. Below there is an image before and after pressing "Merge" (Image 32) and after pressing "Destroy" (Image 33).

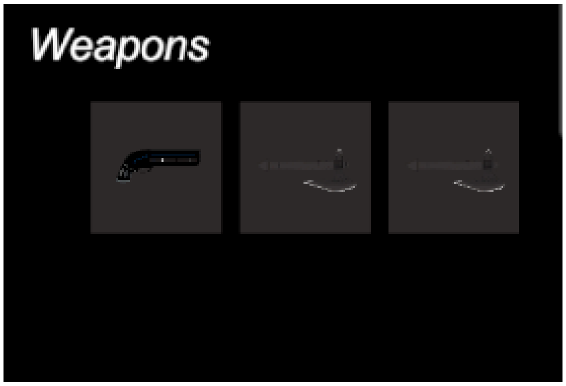


Image 32a. Before “Merge”

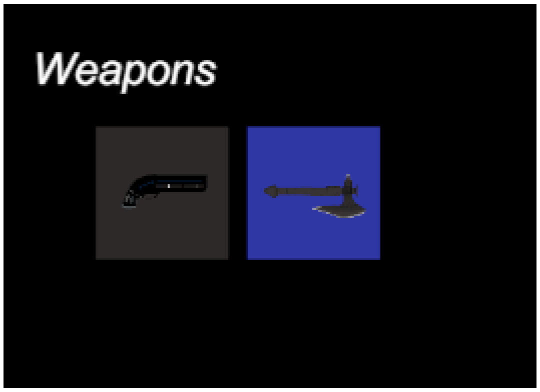


Image 32b. After “Merge”

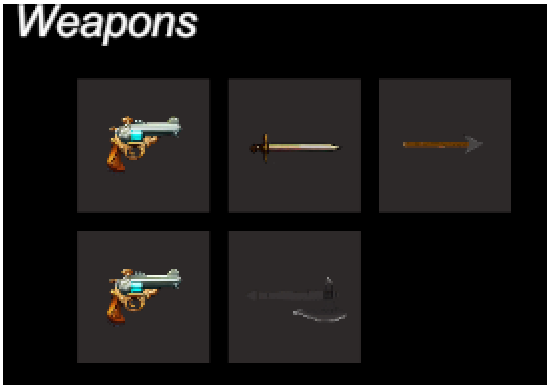


Image 33a. Before “Destroy”

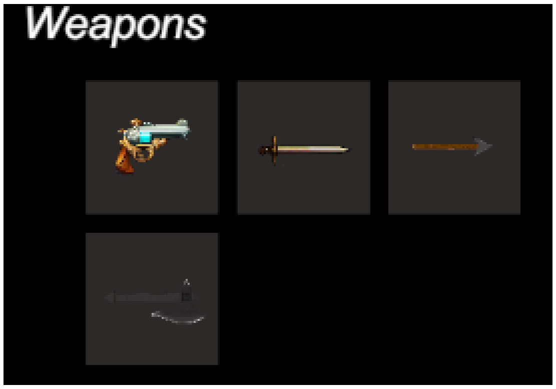


Image 33b. After “Destroy”

The popup window with the 3 Options on pressing left mouse button is shown in the image below (Image 34).

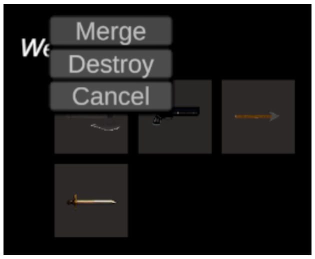


Image 34. Weapon options

The inventory changes can be seen in the image below (Image 35) when purchasing new items or stacking them up.

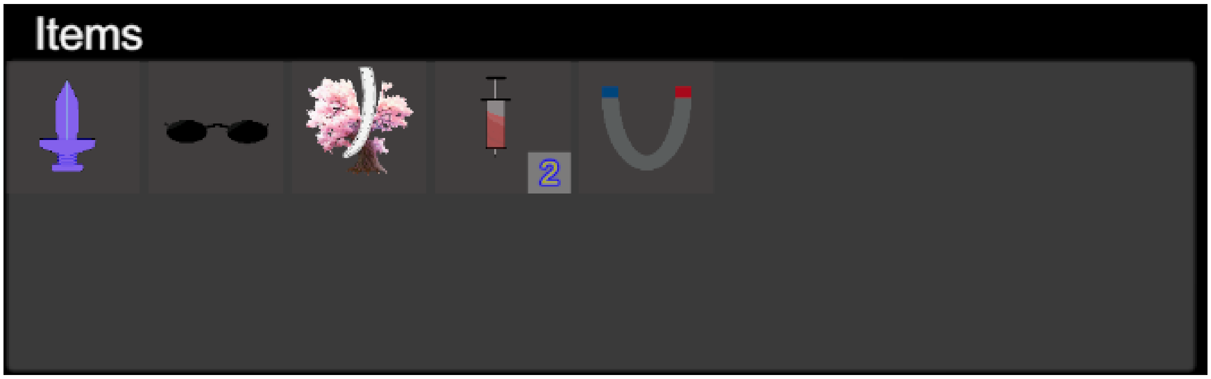


Image 35. Item's Inventory

6.5.3 Stats

The stats displayed in the shop are the same as the player's stats. Every value is shown and updated on each successful purchase. If an item is bought multiple times, then the stats change equal to the stat changes the item has times the number of that item in the inventory. It is worth noting that the damage output of each weapon is calculated based on the following equations: $Damage = base\ weapon\ damage + (weapon\ type\ boost * amplifier * 0.01)$ $Total\ Damage = damage * percentage\ damage\ boost * 0.01$ and if it is a critical hit the damage becomes: $Final\ Damage = Total\ Damage * Critical\ Damage$. So first we add the Percentage damage and then the critical damage multiplier.

To continue to next wave if the player has finished shopping, he simply needs to press the "Start Next Wave" button and starts fighting again the enemies only that this time he is stronger than before.

6.6 End criteria

There are 2 ways to end the run thus the game either by dying (reducing player health to 0) or by defeating/surviving wave 20. When the player dies, the controls are disabled while a smooth transition is playing blackening the screen. The new window has 3 buttons, so 3 options. Either go back to the portal map or go to main menu or quit the game and close the application. On winning the run another window pops up with the same 3 options plus indicating that an upgrade token has been received to use in the forger. The corresponding images for each of the above situations are shown below (Image 36 and 37)

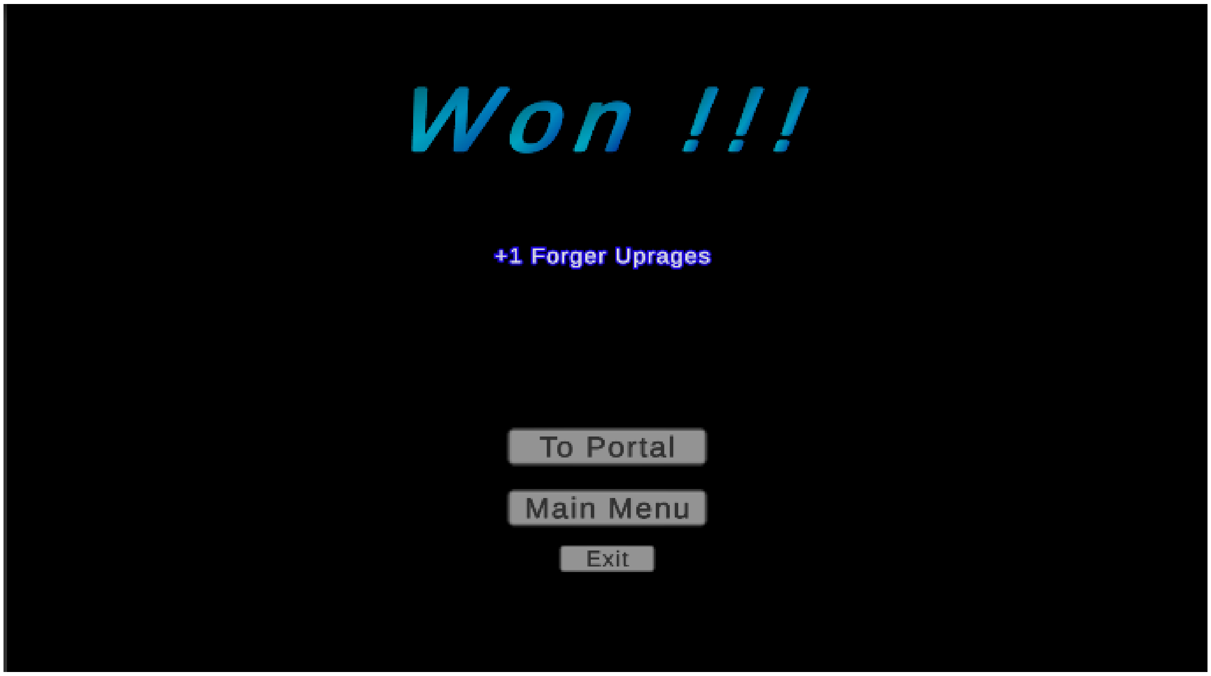


Image 36. Won Run

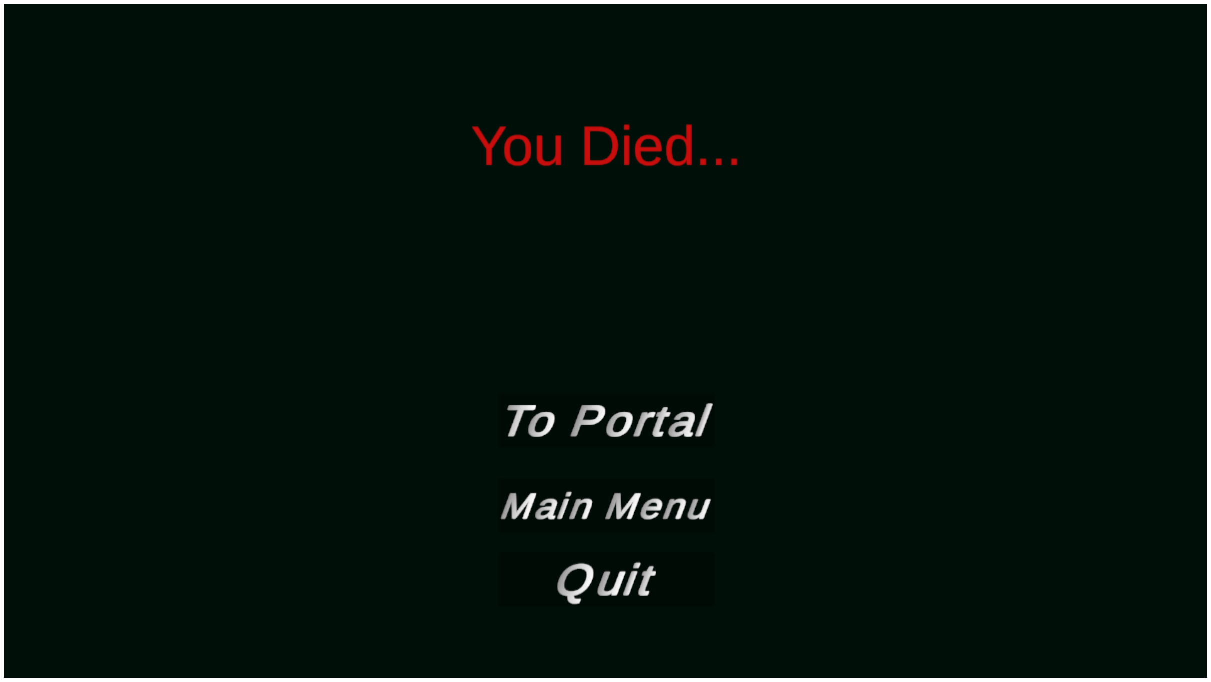


Image 37. Died Window

7. SECURITY

Security in games is very important for protecting sensitive information and providing a fair gaming experience. Some of the challenges are preventing hackers from stealing credit card numbers, cheating or directing users to malicious sites. Because this is an offline single player game, we need only to focus on preventing people to cheat or edit the saved values. To prevent people from this action, all save data has been encrypted before saving into Unity's Players Pref file which can be loaded any time. Player Pref is a file in the Unity's building project which allows developers to save and load data with a specific format. To save a value string, integer or float we need to provide a key combined with that value so that we can retrieve it later using the key. To load back the data, we simply use that key and if it exists in the Player Pref file it returns the value, else it returns nothing.

There are many ways to encrypt data into a file or modify the saving data so that it is not readable. In our case, the implemented class AES established by the U.S National Institute of Standards and Technology (NIST) in 2001 was used from C# libraries. AES stands for Advanced Encryption Standard and it derives from the Symmetric Algorithm class which use a chaining mode called cipher block chaining (CBC), which require a key and an initialization vector (IV) to perform cryptographic transformation on data. Because an instance of the AES class is created each time, we want to encrypt a text (string), after closing the Application that instance is lost and we can't retrieve the data that was encrypted into our file because we get a new key and IV the next time, we try to decrypt the text. To overcome this problem a constant string is needed for both the key and the IV value. The string can contain any number, character and/or symbol and the longer it is the more secure our data becomes.

The base AES encryption method takes an array of bytes as a parameter to be encrypted, but because we want to save text only inside the Player Pref, it is modified to take in a string parameter instead of the byte array. To encrypt the given string (text), we need to convert the string into a byte array. Afterwards, we create an AES instance with our constant key and IV values so that we can decrypt it anytime. Based on the key and IV we create an encryptor, which is forwarded into a memory stream and then a crypto stream where it is encrypted. After the encryption ends the streams are closed and the new encrypted string is returned to be saved. The role of the memory stream is to store the data it's given in memory, in order to retrieve it later.

For the decryption we first have to encrypt the Player Pref's key so that we can find its value if it exists inside of the file then send the retrieved string to the descriptor. The descriptor is identical to the encryptor taking in a string parameter, converting the string to a byte array, then creates an instance of the AES class with our constant key and IV values. After that, like in the encryptor we created an encryptor, here instead we create a descriptor with our key and IV, then we open a memory stream into a crypto stream where it is decrypted and returned through the memory stream to us back in a string format.

8. CONCLUSION AND FUTURE UPDATES

In this section we will discuss the conclusions of this little journey of programming as well as some future updates, adding new mechanics and content.

8.1 Conclusions

Keeping the section where we talked about some similar games in mind, we could certainly say that the job of a game developer is not an easy one. Due to constant new game releases and changes to the old ones, a game developer can never rest and leave a project for several months without any changes because the loss of his player base will decrease drastically, especially if the video game doesn't have any replay-ability or end-game content. Despite all the difficulties faced during this project, it came to an end and the result is a finished video game that could be published anytime in a gaming platform or a website in order for people to download install and play it even tomorrow. Publishing would also help with some useful statistics like what specifications are needed for a computer to have a smooth gaming experience without having fps drops or crashes. The game will be available on a charity site called itch.io as soon as I can and the new pricing policy of Unity is explained.

In the future gaming industries will continue to grow says Cam Shea in his article (2020) and that gaming experience will significantly change with the new technologies, new GPU's and hardware such as consoles that will be released. Virtual reality, AI implementation and streaming technology will improve even more in the future which will open new paths and ideas for gaming experiences. Lastly, a technology of pixel rendering with procedural generation that is in early stage of development will make games outperform movies in terms of performance. The final game project can be found in my repository on GitHub in the following link: Video Game.

8.2 Future updates

This video game is still in an alpha phase. This means that it has room for improvements and optimization. Some of the future plans are listed below.

Elemental reaction system

This will be a new system implementing elemental type damage. Currently elemental damage is like all the other damaging types but when this system is introduced it will change. There will be different types of elemental effects (primary fire-water-thunder-wind), each having their own effects and impact on enemies. Also, when an enemy is afflicted with an elemental status, he is on fire as an example, using another elemental effect on that same target will cause a reaction based on the two elemental types that caused this reaction.

Character selection

Like in many video games, the player can choose a character he likes from a variety of selections, or customize him. The main concept of adding more characters to this game is to create new selectable characters for the player, where each one of these have a unique skill to activate.

Achievements

Even though the game has already some achievements, adding new achievements both simple and more complex ones is included in the future plan.

New modes

Another feature that will be added is mode selection. The only mode that the player can choose now is the survival mode of these specific waves, but when the new modes are released, the player can choose endless mode or fighting in randomly generated maps too. Moreover, there might be an option to change the play style from 2D to 3D, expanding the playing possibilities.

More options and design changes

A modern video game has to provide lot of options to modify the gaming experience or the interactions with the UI as well as a beautiful design of Menus, characters, enemies and assets in general. This is very time consuming that's why it has to be done step by step.

More items and weapons

With the basic logic already done, and while keeping the code clean and easily readable/editable, it is very easy to add more items and Weapons anytime into the game. The most difficult part would be creating the sprites to be displayed in the shop and on the map but apart from that it is not a challenge.

9. UNITY'S NEW POLICY

On September 13th in 2023 Unity announced a new pricing policy which is very unclear and an explanation must be given. From 1st January 2024, Unity will charge for every copy of the game after reaching a threshold of 200,000 (thousand) dollar revenue in the last 24 months and the lifetime installs are more than 200,000 (thousand), any developer that is using Unity Personal or Student version. For Unity Pro and Enterprise, the fee still applies together with the yearly subscription fee, but the added fee is reduced per copy.

Unity doesn't clarify what is considered a new install causing huge panic to developers that have used and are using Unity's Game Engine. A simple example is what happens if the same person downloads the game, uninstalls it and then re downloads it. Will it count as a single install or not. Also, what will happen with all the free games or demos in the market, do they have to pay this fee even though they don't have a profit? Lastly and the most serious concern is what will happen to the developer if the game gets cracked releasing countless of copies in "pirate" website that each person in the world can freely download from the internet. Will these installs count towards the total install count according to unity’s new policy or is there a system to prevent these accidents.

These are all serious questions that Unity has to answer as soon as possible so that every developer can take action if needed. Because of the above statements, the future plans are being stalled till Unity provides us with more information about this new policy and answers these important statements.

References

Special thanks to the websites that provided useful information or services that were used in the document or in this project and to the other game developers that made free game assets for everyone to use.

Player sprite retrieved from: https://www.gameart2d.com/ninja-girl---free-sprites.html

Diagram editor/creator: https://app.diagrams.net/

Achievement Info retrieved from: https://en.wikipedia.org/wiki/Achievement_(video_games)

Song playing in game retrieved from: Free Music Tracks For Games | Audio Music | Unity Asset Store

Sound Effects retrieved from: https://assetstore.unity.com/packages/audio/sound-fx/rpg-essentials-sound-effects-free-227708

History of video games retrieved from: https://en.wikipedia.org/wiki/History_of_video_games

Similar game data retrieved from: https://steamdb.info/

Photoshop Wiki retrieved from: https://en.wikipedia.org/wiki/Adobe_Photoshop

Dr. Libby (2018). Nutritional biochemist, author and speaker. Side effects of fast living retrieved from: The Side-Effects of Fast Living - Dr Libby

Daniel Doan (2017). Why are modern games so popular retrieved from: https://gamedevlibrary.com/gamedev-thoughts-why-are-modern-roguelike-games-so-popular-53c1f5a05485

Peter Finn (2017). Video game analyst. Importance of achievements in video games retrieved from: Why Achievements Are More Important Than You Think (withaterriblefate.com)

Steve Russel, Martin Graetz, Wayne Wiitanen (1962). MIT students that developed the game Spacewar!

Thomas and John Knoll (1987). Thomas Knoll created a program named Display, a program to display grayscale images, which caught John's (Thomas's brother) attention and worked together on this project renaming it after to Photoshop

Ton Roosendaal (1994). Roosendaal is a Dutch art director and a self-taught software developer who started writing his first lines of code for Blender in 1994

Anders Hejlsberg (2000). He is a Danish software engineer who co-designed several programming languages and developed C# programming language

Ecma (1961). ECMA-334. It is a nonprofit standards organization for information and communication systems.

ISO/IEC 23270 (1947). It's an international standard development organization composed of representatives from the national standard organizations of member of member countries. The organization develops and publishes standardization in all technical and nontechnical fields other than electrical and electronic engineering, which is handled by the IEC (Electrical electrotechnical commission)

David Michael and Sandie Chen (2005). Serious games are games that educate, train and inform. Serious games retrieved from: Serious Games: Games That Educate, Train, and Inform | Semantic Scholar

Ila Adhikari, Filip Poutintsev (2021). Advantages and disadvantages of video games and conclusion whether they are good or bad

U.S National Institute of Standards and Technology NIST (2001). Established AES encryption protocol which is used in wifi connections, VPNs and many more to encrypt texts

Ken Allsop (2023). 2022 indie game remains near flawless at 200.000 Steam reviews. Popular indie game retrieved from: https://www.pcgamesn.com/vampire-survivors/steam-reviews

Cam Shea (2020). Professional writer for games. Article of what games might be like in 2030 and how gaming technology will improve

Dr. Michael Breus (2023). Clinical psychologist and sleep medicine expert. How stress can impact our sleep retrieved from: https://sleepdoctor.com/mental-health/stress-and-sleep/

Hansa Bhargava (2021). Physician, Author CBCT instructor CMO Medscape. Survey results of how games impact on mental health retrieved from: https://www.forbes.com/sites/hansabhargava/2021/10/04/hiding-in-plain-sight-the-impact-of-gaming-on-mental-health/

WHO (World Health Organization, 2022). Research on how much people's anxiety and stress levels increased after the start of pandemic retrieved from: https://www.who.int/news/item/02-03-2022-covid-19-pandemic-triggers-25-increase-in-prevalence-of-anxiety-and-depression-worldwide

Lela Moore (2023). Mental health analysis and how it can change our behavior. Medically reviewed by psychologist Marney A. White. Mental health analysis retrieved from:

Good Mental Health: What is it? (psychcentral.com)