

**Bullet Ballet**  
Design and development of a 2D  
shoot'em up and roguelite hybrid game



Andriy Kokhan Shpakova

Final Degree Work  
Bachelor's Degree in  
Video Game Design and Development  
Universitat Jaume I

May 16, 2023

Supervised by: Emilio Bueso Aparici



To my mother Tatyana, for supporting me in every single thing I do and always being there in times of need.





# ACKNOWLEDGMENTS

First of all, I would like to thank my Final Degree Work supervisor, Emilio Bueso Aparici for his guidance along this project.

I also would like to thank my mother for supporting me in each step of this journey with all of her might and never questioning my dreams.

And of course, thank you to all of my friends who helped me test the project and gave me tons of emotional support.

Last but not least, I also would like to thank Sergio Barrachina Mir and José Vte. Martí Avilés for their inspiring [LaTeX template for writing the Final Degree Work report](#), which I have used as a starting point in writing this report.



# ABSTRACT

This document presents the project report of the Video Game Design and Development Degree Final project by Andriy Kokhan.

Bullet Ballet is a 2D top-down shoot'em up game that seeks innovation by mixing this genre with rogue-like elements present in the emerging "survivors-like" sub-genre. The objective is to make the player live a fun and engaging power fantasy with lots of freedom in terms of character building.

Moreover, the game is presented in a cute and lighthearted graphical cartoon style aiming to be enjoyed by video game players of all ages and explores appealing visual effects such as particle and trail systems.



# CONTENTS

<b>Contents</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Work Motivation . . . . .	1
1.2 Objectives . . . . .	2
1.3 Environment and Initial State . . . . .	2
<b>2 Planning and resources evaluation</b>	<b>5</b>
2.1 Planning . . . . .	5
2.2 Resource Evaluation . . . . .	9
<b>3 System Analysis and Design</b>	<b>11</b>
3.1 Artistic Design . . . . .	11
3.2 Game Design . . . . .	14
3.3 Interface Design . . . . .	15
3.4 Sound Design . . . . .	20
3.5 Requirement Analysis . . . . .	20
3.6 System Design . . . . .	21
<b>4 Game System Designs</b>	<b>27</b>
4.1 Genres and Inspirations . . . . .	27
4.2 Character Statistics . . . . .	28
4.3 Projectile Statistics . . . . .	30
4.4 Enemy Statistics . . . . .	31
4.5 Bullets . . . . .	32
4.6 Enemy Types and Behaviour . . . . .	51
4.7 Containers and Items . . . . .	52
<b>5 Work Development and Results</b>	<b>55</b>
5.1 Work Development . . . . .	55
5.2 Results . . . . .	56
5.3 Beta test . . . . .	56
<b>6 Conclusions and Future Work</b>	<b>59</b>

6.1	Conclusions . . . . .	59
6.2	Future work . . . . .	60
	<b>Bibliography</b>	<b>61</b>
	<b>A Appendix</b>	<b>63</b>
	<b>List of Figures</b>	<b>63</b>
	<b>List of Tables</b>	<b>64</b>
	<b>B Source code</b>	<b>65</b>

# INTRODUCTION

## Contents

---

1.1	Work Motivation . . . . .	1
1.2	Objectives . . . . .	2
1.3	Environment and Initial State . . . . .	2

---

This chapter covers the motivations that led into taking on this project as well as the initial ideas and expectations I had in mind and how they ended up evolving along the way of the project development [7].

## 1.1 Work Motivation

Approximately one year ago, I discovered a small indie video game called *Vampire Survivors* while it was on its early access phase. It was cheap, not so good-looking and very boring at first glance. However its gameplay ended up being really addictive, the developers always listened to the community with regular updates and in the end it even gave birth to a whole new game sub-genre: the "*survivors-like*".

In summary, *Vampire Survivors* is an action rogue-like game where the player's goal is to survive hordes of enemies for 30 minutes. As the player defeats enemies and picks up experience crystals, the character will level up. Upon leveling up, the player can pick a weapon or passive item out of a random pool of three choices (carrying a maximum of 6 of each). This was the most fun part for me, trying out different weapon combinations and coming up with new cool builds during each run.

In this game, the character attacks automatically, so the only interaction that the player has over the game is moving the character and decision making. My love and passion for that game led me to start this project, a brand new game based on the foundations of *Vampire Survivors* but with a more interactive gameplay, new mechanics and a cozier art direction.

## 1.2 Objectives

The main goals that led to the development and gave shape to this project were:

- Design and develop a fun video game with great replayability value, where each game feels fresh and gives the player possibilities to try new fun strategies.
- Innovate in the way roguelite games are designed, taking inspiration from the emerging “survivors-like” roguelite game genre.
- Develop a balanced and challenging but satisfying game flow where the player can feel very powerful if he works towards his understanding of the game.
- Design a fitting aesthetic, interface and assets for the game, while leaning more towards a lighthearted style than a serious one.
- Explore and make visually pleasing and unique effects thanks to Unity’s particle system to make the game look lively and enhance the feedback of what’s happening on the screen.

## 1.3 Environment and Initial State

The initial goal for this project was the development of a 2D shoot’em up. However, I came up with the idea of fusing up this genre with the recent survivors-like rogue-like trend. What motivated my work was fusing up those two genres and expanding on the mechanics that they usually offer, creating a fresh and innovative approach to this kind of video games. After checking up with my supervisor, he greenlighted my ideas and agreed that this could be an interesting project.

My take on fusing this genres and giving them a new spin was decided even before the development phase started. I wanted to make a game where instead of different weapons, the player would have a set of bullets, each one firing different types of projectiles with different statistics, behaviours, effects, appearances and so on. On level up, the player would be able to get a new bullet which could be swapped for one of the bullets currently in their magazine or discarded. This would let the player build their magazine as they survive constant waves of enemies while thinking on what bullets suit better their current playstyle. Of course, this would be the main mechanic and the most fundamental system in the game.



---

The bullet system which the game revolves around would innovate both of the genres this game is based on. As for the shoot'em up genre, this system brings decision making to the table: the players would need to adapt to the bullets they find, thinking on which ones to pick in order to better synergize with their current loadout and improve their chances of survival. As for the survivors-like genre, this system would provide greater interactivity and a more engaging and dynamic gameplay, since the player needs to aim and shoot in order to defend itself from their foes.



# PLANNING AND RESOURCES EVALUATION

## Contents

---

2.1	Planning . . . . .	<b>5</b>
2.2	Resource Evaluation . . . . .	<b>9</b>

---

This chapter encapsulates the most technical part of the work. Here is where the project planning and resources used in its development process are discussed.

## 2.1 Planning

The next list shows and explains all of the planned tasks that have been taken care of during the development of this project. Note that most of the tasks weren't done one by one, this was a reiterative effort as some things had to be worked on along the way and others' priorities were constantly shifted. At the end of the list, there's a Gantt chart showcasing the tasks in a visual way (see Figure 2.1).

- **FINAL DEGREE PROJECT PREPARATION**

- **Technical proposal (2 hours):** this was the starting point of the project, the redaction of a document containing the initial ideas and main focus of the project and an overall list of goals.
- **Game Design Document (8 hours):** once the proposal was approved it was time to look at the bigger picture. The writing of this document had to be done in order to settle all of the important aspects of the game in a deeper and more detailed manner.

- **MAIN GAME PROGRAMMING**

- **Player and camera (10 hours):** the development and refinement of the player and camera controls in order to achieve the smoothest experience.
- **Enemies (25 hours):** the game needed different enemies with different behaviours to make it more interesting and diverse. The enemy statistics system had to be taken care of too, things like physics, health, status effects and more.
- **Bullet system (100 hours):** the game revolves around this mechanic, so a lot of time and thought had to be put in here. The hardest and most important thing that was done was the foundation that all of the bullets would be built upon, but later this system was expanded giving each bullet different behaviours and effects as the development progressed. This system further expanded by letting the player enhance the character base stats in case the dropped bullets were discard. All of this also needed a few screens and interactive interfaces so everything could work.
- **Game flow and stat scaling (12 hours):** things like the rate at which the enemies spawn, when and where each one of them starts appearing, their life, speed, etc. All of this had to be balanced and scaled designing formulas that change as the in-game time progresses.
- **Terrain generation (10 hours):** the game environment had to be infinite, so a simple but effective tile based world generation was needed.
- **Main menu (5 hours):** a starting screen that would greet the player and let the game be played.
- **End game screen (5 hours):** upon game over or game completion, the game needed a screen displaying information of interest from the last game, and give options to play again or get back to the main menu.
- **User Interface (5 hours):** stuff like a custom cross hair, life bars, experience bars, timer and etc. the most important information of the game in course need a clear representation on screen.

- **ART AND ANIMATIONS**

- **Player (6 hours):** the art design and animations of the playable character and his weapon.
- **Enemies (18 hours):** the different enemies that appear in the game had to be drawn and animated as well.
- **Bullets (25 hours):** the bullets involved in the game had to be designed as well as the projectile they would shoot when used.
- **Effects and particles (25 hours):** particles, trails, glowing materials and different effects for the bullets had to be designed to make the game more visually appealing and each bullet more unique. Some other effects like enemy hit effect, enemy death effect and gun smoke had to be done too.

- **Environment (6 hours):** the design of world tiles and environmental assets such as flowers or breakable boxes.
- **Menus and user interface (10 hours):** stuff like buttons, menu animations, health bars and so one had to be worked on.

- **GAME TESTING**

- **Balancing (10 hours):** tweaking the enemy and player statistics so the game doesn't feel too easy but neither too difficult, while rewarding players who take their time to understand the mechanics and hone their skills.
- **Bug fixes and refinements (10 hours):** there are always bugs to be fixed and elements that could be improved, and it's important to prioritize the most critical ones.

- **PRESENTATION PREPARATION**

- **Powerpoint (5 hours):** the powerpoint presentation to exhibit the whole project.
- **Project Memory (25 hours):** the time consumed in order to redact and put together this document.

The target time to develop this project was around 300 hours, however some parts took a little more time than expected, but the total time spent isn't that much higher. Here's a summary of all the work done in Gantt diagram format (see Figure 2.1).

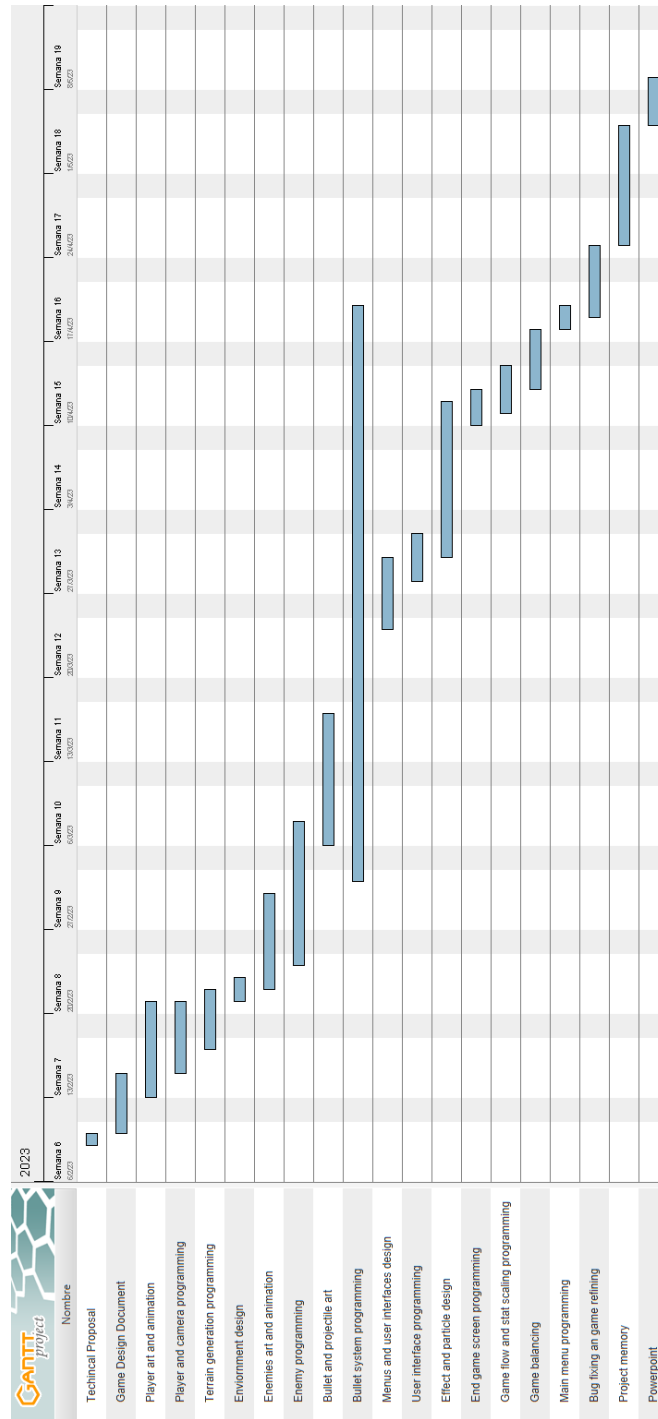


Figure 2.1: Gantt chart depicting the tasks that have been done (made with Gantt Project)

## 2.2 Resource Evaluation

The resources used to design and develop this project were the following:

- **Hardware**
  - **Operative system:** Windows 10 64-bits
  - **CPU:** AMD Ryzen 7 3700X
  - **GPU:** NVIDIA GeForce RTX 3060 Ti
  - **RAM:** 16 GB
  
- **Software**
  - **Unity 2021.3.17f1** - game engine used to create and work on the project [18].
  - **Visual Studio 2019** - the integrated development environment that comes with Unity. A powerful tool that integrates Unity's functions and helps to save a lot of time with its auto completion feature [14].
  - **Adobe Photoshop CS6** - one of the most powerful tools in terms of graphic design, used to draw and design all of the assets for the game [1].
  - **DaFont** - free to use text font archive, used to browse and pick appropriate text fonts for the project [6].
  - **Google Docs** - simple but effective web tools that have been used to make project proposal and game design documents [11].
  - **Gantt Project** - a desktop application used to create the Gantt chart depicting the tasks done [10].
  - **Overleaf** - online LaTeX document editor, which was used to produce this document [16].





# SYSTEM ANALYSIS AND DESIGN

## Contents

---

3.1	Artistic Design . . . . .	11
3.2	Game Design . . . . .	14
3.3	Interface Design . . . . .	15
3.4	Sound Design . . . . .	20
3.5	Requirement Analysis . . . . .	20
3.6	System Design . . . . .	21

---

This chapter analyzes the fundamental parts of Bullet Ballet’s design, being those related to the artistic, game flow, interface and sound design.

## 3.1 Artistic Design

As stated before, the artistic direction was clear from the start: the goal was to achieve a cute, friendly, cartoonish and fun aesthetic that could be enjoyed by players of all ages. Every single game asset was designed taking this philosophy into account.

- **2D assets:** Every single asset of the game, except some of the fonts used for interfaces and texts, were drawn using Adobe Photoshop. To achieve the cartoon look that the project was aiming for, important assets like characters and bullets were drawn with thick outlines. Furthermore, this gives the player visual feedback of what are the important characters or items on screen. Environmental objects that can’t be interacted in any way don’t have outlines at all. As for the color palettes, the game art style uses bright and colorful tones, in most cases using



Figure 3.1: Small sample of the artistic direction of Bullet Ballet.

more or less 3 to 5 different tones of each color to give depth and/or artificially simulate shadows and illumination (see Figure 3.1).

- **Particle effects:** In order to make the game more visually pleasing and give even more personality to each bullet, some of them use effects built thanks to the Unity particle system. These effects include but are not limited to smoke effects when firing heavy looking bullets (shotgun shells, for example), toxic bubbles when firing poisonous projectiles or some other that are not related to bullets, like for example the dust explosion that plays upon enemy defeat (see Figure 3.2).
- **Trail effects:** To further enhance the visual effects and each bullet personality, some projectiles, like for example, sniper-like bullets leave a trail behind as they travel through the space (see Figure 3.3).
- **Illumination effects:** Something that most of the projectile and pickups share in this game is that they produce lightning. When used correctly, lightning makes 2D video games a lot livelier and help the player understand what's happening on the screen at quick glance, as well as to easily identify dangers, useful items and so on. In order to produce this glowing effect, Unity's URP Volume post-processing tools have been used to correctly visualize the custom made glow materials for the game (see Figure 3.4).

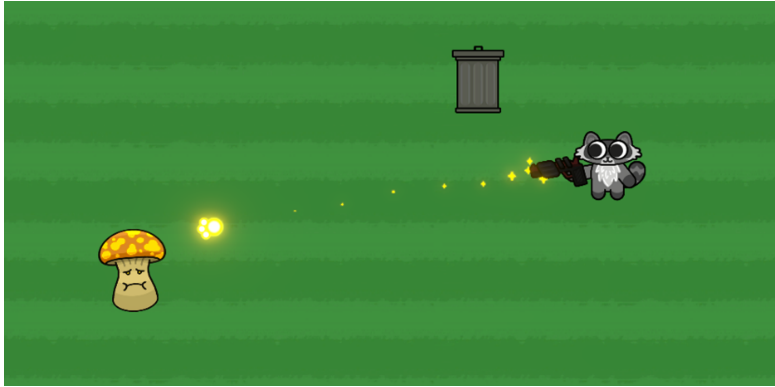


Figure 3.2: Example of "Lucky Cat"'s projectile shining particle effect.



Figure 3.3: Example of a trail left behind by the "Executor"'s projectile.

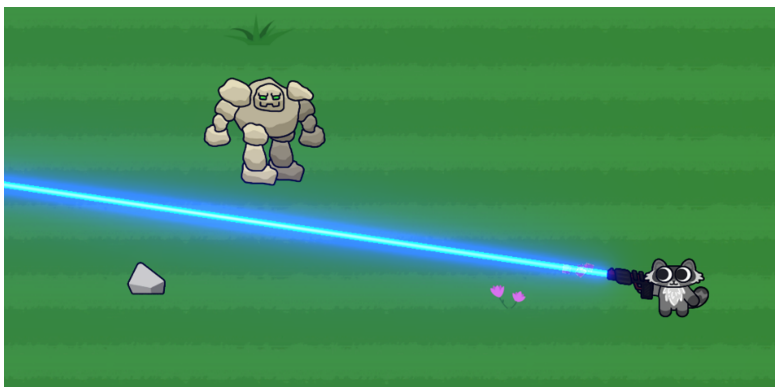


Figure 3.4: Illumination cast by "Railgun"'s ray.

## 3.2 Game Design

This section acts like a summary of the game design and the regular game flow. A more detailed approach to every system will be covered in chapter 4.

As the player launches the game, the first thing they will see is the main menu. Here two options will appear: to play or quit the game. Choosing to quit the game will obviously close the game entirely, but clicking on the play button will launch the loading screen. After the game scene is successfully loaded, the player will gain control over the playable character.

Once on the game screen, the player will be able to move the character using the keyboard and aim and shoot with the mouse. Endless hordes of enemies will start coming after the player, and as they defend themselves and defeat the monsters, those will drop experience orbs. Trash cans and boxes will also be generated randomly all over the map, and if the player breaks them, they will drop experience orbs, health recovery items and/or the magnet power up.

If enough experience is collected, the player will level up and trigger the bullet selection screen. On this screen, a random selection of bullets will be generated and the player will be able to pick one of them and add it to their magazine. In order to add a new bullet to the magazine, the player will have to swap one of the ones they already own by dragging and dropping the new bullet into the desired slot. However, if the player doesn't want to pick any of the bullets they got, this phase can be skipped by pressing the skip button.

If the player picks a new bullet, the game will resume. In the other hand, in case the player skips the bullet picking phase, they will trigger the stat upgrade screen. The stat upgrade screen is similar to the bullet picking one, being the difference that instead of random bullets, this screen offers a random selection of stat upgrades. If the player selects one of this upgrades, the character stat will be powered up accordingly and the game will resume. This screen can be skipped by pressing the skip button too, but this time it will just resume the game.

As the game progresses, new kinds of enemies will start appearing. Moreover, the maximum quantity of enemies that can be on screen and their health statistic scale with the passing of time, making the game harder minute after minute. If the character's health reaches zero or the game time surpasses the 15 minute mark, the player will loose/win the game and the game over screen will show up. This screen will display relevant data regarding the last game and give the player the possibility to play again or go back to the main menu. Choosing either of the two options will trigger the loading screen and bring the player to the desired scene.

Last but not least, the player will be able to pause the game by pressing the "Escape" button. The pause menu will display the character's current stats and give the player the option to resume the game or go back to the main menu.

### 3.3 Interface Design

Since first impressions are very important, designing a fun and appealing main menu is key. In *Bullet Ballet*, the main menu simulates an ammunition box: when the player clicks on it, a raccoon hand will appear and open the box. Inside there are two bullets that act like buttons (play and quit) and a small scribble of the game controls. This way, the player is greeted by a cute animation and can jump right into action, without the need of reading kilometric "how to play" panels or going through boring tutorials (see Figure 3.5).



Figure 3.5: A few frames from the Main Menu animation.

Transitions between the main menu and the game screen (or vice versa) and playing again once a game has finished, will trigger a loading screen. This screen displays a grey colored game logo that fills itself with white color as the current scene loads, acting like a loading bar and informing the player of the game loading progress in a stylish way (see Figure 3.6).



Figure 3.6: Game's loading screen.

Once inside the game scene, the interface displays the most relevant information during the playthrough in a clean and organized manner. At the bottom of the game screen the experience bar can be found, which represents the player's current experience progress towards leveling up. Above is the health bar, that shows how much health points the player currently has respect of their maximum health point value. And once again, above this bar is the player's current magazine: it shows what bullets the player currently has in their magazine and highlights the next bullet that will be fired when the character shoots its gun. Finally, at the top there's a digital clock that informs of the current game time.

The enemies have life bars too, but they will only appear upon taking damage so the screen doesn't get too cluttered. When an enemy is hit by a projectile, a small and brief hit effect will appear to give feedback to the player, and the damage dealt will be shown in form of damage numbers that will float as they disappear. In order to make the player feel powerful, the damage numbers have a considerable size so the player can see how his power grows as they progress, and in case a critical hit is dealt, a small explosion-like icon will show up. All of the health bars and the experience bar have smooth animations as they fill up or get emptied (see Figure 3.7).



Figure 3.7: Screenshot with all of the game HUD elements in display.

Once the player levels up, the bullet pick screen is triggered. On this screen the player can rearrange their magazine or pick a new bullet to swap for an old one. The random selection of new bullet appears at the top while the current player's magazine can be found below. If the player hovers over any bullet, a info panel will pop up at the left part of the screen, showcasing the statistics and description of said bullet. In case the player isn't interested in getting a new bullet, this phase can be skipped by pressing the skip button at the lower part of the screen. As for the bullet statistics, the numbers that appear at the right represent the bullet's base statistic, while the numbers to the left represent the real statistic after combining the base stat with the character current stat (see Figure 3.8).

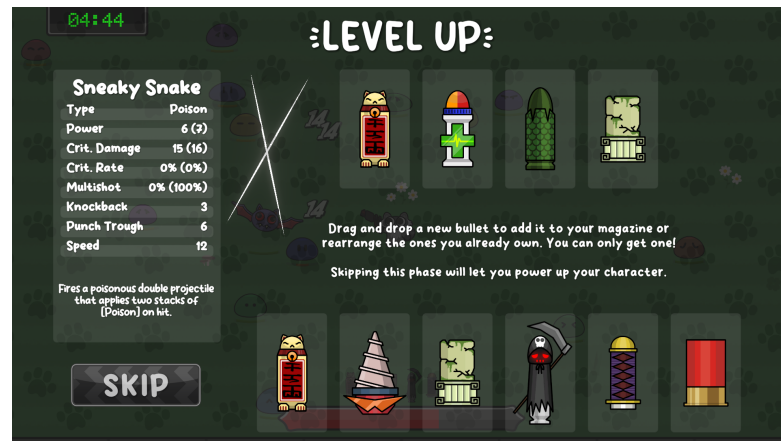


Figure 3.8: Example of a possible bullet pick screen.

Skipping the bullet pick phase will trigger the stat upgrade screen. The random selection of stat upgrades show up in the middle of the screen. Every stat upgrade has its own icon, name and quantity description. Depending on the rarity of the stat upgrade, its name will have a different color (ranging from green, blue, purple and gold). At the left, there is a stat panel that represents the character's current statistics so the player can strategize which stat upgrade suits better their current play style. Hovering over any of the stats from this panel will show a brief description of how said stat influences the game. Once again, this phase can be skipped too by pressing the skip button (see Figure 3.9).



Figure 3.9: Example with a possible stat upgrade screen.



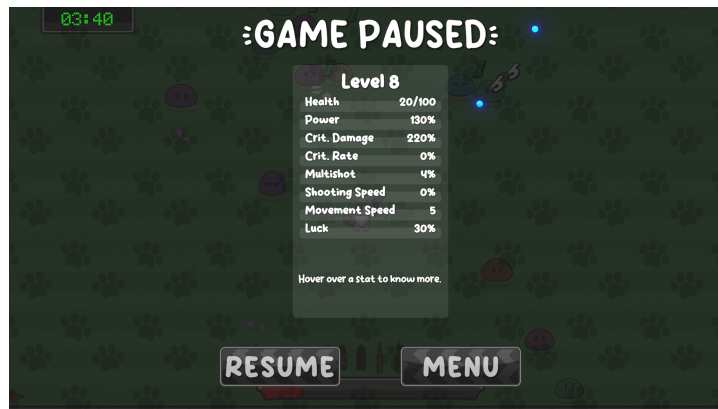


Figure 3.10: Game's pause menu.

In case the game is paused, the game flow will be stopped and the pause menu will appear. In the middle of this screen, a similar character stat panel to the one from the stat upgrade screen can be seen, and it works just in the same way. Below there are two buttons: resume and menu, which if pressed will resume the playthrough or take the player back to the main menu (see Figure 3.10).

If the player reaches the end of the game or the current health point value of the character drops to zero, the game flow will stop, the scene will blurry out and the game over screen will trigger. This screen displays valuable information regarding the last played game and a random real world raccoon trivia at the bottom [8]. Below the game data, there will be two buttons which will allow the player to star another game or go back to the main menu. Pressing any of these two buttons will enable the loading screen until the chosen screen is loaded (see Figure 3.11).

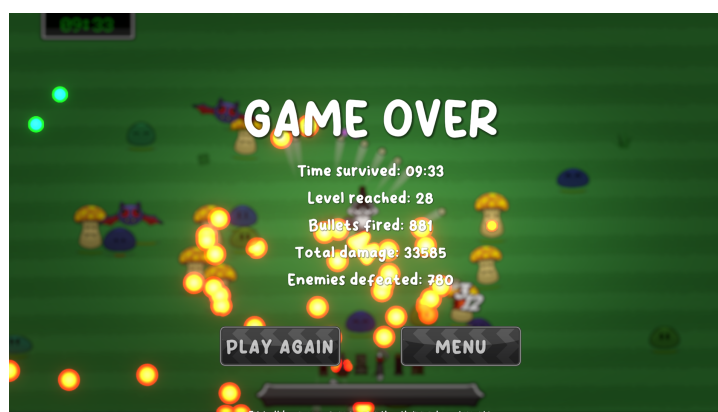


Figure 3.11: Game over screen scenario.

## 3.4 Sound Design

At first, the sound department of the game was meant to be secondary, something to do only if there was enough time. Nearing the end of development of the project, its priority was pushed and it was decided to include sound effects and music in the game. This is why sound related assets are the only ones that were not designed by myself. Instead, royalty free assets under level 0 universal creative commons were used.

In order to avoid the gun shot sound being too monotonous, every time the gun is fired one out of six random gun shot sounds is played [9]. The same happens with buttons, since the interface has a lot of interaction, the sound made upon button press is randomly chosen from a small pool of sounds. Dragging and dropping bullets in the bullet pick screen has its own sounds too [17].

A fitting and cozy background song is played and looped seamlessly during the play trough [15], as well as some other sounds to give the player feedback, like sounds upon item pickup, enemy hit, enemy defeat or game over [2]. Some other secondary sounds were added to the main menu animation in order to make it feel more organic.

## 3.5 Requirement Analysis

### 3.5.1 System Architecture

Bullet Ballet is not a very demanding game, in order to be played it only requires a mouse so the player can take aim, shoot and interact with the interface, a keyboard to move the character and a computer able to support Unity Player's minimum system requirements, which are the following:

- **Operating system version:** Windows 7 / High Sierra 10.13 / Ubuntu 18.04 or higher.
- **CPU :** X64 architecture with SSE2 instruction set support
- **Graphics API:** DX10, DX11, and DX12-capable GPUs
- **Additional requirements:** Hardware vendor officially supported drivers

### 3.5.2 Functional Requirements

Since the base requirements are pretty simple and the core game cycle has been previously explained, identifying the functional requirements is also pretty easy:

- \* R1. The player can start the game.
- \* R2. The player can pause the game.

- \* R3. The player can quit the game.
- \* R4. The player can move, aim and shoot.
- \* R5. The player and his enemies can take damage.
- \* R6. The enemies are able to periodically spawn.
- \* R7. The enemies teleport closer if the payer gets too far away.
- \* R8. The player is able to pick up experience and other items.
- \* R9. The player is able to level up.
- \* R10. The player is able to rearrange their magazine and get new bullets.
- \* R11. The player is able to upgrade their stats on bullet discard.
- \* R12. The game ends if player dies or beats the game.
- \* R13. The relevant information is clearly displayed on the interface.
- \* R14. The projectile system and character stats work properly.
- \* R15. The game environment is infinitely generated.
- \* R16. Different enemies have different stats and behaviours.
- \* R17. Different bullets have different stats and effects.

### 3.5.3 Non-functional Requirements

As for the non-functional requirements, these are the requirements that impose conditions on the design and quality aspects of the project:

- \* R1. The game will have a cozy and friendly art direction.
- \* R2. The bullet system has to offer great variety.
- \* R3. The interface has to be clean, clear and non obstructing.
- \* R4. The game has a unique personality.
- \* R5. The game offers appealing visual effects.
- \* R6. The game is fun for every kind of player.
- \* R7. The game is well balanced and rewarding.

## 3.6 System Design

Based on the previously stated requirements, the following are the cases of use that are possible within the game, as well as a use case diagram (see Figure 3.12).

Requirement:	R1
Actor:	Player
Description:	While on the main menu screen, the player will be able to start the game by pressing the "play" button.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the main menu screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player clicks on the "play" button.</li> <li>2. Loading screen is launched.</li> <li>3. After loading the game, the player can take control.</li> </ol>
Alternative sequence:	None.

Table 3.1: Functional requirement «CU1. PLAY GAME»

Requirement:	R1
Actor:	Player
Description:	The player is able to pause the game.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the game screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player presses the "Escape" button.</li> </ol>
Alternative sequence:	<ol style="list-style-type: none"> <li>1. The game is already paused.</li> <li>2. The level up screen, which also pauses the game, is on.</li> </ol>

Table 3.2: Functional requirement «CU2. PAUSE GAME»

Requirement:	None
Actor:	Player
Description:	While on the main menu screen, the player will be able to quit and close the game.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the main menu screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player clicks on the "quit" button.</li> </ol>
Alternative sequence:	None.

Table 3.3: Functional requirement «CU3. QUIT GAME»

Requirement:	R2
Actor:	Player
Description:	The player is able to return to the main menu by pressing the corresponding button on the pause/game over screen.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the pause or game over screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player clicks on the "main menu" button.</li> <li>2. Loading screen is launched.</li> <li>3. After loading the menu, the player can use the buttons.</li> </ol>
Alternative sequence:	None.

Table 3.4: Functional requirement «CU4. RETURN TO MAIN MENU»

Requirement:	R1
Actor:	Player
Description:	The player is able to move the character using the keyboard and aim using the mouse.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the game screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player presses the WASD keys and/or moves the mouse.</li> </ol>
Alternative sequence:	<ol style="list-style-type: none"> <li>1. The player is pushed by enemies,</li> </ol>

Table 3.5: Functional requirement «CU5. MOVE»

Requirement:	R1
Actor:	Player
Description:	The player is able to shoot the current active bullet from their magazine.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the game screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player presses or holds the left mouse click.</li> </ol>
Alternative sequence:	<ol style="list-style-type: none"> <li>1. The player is trying to shoot faster than their current shooting speed stat allows.</li> </ol>

Table 3.6: Functional requirement «CU6. SHOOT»

Requirement:	R1
Actor:	Player
Description:	The player is able to pick up items when they are inside their pick up radius.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the game screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player collides with an item.</li> </ol>
Alternative sequence:	None.

Table 3.7: Functional requirement «CU7. PICK UP ITEMS»

Requirement:	R9
Actor:	Player
Description:	Allows the player to get a new bullet from a randomly generated selection on level up
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the game screen.</li> <li>2. The player has reached enough experience points to trigger the level up screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player drags and drops the desired bullet into one of their magazine slots.</li> <li>2. The game resumes.</li> </ol>
Alternative sequence:	<ol style="list-style-type: none"> <li>1. The player is able to skip this phase if none of the bullets are of their interest.</li> </ol>

Table 3.8: Functional requirement «CU5. GET NEW BULLETS»

Requirement:	R10
Actor:	Player
Description:	If the player discards the new bullets on level up, they will be able to pick a stat upgrade from a random selection of stat boosts.
Preconditions:	<ol style="list-style-type: none"> <li>1. The player is on the bullet pick screen.</li> </ol>
Normal sequence:	<ol style="list-style-type: none"> <li>1. The player selects one of the upgrades.</li> <li>2. The game resumes.</li> </ol>
Alternative sequence:	<ol style="list-style-type: none"> <li>1. The player is able to skip this phase if they don't want any upgrades.</li> </ol>

Table 3.9: Functional requirement «CU9. POWER UP STATS»

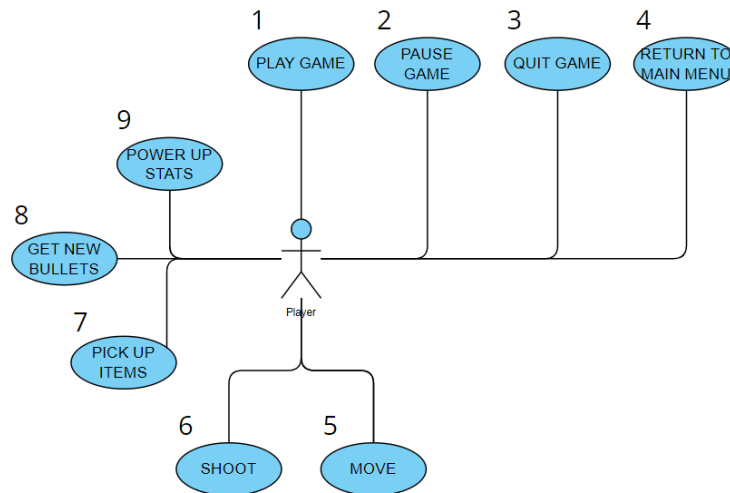


Figure 3.12: Case use diagram (designed thanks to Visual Paradigm [12]).



## GAME SYSTEM DESIGNS

### Contents

---

4.1	Genres and Inspirations . . . . .	<b>27</b>
4.2	Character Statistics . . . . .	<b>28</b>
4.3	Projectile Statistics . . . . .	<b>30</b>
4.4	Enemy Statistics . . . . .	<b>31</b>
4.5	Bullets . . . . .	<b>32</b>
4.6	Enemy Types and Behaviour . . . . .	<b>51</b>
4.7	Containers and Items . . . . .	<b>52</b>

---

This chapter covers a deeper dive into the game systems and mechanics, ranging from genre analysis to design choices and detailed game system dissection.

### 4.1 Genres and Inspirations

As discussed in previous chapters, Bullet Ballet gameplay’s focus was to fuse the 2D shoot’em up genre with the intricacies of the emerging survivors-like sub-genre. The core inspiration for this project was to take the foundations of Vampire Survivors and introduce innovative and original mechanics and gameplay elements in order to create some sort of a hybrid genre.

In terms of artistic design, the project leans towards a whole different direction, looking for a cartoon and fun style as opposed to Vampire Survivor’s dark and pixelated old fashioned look. Some indie games like *Cat Quest* and *Cult of the Lamb* served as a great source of inspiration (see Figure 4.1).



Figure 4.1: Screenshots from some of the games that served as artistic inspiration.

## 4.2 Character Statistics

The playable character has a wide array of different statistics, most of which can be upgraded during the course of the play trough. At the start of a new game, these stats reset to their respective default values. Next is an in depth analysis of every single one of this statistics and how do they affect the game:

- **Level:** the character starts at level zero, and each time that the player reaches the required amount of experience points needed to level up, this number increases by one. On leveling up, the player is able to pick a new bullet to add to their magazine or upgrade one of the other stats.
- **Experience:** this value increases each time the character picks up an experience orb. The required experience to level up scales using the following formula:

$$\text{experience needed to level up} = (\text{current player level} + 1) * 6$$

- **Maximum health:** this is the maximum health cap value that player's current health can't ever surpass. The character starts with a value of 100 maximum health points, but this value can be upgraded. On upgrade, the current health is proportionally healed to fit said increase.
- **Current health:** if the current amount of health points reaches or drops below zero, the game will end. Getting attacked by enemies or hit by their projectiles will deplete this value in varying quantities. The current health point amount can be recovered in different ways, but it will never surpass the maximum health value.
- **Power:** this value is multiplied by the value of every bullet's raw damage in order to calculate the total projectile damage. For example, if a projectile deals 6 points

of damage and the player currently has 200% power, the total damage of that projectile will be 12. The character starts with 100% power.

- **Movement speed:** this value dictates the speed at which the character is able to travel across the map.
- **Shooting speed:** the speed at which the character is able to shoot. The higher this value, the lower will be the cooldown between shots. At the start of the game, the character can only fire one bullet per second. However, this cooldown is reduced by the shooting speed statistic in the following way:

$$\text{cooldown between shots} = 1 - \frac{\text{shooting speed percent}}{100}$$

The shooting speed stat has a soft cap of 75% meaning that the real maximum shooting speed that a player can achieve is 4 shots per second. The game starts with 0% shooting speed.

- **Multishot:** the multishot statistic represents the chances of firing additional projectiles on each bullet shot. For instance, if the character has 50% multishot, there is a 50/50 chance for the character to fire 2 projectiles instead of one on each shot. Another example: having 200% multishot chance means that every single bullet shot will fire two additional projectiles of the same kind. The additional projectiles fired through the multishot chance will have a slight offset in accuracy. The character starts with 0% multishot chance and this value is additive to the multishot stat of each bullet.
- **Critical damage:** in case a projectile deals a critical hit, the total damage dealt by that projectile will be multiplied by the critical damage percent. For example, if a bullet that deals 8 damage turns out to be a critical hit, and the character currently has a 300% critical damage value, the projectile will deal 24 points of damage. The character starts with 200% critical damage and this value is additive to every bullet's critical damage stat.
- **Critical chance:** this value defines the chances of a projectile to deal critical damage. If the critical chance is higher than 100%, let's take 150% for example, there's a 50/50 chance for the projectile to deal double critical damage. The base character critical chance value is 0%, percentage which is additive to a bullet's own critical chance.
- **Luck:** the higher the luck value, the better the odds of the character finding more or higher quality loot.
  - *Bullet pick screen:* at 0% luck the player will always get 3 new random bullets. However if the luck value is 50%, the player will have a 50/50 chance to get an additional fourth choice. This bonus scales up to a maximum of 200% luck, in which case the player would always find 5 new random bullets.

- *Stat upgrade screen*: at 0% luck the player will always get 3 random stat power ups. However if the luck value is 50%, the player will have a 50/50 chance to get an additional fourth random stat upgrade. This bonus scales up to a maximum of 200% luck, in which case the player would always 5 random stat upgrade choices.
- *Container drops*: at 0% luck, the containers will always drop a random item on destruction. The higher the character's luck, the higher are the odds of getting up to a maximum of three drops from each container. However and for example, at 200% luck, a container is able to drop a maximum of three items, but it is not necessarily guaranteed to.

Upon entering the stat upgrade screen, the stat upgrades are generated randomly and they may come in four rarities that translate into how high are the stat boosts. This four rarities are Common (green color, 40% chance), Rare (blue color, 30% chance), Epic (purple color, 15% chance) and Legendary (golden color, 5% chance). The following is a table containing of all of the stats that can be upgraded via the upgrade stat screen and how much of an improvement they are depending on the rarity rolled.

	Common	Rare	Epic	Legendary
Maximum Health	+5	+15	+25	+50
Power	+5%	+10%	+15%	+30%
Movement Speed	+0.15	+0.3	+0.5	+1
Shooting Speed	+5%	+7%	+10%	+20%
Multishot Chance	+4%	+8%	+16%	+32%
Critical Damage	+10%	+20%	+30%	+60%
Critical Chance	+5%	+10%	+15%	+30%
Luck	+10	+20	+30	+60

Table 4.1: Possible stat upgrades and their scalings depending on rarity.

### 4.3 Projectile Statistics

Each single projectile in the game has its own set of statistics with customized stat values. This section covers and explains all of them in detail. Each projectile has...

- **Name**: every projectile has the name of the corresponding bullet that generates them attached.
- **Type**: there are four types of projectiles in the game. These categories exist in order to force the player to explore all of the bullets and build their loadout around this knowledge.
  - *Basic*: weak bullets with no special behaviours. The character starts with 6 bullets of this type until they find better ones.

- *Physical*: powerful bullets with very diverse effects. Designed to synergize with [Like A Samurai] bullet.
  - *Plasma*: futuristic style bullets designed to synergize with [Railgun] bullet.
  - *Poison*: this type of bullet inflicts poison status to the enemies they hit. Designed to synergize with [Antidote] bullet. Poison status inflicts 1% of enemy's maximum health as damage every second per poison stack. A maximum of 10 stacks of poison can be applied to every enemy.
- **Description**: every projectile has a string attached with a brief explanation of its behaviour and effects.
  - **Damage**: the damage value of the projectile, which gets multiplied by the character's power percent upon impact.
  - **Speed**: the travel speed of the projectile. The higher this number, the faster it will go.
  - **Punch trough**: the number of enemies/obstacles/containers a projectile is able to pierce through before autodestroying.
  - **Multishot**: the chance of firing additional projectiles on each shot, this value is additive to the character's multishot percentage.
  - **Knockback**: the force with which a projectile pushes back an enemy upon impact. The higher this value, the further the enemy will be knocked back.
  - **Life span**: the maximum amount of time that the bullet is able to be alive before it autodestroys.
  - **Critical damage**: the damage percent that gets multiplied by the total damage upon achieving a critical hit. This value is additive to the character critical damage percentage.
  - **Critical chance**: the odds of every impact of a projectile being a critical hit. This value is additive to the character's current critical chance.

## 4.4 Enemy Statistics

Lastly and in the same nature as the character and projectiles, the enemies have their own set of statistics, much simpler in their case:

- **Base health**: the starting and maximum value of an enemy health point amount.
- **Current health**: the health point amount an enemy has at a given moment. If this value drops to zero or below, the enemy will die and instantiate the death visual effect, as well as an experience orb.

- **Base movement speed:** the speed at which the enemy is able to travel across the map.
- **Current movement speed:** some factors like knockback or [Fight The Power] bullet effect are able to reduce an enemy's movement speed. For this cases, the current movement speed value is used while the slow effect is in course, and then restored to the base movement speed value.
- **Base damage:** the damage an enemy deals to the character upon contact.

## 4.5 Bullets

After exploring the different statistics that the game has in play, now the bullet system can be discussed too. There is a total of 18 different bullets in the game, each one with custom values for every projectile statistic and different abilities, behaviours and visual effects. Since the whole game revolves around the bullet system, it is important to analyze every single one of them. This section summarizes their custom stats and behaviours.

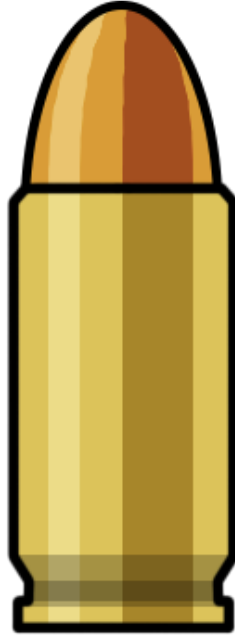


Figure 4.2: Pistol Bullet

**Name:** Pistol Bullet

**Type:** Basic

**Description:** Basic pistol bullet. Has great damage and deals a little knockback.

**Base damage:** 8

**Base speed:** 15

**Base punch trough:** 0

**Base multishot:** 0

**Base knockback:** 5

**Base life span:** 5 seconds

**Base crit chance:** 0

**Base crit damage:** 0



Figure 4.3: Sniper Bullet

**Name:** Sniper Bullet

**Type:** Basic

**Description:** Basic sniper bullet. Pierces through a few enemies, but has no knockback.

**Base damage:** 5

**Base speed:** 50

**Base punch trough:** 2

**Base multishot:** 0

**Base knockback:** 5

**Base life span:** 5 seconds

**Base crit chance:** 15%

**Base crit damage:** 25%



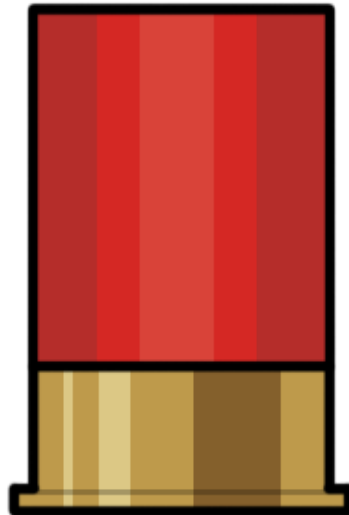


Figure 4.4: Shotgun Bullet

**Name:** Shotgun Bullet

**Type:** Basic

**Description:** Basic shotgun bullet. Best used in close combat.

**Base damage:** 6

**Base speed:** 0

**Base punch trough:** Unlimited

**Base multishot:** 0%

**Base knockback:** 25

**Base life span:** 0.1 seconds

**Base crit chance:** 0%

**Base crit damage:** 0%



Figure 4.5: Lucky Cat Bullet

**Name:** Lucky Cat

**Type:** Physical

**Description:** Each enemy hit increases projectile's critical rate by 7% and critical damage by 14%.

**Base damage:** 3

**Base speed:** 15

**Base punch trough:** 7

**Base multishot:** 0%

**Base knockback:** 2

**Base life span:** 5 seconds

**Base crit chance:** 7%

**Base crit damage:** 14%



Figure 4.6: Executor Bullet

**Name:** Lucky Cat

**Type:** Physical

**Description:** Instantly kills enemies below 20% health.

**Base damage:** 6

**Base speed:** 50

**Base punch trough:** 13

**Base multishot:** 0%

**Base knockback:** 0

**Base life span:** 5 seconds

**Base crit chance:** 13%

**Base crit damage:** 26%



Figure 4.7: Shock Therapy Bullet

**Name:** Shock Therapy

**Type:** Physical

**Description:** Heals the player by a small amount on enemy hit.

**Base damage:** 6

**Base speed:** 15

**Base punch trough:** 0

**Base multishot:** 200%

**Base knockback:** 10

**Base life span:** 5 seconds

**Base crit chance:** 10%

**Base crit damage:** 20%



Figure 4.8: Artificial Violence Bullet

**Name:** Artificial Violence

**Type:** Plasma

**Description:** Permanently upgrades a random bullet stat each time it's fired. Upgrades are shared with other Artificial Violence bullets.

**Base damage:** 5

**Base speed:** 10

**Base punch trough:** 0

**Base multishot:** 0%

**Base knockback:** 3

**Base life span:** 5 seconds

**Base crit chance:** 0%

**Base crit damage:** 0%



Figure 4.9: Fight The Power Bullet

**Name:** Fight The Power

**Type:** Physical

**Description:** Significantly slows enemies on its path as it drills trough them.

**Base damage:** 3

**Base speed:** 3

**Base punch trough:** Unlimited

**Base multishot:** 0%

**Base knockback:** 0

**Base life span:** 10 seconds

**Base crit chance:** 25%

**Base crit damage:** 50%



Figure 4.10: Plasmarang Bullet

**Name:** Plasmarang

**Type:** Plasma

**Description:** Boomeranging projectile that deals more damage and bigger knockback the faster it gets.

**Base damage:** 3

**Base speed:** 15

**Base punch trough:** 10

**Base multishot:** 0%

**Base knockback:** 5

**Base life span:** 5 seconds

**Base crit chance:** 25%

**Base crit damage:** 50%



Figure 4.11: Rock & Stones Bullet

**Name:** Rock & Stones

**Type:** Physical

**Description:** Shoots a rock with great knockback that splits in smaller piercings rocks upon impact.

**Base damage:** 12

**Base speed:** 9

**Base punch trough:** 0

**Base multishot:** 0%

**Base knockback:** 50

**Base life span:** 5 seconds

**Base crit chance:** 15%

**Base crit damage:** 50%





Figure 4.12: Bee Gone Bullet

**Name:** Bee Gone

**Type:** Poison

**Description:** Follows a random enemy and applies one stack of [Poison] on impact.

**Base damage:** 2

**Base speed:** 15

**Base punch trough:** 3

**Base multishot:** 400%

**Base knockback:** 1

**Base life span:** 10 seconds

**Base crit chance:** 25%

**Base crit damage:** 50%



Figure 4.13: Railgun Bullet

**Name:** Railgun

**Type:** Plasma

**Description:** Projects a plasma beam that damages enemies upon contact. Each [Plasma] type bullet in the magazine extends beam duration.

**Base damage:** 3

**Base speed:** 0

**Base punch trough:** Unlimited

**Base multishot:** 0%

**Base knockback:** 2

**Base life span:** 0.5 seconds

**Base crit chance:** 0%

**Base crit damage:** 0%



Figure 4.14: Shooting Star Bullet

**Name:** Shooting Star

**Type:** Plasma

**Description:** Launches a spinning projectile that deals continuous area of effect damage.

**Base damage:** 3

**Base speed:** 15

**Base punch trough:** Unlimited

**Base multishot:** 0%

**Base knockback:** 1

**Base life span:** 5 seconds

**Base crit chance:** 25%

**Base crit damage:** 50%



Figure 4.15: Bangerbox Bullet

**Name:** Bangerbox

**Type:** Plasma

**Description:** Launches a slow soundwave that expands with time and has powerful knockback. Blocks enemy projectiles.

**Base damage:** 12

**Base speed:** 2.5

**Base punch trough:** Unlimited

**Base multishot:** 20%

**Base knockback:** 35

**Base life span:** 2 seconds

**Base crit chance:** 10%

**Base crit damage:** 20%



Figure 4.16: Sneaky Snake Bullet

**Name:** Sneaky Snake

**Type:** Poison

**Description:** Fires a poisonous double projectile that applies two stacks of [Poison] on hit.

**Base damage:** 6

**Base speed:** 12

**Base punch trough:** 6

**Base multishot:** 20%

**Base knockback:** 3

**Base life span:** 5 seconds

**Base crit chance:** 25%

**Base crit damage:** 50%

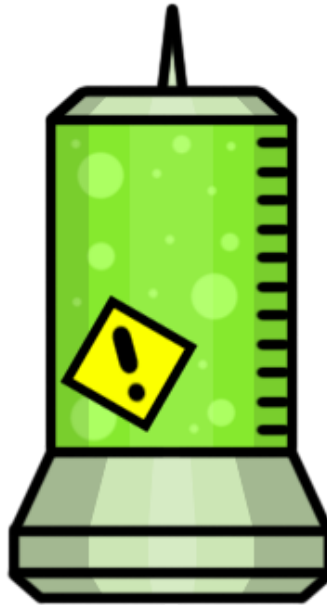


Figure 4.17: Lethal Injection Bullet

**Name:** Lethal Injection

**Type:** Poison

**Description:** Makes up for its slow knockback and range by applying 3 stacks of [Poison] on contact.

**Base damage:** 7

**Base speed:** 0

**Base punch trough:** Unlimited

**Base multishot:** 20%

**Base knockback:** 17

**Base life span:** 0.1 seconds

**Base crit chance:** 25%

**Base crit damage:** 50%

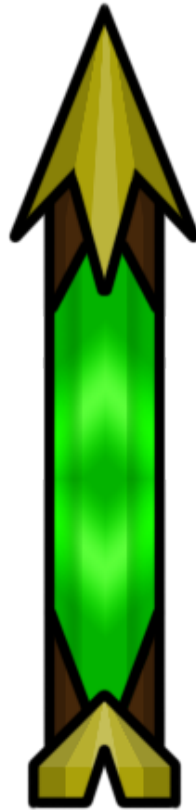


Figure 4.18: Antidote Bullet

**Name:** Antidote

**Type:** Poison

**Description:** Consumes the [Poison] stacks of the enemies it pierces instantly triggering 3 instances of [Poison] damage per stack.

**Base damage:** 5

**Base speed:** 50

**Base punch trough:** 10

**Base multishot:** 20%

**Base knockback:** 0

**Base life span:** 5 seconds

**Base crit chance:** 25%

**Base crit damage:** 50%

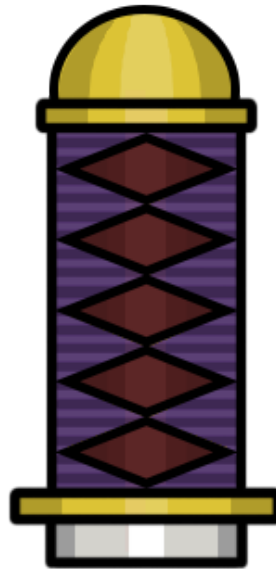


Figure 4.19: Like A Samurai Bullet

**Name:** Like A Samurai

**Type:** Physical

**Description:** Performs a powerful slash for every [Physical] type bullet in the magazine.

**Base damage:** 12

**Base speed:** 0

**Base punch trough:** Unlimited

**Base multishot:** 0%

**Base knockback:** 2

**Base life span:** 0.2 seconds per slash

**Base crit chance:** 10%

**Base crit damage:** 100%



## 4.6 Enemy Types and Behaviour

As the game progresses, different types of enemies will start appearing. There are 4 in total, and they grow stronger and appear more frequently as the time passes by. Next is a description of every kind of enemy and the philosophy behind their design:



Figure 4.20: The four enemy types that the player may encounter.

- **Slimes:** slimes are the most common enemy that the player will face. They are slow and don't have too many health points. This enemy type is easy to avoid but may become dangerous if too many of them end up surrounding the player. Slimes' colors and faces are completely randomly generated, so the player won't ever see two identical slimes.
- **Bats:** they are very fast and move in unpredictable ways, however they possess an even lower health amount than slimes. Defending against bats should be the top priority of the player. Experience players will notice that shotgun-like or homing bullets are very effective against them.
- **Mushrooms:** this kind of enemy is really fast, but they won't get too close to the player. Mushrooms move towards the player but keep their distance as they shoot dangerous spores. Letting too many mushroom enemies surround the player will prove dodging their attacks very difficult, so they should be their second focus. Shotgun-like bullets are ineffective against them, but long distance projectile work really well.
- **Golems:** they are the healthiest enemy in the game, and what's more, they will continuously keep healing themselves if the player doesn't defeat them fast. In return, they are very slow and only appear in later stages of the game. Building a lot of damage is essential to be able to defeat them.

As the time passes, enemy base health increases using the following formula:

$$\text{current max health} = \text{base health} * \left(1 + \frac{0.5 * \text{current game minute}}{15}\right)$$

The maximum monster number that can be alive at the same time, once again scales with game time, and it is decided by this formula:

$$\text{maxenemynumber} = 30 + (\text{currentgametime} * 5)$$

The last variable in the monster spawning rate is the speed at which they appear. This number is calculated this way:

$$\text{spawncooldowninseconds} = 2 + \left(\frac{\text{currentgameseconds}}{1000}\right)$$

Formulas aside, not every monster appears right from the start of the game. The chances of each enemy appearance vary depending on the current game minutes and are designed by hand picked values to balance the game appropriately. Said values are the following:

Minute	<3	4	5	6	7	8	9	10	11	12	13	14	15
Slime %	100%	90%	85%	75%	75%	75%	70%	60%	60%	60%	60%	50%	0%
Bat %	0%	10%	15%	25%	15%	10%	10%	15%	20%	10%	10%	5%	0%
Mushroom %	0%	0%	0%	0%	10%	15%	20%	25%	15%	15%	15%	20%	20%
Golem %	0%	0%	0%	0%	0%	0%	0%	0%	5%	10%	15%	30%	70%

Table 4.2: Every enemy appearance rate in function of the current game time.

## 4.7 Containers and Items

As the player travels across the map, they will randomly encounter containers such as wooden boxes and trash cans. If these environmental props get shot, they will drop useful items as they play a breaking animation. If the character gets close enough to a pick up item, it will be pulled towards the character and give them different benefits depending on the item type (see Figure: 4.21).



Figure 4.21: Every container and item available in the game.

There are three types of pick up items in the game:

- **Experience orbs:** the rarer the experience orb, the more experience points it will give to the player. Higher luck stat translates into better odds of finding higher quality experience from enemy drops.
  - *Common orbs:* blue and common orbs that give 1 experience point.
  - *Uncommon orbs:* somewhat rarer orbs that give 5 experience points.
  - *Rare orbs:* the most uncommon of them all, gives 15 experience points.
- **Health recovery:** the heart shaped item restores character's health on pickup, taking into consideration the game time as follows  $health\ restored = 20 + (20 * \frac{current\ game\ minute}{10})$ .
- **Magnet:** instantly attracts all of the pick up items in a really wide area.

As stated in the Character Statistics chapter, containers may drop up to 3 items at once depending on the character's luck stat. However, luck does not affect the chances of getting better items from containers, their rates always stay the same and they are the following:

	Chances to drop from container
Uncommon experience orb	40%
Rare experience orb	30%
Health restoration item	15%
Magnet	5%

Table 4.3: Chances of each possible item to be dropped upon container destruction.



## WORK DEVELOPMENT AND RESULTS

### Contents

---

5.1	Work Development . . . . .	<b>55</b>
5.2	Results . . . . .	<b>56</b>
5.3	Beta test . . . . .	<b>56</b>

---

This chapter covers the intricacies of the development process of Bullet Ballet, all of the hardships and achievements that were carried out along the way. The final results and the playable final version of the game can be found at the end of this chapter.

### 5.1 Work Development

At first this seemed like an ambitious project, and in some ways it was. The lack of time was one of the biggest problems. As it always happens with big projects, some tasks were underestimated and took way longer than expected, or vice versa. One of the hardest parts of the development of this game was the design and programming of the projectile system. There were too many things to take in consideration to build a just and appealing stat system, but I believe in the end it ended up working really well. The projectile system served as an exceptional foundation for every single projectile in the game, and the time working on this system was well spent.

The second part that ended up being one of the hardest was the development of the magazine and random bullet drop system. Getting this system to work as intended was an odyssey. A lot of different game objects and interfaces [3] are involved in this part of the project and finding a way to correctly interconnect them and make them interact was hard. However, despite all of the time spent on this systems, it was well worth it

and ended up working like a charm (of course, if we don't take into account the chaotic coding behind the scenes).

This hardships took away a lot of time from another aspects of the game that could be improved or expanded upon. The world generation system is pretty simple, since environments in this kind of games are secondary, but it could be better if I had time to design more and better looking assets. And the sound effects and music are not that great, since I haven't been taught to produce them myself and learning to do so is way too time consuming, so I ended up using the best royalty-free ones that I could find.

Some art assets like for example the monster designs could be better, or the weapon that the character wields could be more aesthetically fitting, but the time was ticking and there were other priorities at the time. In terms of artistic design, pretty much all of the visual effects look really good. However, it was really hard getting the glow effects [13] to work and they were implemented in a very limiting way.

Nonetheless, despite all of the hard times I had along this journey and even if some aspects had to be rushed or are not the best they could have been, this was a passion project and I'm happy about how it turned out in the end.

## 5.2 Results

The project was finished in time and the game came out well polished. After finishing the project, a brief 2 day small scale beta test was held. The feedback given by the testers and the project supervisor were very useful and helped to improve and balance the game further.

The source code of the project can be found here:

«SOURCE CODE»

The assets and PSD files designed for the project can be reviewed here:

«GAME ASSETS»

The final playable version of the game can be downloaded from here:

«FULL GAME»

## 5.3 Beta test

During the last 2 days before this work report was handed out, a small beta test was held between volunteering players. They were given the chance to try the game and share their scores as well as some opinions about the game.



Figure 5.1: Collage of some of the scores sent by the beta testers.

This is a translated compilation of the beta testers' opinions and rants about the game:

**Name:** León.

**Do you have any experience within the "survivors-like" genre?:** Yes, I've played Vampire Survivors, 20 minutes till dawn and survivor.io.

**What were the things that you liked the most in this game?:** I find the bullet swapping mechanic very fun and original

**What were the ones that you didn't like?:** None!

**What would you improve?:** Maybe zooming out the camera a little bit, in this kind of games you need to have a great field of vision.

**Your favourite bullet?:** The pink chainsaw one, it's really strong.

**Name:** Dani.

**Do you have any experience within the "survivors-like" genre?:** No, I don't.

**What were the things that you liked the most in this game?:** The variety of bullets and their effects.

**What were the ones that you didn't like?:** Bats are too annoying, it's very hard to hit them when they get close.

**What would you improve?:** Various things: more synergies between bullets, bosses, power ups like temporal invincibility or really high attack speed, and sounds. These changes would improve the game a lot.

**Your favourite bullet?:** Railgun.

**Name:** Tania.

**Do you have any experience within the "survivors-like" genre?:** Just a little.

**What were the things that you liked the most in this game?:** The interface aesthetics, the gameplay is very engaging and it motivates you to try out different kinds of bullets, and the enemy types.

**What were the ones that you didn't like?:** I believe that too many enemies appear at the beginning.

**What would you improve?:** Make the enemies spawn more slowly.

**Your favourite bullet?:** The purple one that deals damage in area.

**Name:** Rodrigo.

**Do you have any experience within the "survivors-like" genre?:** No, I don't.

**What were the things that you liked the most in this game?:** The bullet management aspect and the way you have to play around their types.

**What were the ones that you didn't like?:** The beginning of the game feels kinda slow, but I guess that this is something you would expect in this kind of games. And the bats are too hard to hit.

**What would you improve?:** Music and sound effects (obviously) and I think that the game needs to be a little more forgiving, maybe tone down the enemy damage or make more ways to heal. **Your favourite bullet?:** The one that shoots bees.

All of the precious feedback given by the beta testers was taken into consideration and some aspects of the game were tweaked in order to offer a better experience. Some of this adjustments were:

- Adding music and a lot of sound effects.
- Zooming out the camera in order to expand the player's vision.
- Making the bat enemy movement speed a little slower.
- Considerably decreasing the spawn rate of enemies.
- And some minor bug fixes.



## CONCLUSIONS AND FUTURE WORK

### Contents

---

6.1	Conclusions . . . . .	<b>59</b>
6.2	Future work . . . . .	<b>60</b>

---

In this final chapter, I express my conclusions regarding the project and what are my plans for the future.

### 6.1 Conclusions

Working on this Degree Final Project was a great opportunity to put into practice all that I've learnt so far about video game design and development. Not only that, but I've also learnt new things along this journey.

Unfortunately, my end of grade practices consumed a lot of time and energy, so I couldn't work as much as I wished on this project and some parts of it may seem rushed. However, this was the perfect opportunity to work on a passion project and I'm glad that my supervisor approved my ideas and let me work on them. Even if the time was scarce, the effort put into this project was enormous and the last month of work was restless. These inconveniences aside, I'm really happy about how the game turned out and I believe that it's a very fun game with fresh and polished ideas.

Overall and despite all of my hardships, this was a great experience. I've worked on a project with a lot of eagerness, I've honed my knowledge and skills, and acquired some new ones that will be useful in the future.

## 6.2 Future work

I have had tons of ideas for this project, and unfortunately lots of them couldn't make it into the final version. There are a lot of ways to improve the game, expand on the current foundations and explore new mechanics. It would also be great learning sound effect production and working with an artist on a soundtrack for the game; this way 100% of the game assets would be done by me.

Eventually, it would also be nice to publish the game on online stores. For example, itch.io [4] would be a great start, since publishing games in their store is completely free. Later, if the game grabs enough attention and gathers some funds, a license could be bought to publish the game in a more professional platform like Steam [5].

Right now, I'm a little burnt out because of all the recent work I had to do on the project and I plan to stay focused on finishing my studies, but I would gladly keep working on expanding this game in the near future.

## BIBLIOGRAPHY

- [1] Adobe. Adobe photoshop. <https://www.adobe.com/es/products/photoshop.html>. Accessed: 2023-05-03.
- [2] Chhoff. Classic ui sfx | audio pack for games or twitch. <https://chhoffmusic.itch.io/classic-ui-sfx>. Accessed: 2023-05-08.
- [3] Coco Code. Drag and drop in unity ui - create your own inventory ui! <https://youtu.be/kWRyZ3hb1Vc>. Accessed: 2023-05-04.
- [4] Leaf Corcoran. Itch.io. <https://itch.io>. Accessed: 2023-05-15.
- [5] Valve Corporation. Steam. <https://store.steampowered.com>. Accessed: 2023-05-15.
- [6] DaFont. Dafont. <https://www.dafont.com/es/>. Accessed: 2023-05-03.
- [7] Germán Fabregat Lluca. Guía para la redacción de las memorias. [http://mermaja.act.uji.es/itis/IS31/guia\\_memoria.pdf](http://mermaja.act.uji.es/itis/IS31/guia_memoria.pdf). Accessed: 2019-02-28.
- [8] Facts.net. Raccoon facts. <https://facts.net/raccoon-facts/>. Accessed: 2023-05-04.
- [9] FilmmakersManual. Filmmakersmanual's freesound gallery. <https://freesound.org/people/FilmmakersManual/>. Accessed: 2023-05-08.
- [10] GanttProject. Gantt project. <https://www.ganttproject.biz>. Accessed: 2023-05-03.
- [11] Google. Google documents. <https://www.google.com/docs/about/>. Accessed: 2023-05-03.
- [12] Visual Paradigm International Ltd. Visual paradigm online. <https://www.visual-paradigm.com>. Accessed: 2023-05-04.
- [13] Unity Manual. Bloom universal rp 7.1.8. <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@7.1/manual/post-processing-bloom.html>. Accessed: 2023-05-04.
- [14] Microsoft. Visual studio ide. <https://visualstudio.microsoft.com/es/>. Accessed: 2023-05-03.

- [15] Jordan Ottesen. Loopable video game music (volume 1). <https://jottesen.itch.io/loopable-video-game-music>. Accessed: 2023-05-08.
- [16] Overleaf. Overleaf latex editor. <https://www.overleaf.com>. Accessed: 2023-05-03.
- [17] Shapeforms Audio Rich. Shapeforms audio free sound effects. <https://shapeforms.itch.io/shapeforms-audio-free-sfx>. Accessed: 2023-05-08.
- [18] Unity Technologies. Unity. <https://unity.com/es>. Accessed: 2023-05-03.



## APPENDIX

### LIST OF FIGURES

2.1	Gantt chart depicting the tasks that have been done (made with Gantt Project)	8
3.1	Small sample of the artistic direction of Bullet Ballet.	12
3.2	Example of "Lucky Cat"'s projectile shining particle effect.	13
3.3	Example of a trail left behind by the "Executor"'s projectile.	13
3.4	Illumination cast by "Railgun"'s ray.	13
3.5	A few frames from the Main Menu animation.	15
3.6	Game's loading screen.	16
3.7	Screenshot with all of the game HUD elements in display.	17
3.8	Example of a possible bullet pick screen.	18
3.9	Example with a possible stat upgrade screen.	18
3.10	Game's pause menu.	19
3.11	Game over screen scenario.	19
3.12	Case use diagram (designed thanks to Visual Paradigm [12]).	26
4.1	Screenshots from some of the games that served as artistic inspiration.	28
4.2	Pistol Bullet	33

4.3	Sniper Bullet . . . . .	34
4.4	Shotgun Bullet . . . . .	35
4.5	Lucky Cat Bullet . . . . .	36
4.6	Executor Bullet . . . . .	37
4.7	Shock Therapy Bullet . . . . .	38
4.8	Artificial Violence Bullet . . . . .	39
4.9	Fight The Power Bullet . . . . .	40
4.10	Plasmarang Bullet . . . . .	41
4.11	Rock & Stones Bullet . . . . .	42
4.12	Bee Gone Bullet . . . . .	43
4.13	Railgun Bullet . . . . .	44
4.14	Shooting Star Bullet . . . . .	45
4.15	Bangerbox Bullet . . . . .	46
4.16	Sneaky Snake Bullet . . . . .	47
4.17	Lethal Injection Bullet . . . . .	48
4.18	Antidote Bullet . . . . .	49
4.19	Like A Samurai Bullet . . . . .	50
4.20	The four enemy types that the player may encounter. . . . .	51
4.21	Every container and item available in the game. . . . .	52
5.1	Collage of some of the scores sent by the beta testers. . . . .	57

## LIST OF TABLES

3.1	Functional requirement «CU1. PLAY GAME» . . . . .	22
3.2	Functional requirement «CU2. PAUSE GAME» . . . . .	22
3.3	Functional requirement «CU3. QUIT GAME» . . . . .	23
3.4	Functional requirement «CU4. RETURN TO MAIN MENU» . . . . .	23
3.5	Functional requirement «CU5. MOVE» . . . . .	24
3.6	Functional requirement «CU6. SHOOT» . . . . .	24
3.7	Functional requirement «CU7. PICK UP ITEMS» . . . . .	25
3.8	Functional requirement «CU5. GET NEW BULLETS» . . . . .	25
3.9	Functional requirement «CU9. POWER UP STATS» . . . . .	26
4.1	Possible stat upgrades and their scalings depending on rarity. . . . .	30
4.2	Every enemy appearance rate in function of the current game time. . . . .	52
4.3	Chances of each possible item to be dropped upon container destruction. . . . .	53



## SOURCE CODE

The full project contains more than 50 different scripts. To get a better grasp on how Bullet Ballet's systems work, it is highly advised to check the complete source code if needed. The link that leads to the full source code of the project can be found in the previous chapter. As a small guide, some of the core scripts that could be of interest are the following:

- **Projectile.cs** - the foundation of every bullet.
- **EnemyStats.cs** - the foundation of every enemy.
- **EnemySpawner.cs** - enemy appearances manager.
- **GameStats.cs** - character stats and game flow manager.
- **DraggableItem.cs** - bullet pick interface manager.
- **PowerUpButton.cs** - stat upgrade interface manager.

