Luo Yang 20099902 Software System Practice

Supervisor: Deirdre O’Halloran

Abstract

Infinite Dungeoner: A rogue-like bullet-hell top-down shooter game

Infinite dungeoner

Final Year Project

Table of Contents

[1 Game Overview 4](#_Toc133431068)

[1.1 Game Concept 4](#_Toc133431069)

[1.2 Genre 4](#_Toc133431070)

[1.3 Target Audience 4](#_Toc133431071)

[1.4 Game Flow Summary 5](#_Toc133431072)

[1.5 Look and Feel 5](#_Toc133431073)

[1.6 Project Scope 5](#_Toc133431074)

[1.6.1 Number of Levels 5](#_Toc133431075)

[1.6.2 Number of Locations 5](#_Toc133431076)

[1.6.3 Number of Weapons 6](#_Toc133431077)

[1.6.4 Number of Enemies 6](#_Toc133431078)

[1.6.5 Number of Supplies 7](#_Toc133431079)

[2 Gameplay and Mechanics 8](#_Toc133431080)

[2.1 Gameplay 8](#_Toc133431081)

[2.1.1 Game Progression 8](#_Toc133431082)

[2.1.2 Mission/Challenge Structure 8](#_Toc133431083)

[2.1.3 Objectives 8](#_Toc133431084)

[2.1.4 Play Flow 9](#_Toc133431085)

[2.2 Mechanics 9](#_Toc133431086)

[2.2.1 Physics 9](#_Toc133431087)

[2.2.2 Movement 9](#_Toc133431088)

[2.2.3 Objects 10](#_Toc133431089)

[2.2.4 Actions 10](#_Toc133431090)

[2.2.5 Combat 10](#_Toc133431091)

[2.2.6 Immunity 11](#_Toc133431092)

[2.2.7 Score 11](#_Toc133431093)

[2.2.8 Collision 11](#_Toc133431094)

[2.2.9 Chests 11](#_Toc133431095)

[2.3 Screen Flow 12](#_Toc133431096)

[2.3.1 Main Menu Scene 12](#_Toc133431097)

[2.3.2 Instructions Scene 12](#_Toc133431098)

[2.3.3 High Score Scene 12](#_Toc133431099)

[2.3.4 Character Selector Scene 13](#_Toc133431100)

[2.3.5 Main Game Scene 13](#_Toc133431101)

[2.4 Game Options 13](#_Toc133431102)

[2.5 Replaying and Saving 14](#_Toc133431103)

[2.6 Cheats and Easter Eggs 14](#_Toc133431104)

[2.6.1 Cheats 14](#_Toc133431105)

[2.6.2 Easter Eggs 14](#_Toc133431106)

[3 Story, Setting and Character 15](#_Toc133431107)

[3.1 Story and Narrative 15](#_Toc133431108)

[3.1.1 Source of Game Inspiration 15](#_Toc133431109)

[3.1.2 Back Story 15](#_Toc133431110)

[3.2 Game World 16](#_Toc133431111)

[3.2.1 General Look and Feel of World 16](#_Toc133431112)

[3.2.2 Area#1 Catacomb 16](#_Toc133431113)

[3.2.3 Area#2 Sorcery 17](#_Toc133431114)

[3.3 Characters 17](#_Toc133431115)

[3.3.1 Character#1 – The General 17](#_Toc133431116)

[3.3.2 Character#2 – The Scientist 17](#_Toc133431117)

[3.3.3 Character#3 – The Thief 18](#_Toc133431118)

[3.3.4 Character#4 – The Cheater 18](#_Toc133431119)

[4 Levels 18](#_Toc133431120)

[5 Technical 19](#_Toc133431121)

[5.1 Target Hardware 19](#_Toc133431122)

[5.2 Development Hardware and Software 19](#_Toc133431123)

[5.2.1 Development Hardware 19](#_Toc133431124)

[5.2.2 Development Software 19](#_Toc133431125)

[5.3 Development Procedures and Standards 20](#_Toc133431126)

[5.3.1 Development Procedures 20](#_Toc133431127)

[5.3.2 Development Standards 20](#_Toc133431128)

[5.4 Game Engine 20](#_Toc133431129)

[5.5 Network 20](#_Toc133431130)

[5.6 Scripting Language 20](#_Toc133431131)

[6 Interface 21](#_Toc133431132)

[6.1 Visual System 21](#_Toc133431133)

[6.1.1 HUD 21](#_Toc133431134)

[6.1.2 Menus 21](#_Toc133431135)

[6.1.3 Rendering System 21](#_Toc133431136)

[6.1.4 Camera 21](#_Toc133431137)

[6.1.5 Light Models 21](#_Toc133431138)

[6.2 Control System 22](#_Toc133431139)

[6.3 Audio 22](#_Toc133431140)

[6.4 Music 22](#_Toc133431141)

[6.5 Sound Effects 22](#_Toc133431142)

[7 Artificial Intelligence 22](#_Toc133431143)

[7.1 Enemy Movement AI 23](#_Toc133431144)

[7.2 Enemy Weapon AI 23](#_Toc133431145)

[8 Game Art 23](#_Toc133431146)

[9 Management 24](#_Toc133431147)

[9.1 Schedule 24](#_Toc133431148)

[9.1.1 Prototype 1 – October and November, 2022 24](#_Toc133431149)

[9.1.2 Prototype 2- December, 2022 24](#_Toc133431150)

[9.1.3 Prototype 3- January, 2022 24](#_Toc133431151)

[9.1.4 Prototype 4- February, 2022 25](#_Toc133431152)

[9.1.5 Prototype 5 – March, 2022 25](#_Toc133431153)

[9.1.6 Prototype 6 – April, 2022 25](#_Toc133431154)

[9.2 Version Control 25](#_Toc133431155)

[9.3 Test 26](#_Toc133431156)

[9.3.1 Test Schedule 26](#_Toc133431157)

[9.3.2 Bug Log 26](#_Toc133431158)

[9.3.3 Tests and Users Feedback 27](#_Toc133431159)

[10 Reflection 28](#_Toc133431160)

[10.1 What went right and what went wrong 28](#_Toc133431161)

[10.2 What I learned from the process 29](#_Toc133431162)

[10.3 What I would do differently if I had to do it all over again 29](#_Toc133431163)

[10.4 What I would do if I had more time to develop this project further 30](#_Toc133431164)

[10.5 What were the most challenging features to implement 30](#_Toc133431165)

[11 Appendices 31](#_Toc133431166)

[11.1 Material Source 31](#_Toc133431167)

[11.2 References 31](#_Toc133431168)

[11.3 Bibliography 32](#_Toc133431169)

[11.3.1 Udemy Courses 32](#_Toc133431170)

[11.3.2 YouTube Videos 32](#_Toc133431171)

[11.3.3 Bilibili Videos (A Chinese video platform) 32](#_Toc133431172)

[11.4 Project Video Log 32](#_Toc133431173)

[11.5 User Manual 32](#_Toc133431174)

[11.6 Game Online Deployment Link 33](#_Toc133431175)

[11.7 Version Control 33](#_Toc133431176)

[11.8 Balance of the value in the game 33](#_Toc133431177)

Table of Figures

[Figure 1 Logical Model of Maps 6](#_Toc133330109)

[Figure 2 Main Menu Scene 12](#_Toc133330110)

[Figure 3 Instructions Scene 12](#_Toc133330111)

[Figure 4 High Score Scene 13](#_Toc133330112)

[Figure 5 Character Selector Scene 13](#_Toc133330113)

[Figure 6 Pause Menu 14](#_Toc133330114)

[Figure 7 Character#1 The General 17](#_Toc133330115)

[Figure 8 Character#2 The Scientist 18](#_Toc133330116)

[Figure 9 Character#3 The Thief 18](#_Toc133330117)

[Figure 10 Example of Characters 23](#_Toc133330118)

[Figure 11 Example of Weapons 24](#_Toc133330119)

[Figure 12 Example of Enemies 24](#_Toc133330120)

[Figure 13 Chest room with bug 26](#_Toc133330121)

[Figure 14 Chaotic corridor around chest room 27](#_Toc133330122)

# 1 Game Overview

## 1.1 Game Concept

The game concept of Infinite Dungeoner is a rogue-like bullet-hell top-down (2.5D) shooter, where players can go to each room by moving up, down, left, and right. Enemies will spawn if the player enters a room for the first time, unless entering a chest room. While the player is fighting enemies, the doors of the room will be locked, and will only unlock after defeating all enemies. The door to the boss room will not open until all rooms containing enemies on current level are cleared. After defeating the boss of this level, press the Enter key to enter the next level.

After the player enters a room with enemies or defeats all enemies in the current room, a chest may be spawned. Defeating the boss and the chest room will definitely spawn a chest. There are three types of items that will be generated in the box, which are recovery potions, ammo, and new weapons. Recovery Potions and Ammo will restore the player's current HP and ammo of the current weapon according to the ratio displayed.

Players can get points from kills, and continuous kills will increase the multiplier. But if the player is attacked by an enemy, the multiplier will be reset to 1. This is designed to encourage players to dodge enemy bullets as much as possible while killing more enemies.

The game will use many free assets. There are inevitably some repetitive features or gameplay in games, but this is the situation for most games of this type. I will try to add elements that attract players as much as possible, and mix excellent features from famous games. For example, a large number of weapons with interesting appearances, dynamic difficulty feature based on the player's current state.

## 1.2 Genre

(1) Role Playing, (2) Roguelike, (3) Single Player, (4) Offline, (5) Bullet hell, (6) Top-down perspective. (7) Shooter

## 1.3 Target Audience

Everyone 10+ (some violence included in the game). The game contains a variety of game elements, I think players who like rogue-like or dungeon exploration games will be attracted. The game is difficult to get started due to the bullet-hell element. Combined with score leaderboard system, some experienced players who seek difficulty may also be attracted.

## 1.4 Game Flow Summary

The main menu will be displayed at the beginning of the game. Players can select to “Start New Game”, “Enter their name”, “Select characters”, “Read Instructions”, “Check High Scores” there. In the game, players need to defeat all the enemies in each room, and then defeat the level boss. During the adventure process, they can find new weapons and supplies. After defeating the boss, they can move to the next level.

## 1.5 Look and Feel

This is a top-down game, which means it’s a 2D game that provides a overhead or bird’s-eye view of the action. This is a good way to create a 3D feel in a 2D game. Due to the limited quantity and quality of art materials, the game will look and feel like a 00s game. In general, the game will be similar to “Enter the gungeon”, “The Binding of Isaac”, “Soul Knight”. (XoLo921, 2023)

## 1.6 Project Scope

### 1.6.1 Number of Levels

There are a total of 6 levels in the game, and the difficulty will gradually increase. The number of monsters in each room will be more, the HP of monsters will be higher, and the weapons used by monsters will be more powerful. The room styles of the first three levels and the last three levels are the same.

### 1.6.2 Number of Locations

Locations will be procedurally randomly generated based on their characteristics and make up the map for each level.

As the level increases, the size of the map will also increase, which is intuitively reflected in the increase in the number of rooms that must pass from the starting point of each level to the level boss room.

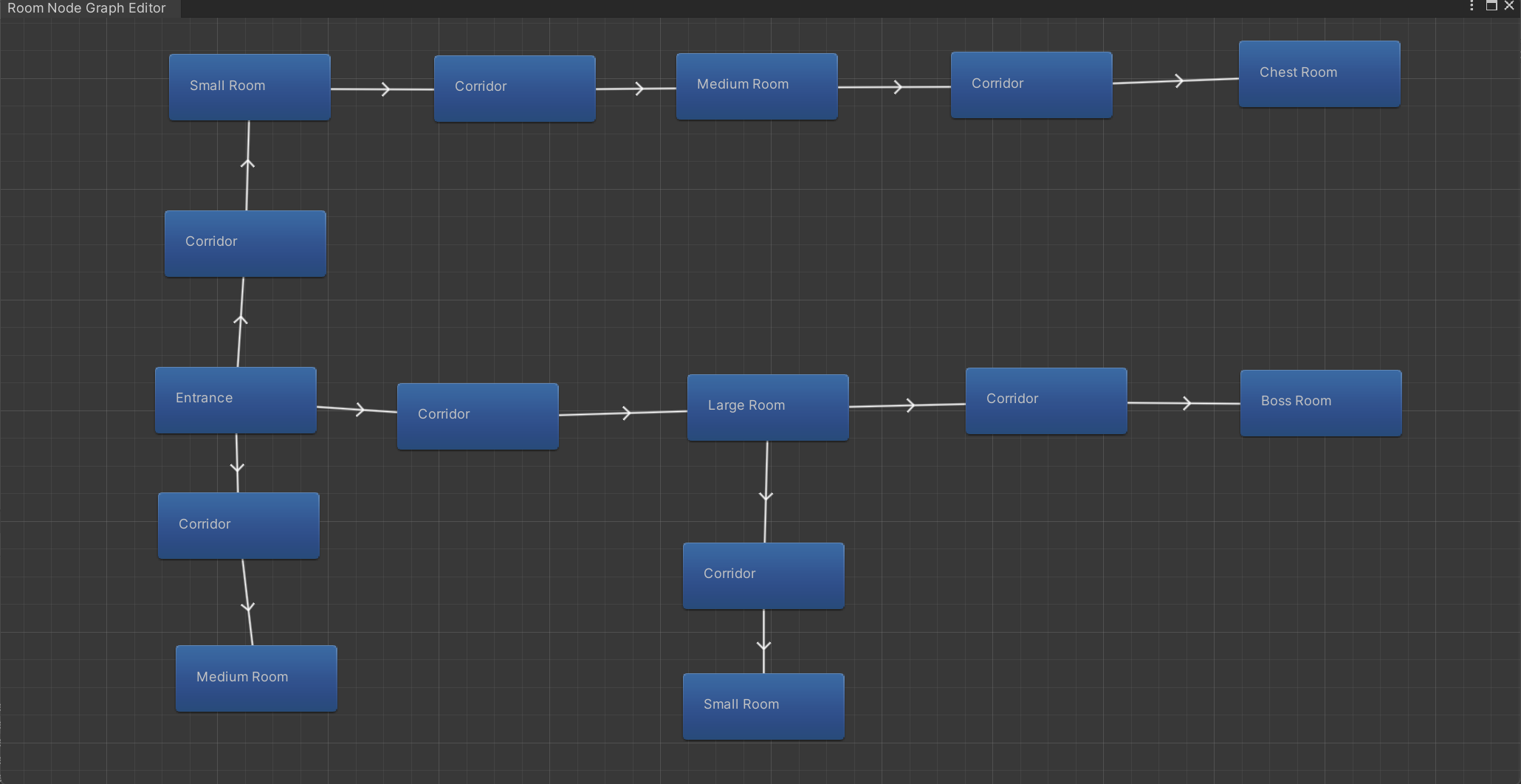


Figure 1 Logical Model of Maps

* Entrance: The start point for each level. The player always stands in the center of the entrance whenever they start a new level.
* Corridor: The corridor connects two rooms, no enemies and chests will be generated in the corridor
* Small Room: Small number of total enemies, small number of concurrent enemies, large spawn interval, low chest spawn rate.
* Medium Room: Medium number of total enemies, medium number of concurrent enemies, medium spawn interval, medium chest spawn rate.
* Large Room: Large number of total enemies, large number of concurrent enemies, small spawn interval, high chest spawn rate.
* Boss Room: Generate different numbers and types of BOSS based on the current number of levels. A chest will always be generated and must contain recovery potions, ammo, and new weapon.
* Chest Room: A chest will always be generated, including at least two of recovery potions, ammo, and new weapon.

### 1.6.3 Number of Weapons

27 player weapons and 8 enemy weapons have been implemented. Currently, player weapons include Sniper, MP7, Laser Blaster, Pistol, Revolver, Shotgun, Rocket Launcher, QBZ, Laser Pistol. In addition to the above-mentioned conventional firearms, enemy weapons also have staffs and magic that fire a large number of bullets at once, which is the embodiment of bullet hell.

### 1.6.4 Number of Enemies

There are a total of 16 types of enemies in the game, 6 of which are bosses.

Common enemies are:

* Grimonk: Masked enemy with QBZ that look like ghosts and only appear in the last three floors.
* Blue Medusa: Blue variety of Medusa, use magic to cast snake-shaped and star-shaped bullets, appear in the last three levels.
* Green Medusa: Green variety of Medusa, don’t use any weapons, damage by contact, appear in every level.
* Red Medusa: Red variety of Medusa, use magic to cast snake-shaped and star-shaped bullets, only doesn’t appear in the first level, and the movement speed is slower than other varieties of Medusa.
* Mud Rock: Stone man, use QBZ weapon, high hp, slow movement speed, appear in the last three levels.
* Orc: Orc, use QBZ weapon, high hp, slow movement speed, appear in the last three levels.
* Blue Slime Block: Blue variety of Slime Block, use Shotgun weapon, only doesn’t appear in the first level.
* Green Slime Block: Green variety of Slime Block, use QBZ weapon, appear in every level.
* Red Slime Block: Red variety of Slime Block, use Pistol weapon, appear in the last three levels. Higher movement speed than other varieties of Slime Block.
* Slizzard: Looks like purple eggplant, use staff to cast triangle-shaped and square-shaped bullets, appear in the last three levels.

Boss are:

* Blue Slime Block King: Blue variety of Slime Block, use MP7 weapon, may be one of the bosses of the third level.
* Green Slime Block King: Green variety of Slime Block, use Shotgun weapon, may be one of the bosses of the second or third level.
* Red Slime Block King: Red variety of Slime Block, use Pistol weapon, may be one of the bosses of the first, second or third level.
* Blue Skull Face: Blue variety of Skull Face, use Pistol weapon, may be one of the bosses of the fourth, fifth or sixth level.
* Green Skull Face: Green variety of Skull Face, use Shotgun weapon, may be one of the bosses of the fifth or sixth level.
* White Skull Face: Blue variety of Skull Face, use MP7 weapon, may be one of the bosses of the sixth level.

### 1.6.5 Number of Supplies

Players can get three types of supplies from chests, they are recovery potions, bullets and, new weapons. There will be a percentage value displayed below the available supplies. The recovery potion will restore the player's HP according to the displayed percentage. Ammo will restore the weapon currently held by the player according to the displayed percentage. The value of the supplies is determined by many factors, including but not limited to the room type, the spawn condition (generated after entering or cleared), the number of current levels, and the current state of the player.

# 2 Gameplay and Mechanics

## 2.1 Gameplay

### 2.1.1 Game Progression

In this game, players need to reach the deepest level of the dungeon, defeat all levels’ bosses and try to get score as much as possible. During the process of getting to the bottom, players need to clear each level and for each level, they need to clear each room. After entering any room with enemies, the doors will be closed. Only after all enemies have been eliminated, the doors of the room can be unlocked. Usually, a series of such rooms are experienced before reaching the level boss’s room. After defeating the level boss, they can press the “Enter” key to enter the next level. Players will repeat this until they meet the final Boss. After they defeat the final bosses, their name and score will be recorded. However, if players lose all health points before they defeat the final bosses, they need to restart the game from the entrance of the dungeon with nothing, this is one of the characteristics of rogue-like.

### 2.1.2 Mission/Challenge Structure

Players need to clear all rooms containing enemies in the current level before they can go to the boss room. After defeating the BOSS, you can go to the next level.

The number of enemies spawned in each room is determined by the current level and room type. Normally, the higher the current level, the more enemies there are in the room. At the same time, the number of enemies in large rooms is greater than the number of enemies in medium rooms, the number of enemies in medium rooms is greater than the number of enemies in small rooms. After the player enters the room or clears the room, there may be a chest generated. But usually, chests that spawn upon entering a room will contain lower quality supplies than chests from defeating enemies.

The spawning position of the enemy is not completely random, it will be randomly generated from the pre-set position, which helps to avoid unreasonable room layout and enemies spawning in unreasonable positions.

### 2.1.3 Objectives

Defeat the enemies in each room and reach the boss of the current level. Repeat until the final level and defeat the final boss

### 2.1.4 Play Flow

The player will start the game at the entrance of the dungeon. After entering the dungeon, there is the first level. The player will stand at the center of the start room of the first level, and then they can enter the next room and defeat enemies inside. Finally, they will find the boss room on the first level. After they defeat the boss, they can press the "Enter" key to move to the next level. They will repeat the process of clearing each level before they reach the final level and fight the final boss. After defeating the final boss, "You Win" will pop up and their score will be recorded in the leaderboard and the game will be reset to the main menu scene. Also, if the player dies in the dungeon, after the game over popping up, their score will be recorded and the game will be reset too.

## 2.2 Mechanics

### 2.2.1 Physics

This is a 2D game, so it doesn’t reflect the height difference and hence the bullet won’t fall but they still have the range, if they reach the range they will disappear. Others are basically the same as the real universe, if the player shoots while moving, the bullet will have the same deflection. Some weapons may have accuracy issues, there may be a certain degree of bias. Also, some obstacles in the game can resist bullets, such as walls, and wooden boxes. But they can be destroyed if they are fragile like wood products or glass bottles.

The game uses Unity's built-in Rigidbody 2D component, which is used to place objects under the control of the physics engine so that they are properly accelerated and affected by collisions and forces. With the Rigidbody 2D, I created some movable objects in the game, such as tables and chests. Players can move these items by pushing them. (Unity, 2020)

### 2.2.2 Movement

#### Player Movement

Players can use “WASD” to control the character to move up, down, left and right. If the player encounters an object, such as walls, statues, that is set to be impassable while moving, they will be stopped.

Players can use the right mouse button to perform a dodge roll. Cannot shoot while rolling but being immunity, which means they won’t be hurt by enemies or enemies’ ammo.

#### Enemy Movement

Enemy movement is controlled by “EnemyMovementAI” class. In this class, A\* pathfinding is implemented. The program itself will provide each enemy with the closest path from the enemy itself to the player. If the distance of the path is less than the tracking range attribute of enemy itself, the enemy will move towards the player along the path. (Vozul, 2023)

In addition, there are some movable objects like tables and chests in the game. They will be recognized as obstacles by the enemy's movement AI, and the dynamically updated obstacle positions will also be calculated when updating the path.

### 2.2.3 Objects

Objects that can be interacted with in the game are mainly divided into the following three categories: decorative objects, spawned objects and moveable objects.

**2.2.3.1 Decorative Objects**

These objects are mainly used for room decoration and are usually placed in the corner or near the wall. Players can destroy these decorations by contacting or shooting them. At the same time, these fragile decorations can also help players block an enemy bullet.

**2.2.3.2 Spawned Objects**

Objects in this category mainly refer to supplies spawned from chests. If it is a recovery potion or ammo, the percentage value will be displayed below the object image. If the player gets close to them and presses the “E” key, it will restore health or the ammo of the currently held weapon according to the displayed percentage value.

**2.2.3.3 Moveable Objects**

There are two types of movable objects in the game, they are overturned tables and chests. Players can use these objects to dodge enemy bullets, just like in cowboy movies.

### 2.2.4 Actions

Players mainly interact with various objects in the game through the “E” key. It should be noted that the weapons held by the player have colliders. The interaction may not be successful due to insufficient distance.

Players can hold multiple weapons. They can switch weapons through the number keys and mouse wheel. Using ‘-’ key to set the current weapon to position 1.

### 2.2.5 Combat

Each character has an initial weapon with unlimited ammo. Whenever the player first enters a room that is not a corridor or chest room or entrance, all doors of that room will be locked, the background music will change, and enemies will start spawning. Only when the player kills all the enemies, the door of the room will be unlocked, and the background music will return to normal. Chests will have a chance to be generated when the player first enters the room, or after the room is cleared.

The boss battle will be triggered in the boss room. According to the current level, a specific number and type of boss will be generated. After defeating the boss, a chest will be generated. And then players can press “Enter” key to enter the next level. They can loot the supplies left in the current level before entering the next level.

### 2.2.6 Immunity

Players are immune while performing a dodge rolling. If they got hit by an enemy, they will also be immune for a few seconds. This is designed to prevent sudden death. As the number of levels increases, the number of enemies and the scale of the bullet-hell will be larger. Without the setting of the immune mechanism, the player will die easily and quickly.

### 2.2.7 Score

Players can earn points during the game. The score obtained follows this formula: “Enemy’s HP” \* “Multiplier”. The health point of the enemy is fixed, and the part that the player can influence is the multiplier. The setting of the multiplier is that if the player kills the enemy, the multiplier will be increased by 1, and if the player is damaged, the multiplier will be decreased by 1. This is intended to encourage players to kill enemies while taking as little damage as possible.

### 2.2.8 Collision

The game uses the Circle Collider 2D and Polygon Collider 2D components provided by Unity to deal with collision detection. (Unity, 2021) These components are used to monitor the contact between game objects. For example, character in contact with impassable terrain, Bullets collide with players, enemies, and various objects in the room. (Unity, 2021)

### 2.2.9 Chests

Players can press E to open the chests, they can get hp restore potions, ammo, and new weapons from the chests. In the game, I introduced the design of dynamic difficulty. The content of the chest is not only affected by the number of levels, room type and generation method, but also the state of the player. If low hp, more potions, if low ammo, more ammo. If weapons are bad, generate a better weapon. Normally, a high-quality chest will be generated in the chest room and cleared boss room. While in small room, medium room, or large room, there is a chance to generate chest when the player first enters the room and after clearing the room. But the quality of the former is often lower than the latter

## 2.3 Screen Flow

In the Unity engine, the game usually only uses one screen, but it will switch between multiple scenes. The scenes used in this game are as follows

### 2.3.1 Main Menu Scene

Main menu scene is the entrance of the entire game program. The character selector scene is embedded in this scene by default. This scene also contains four buttons for exiting the game, jumping to the main game scene (Start the game), jumping to the instructions scene, and jumping to the high score scene.

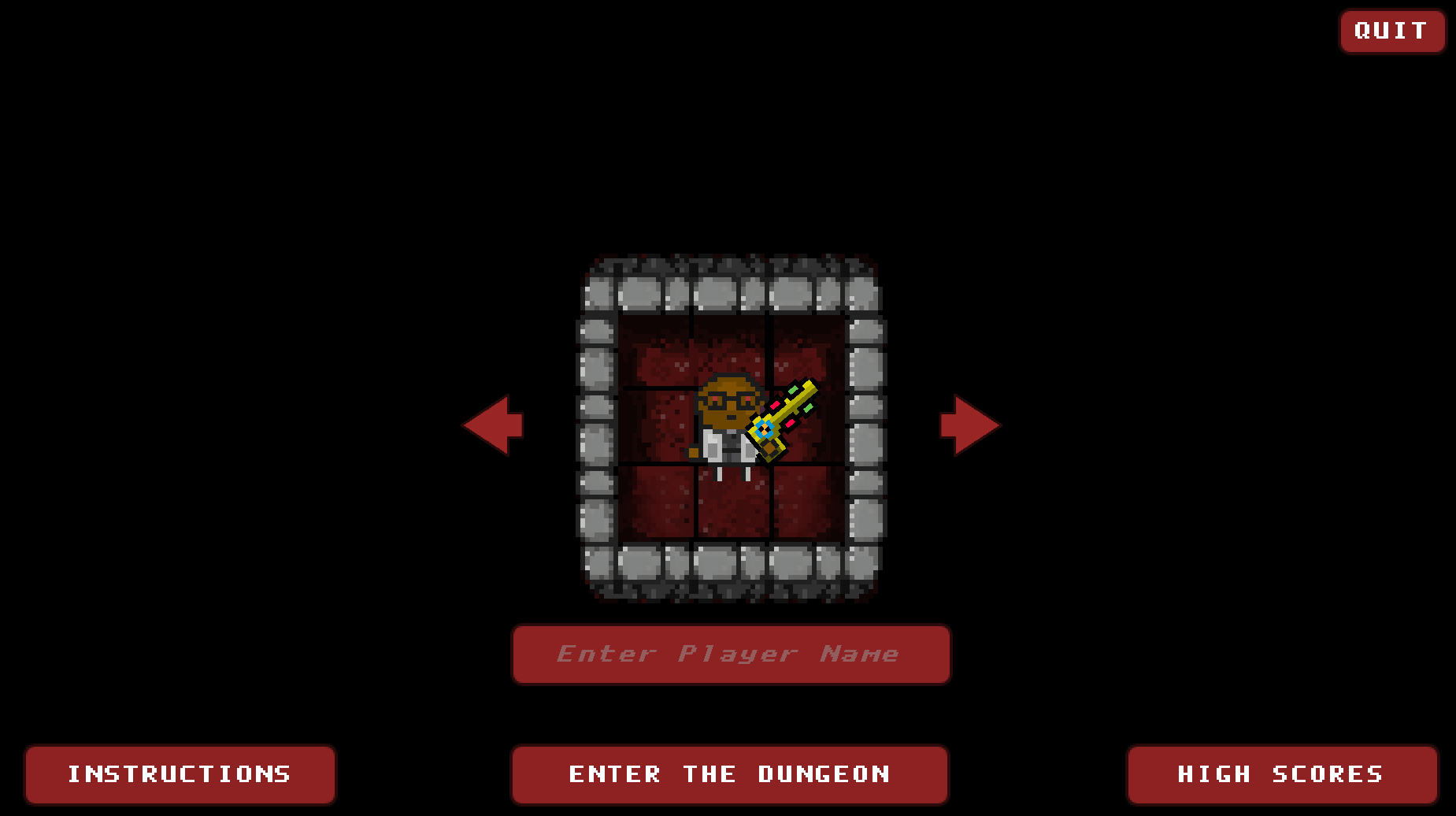


Figure 2 Main Menu Scene

### 2.3.2 Instructions Scene

This scene is mainly used to introduce the gameplay to the player. Players can see the operation method and precautions of the game here. Also, this scene contains a button used for returning back to the main menu scene.

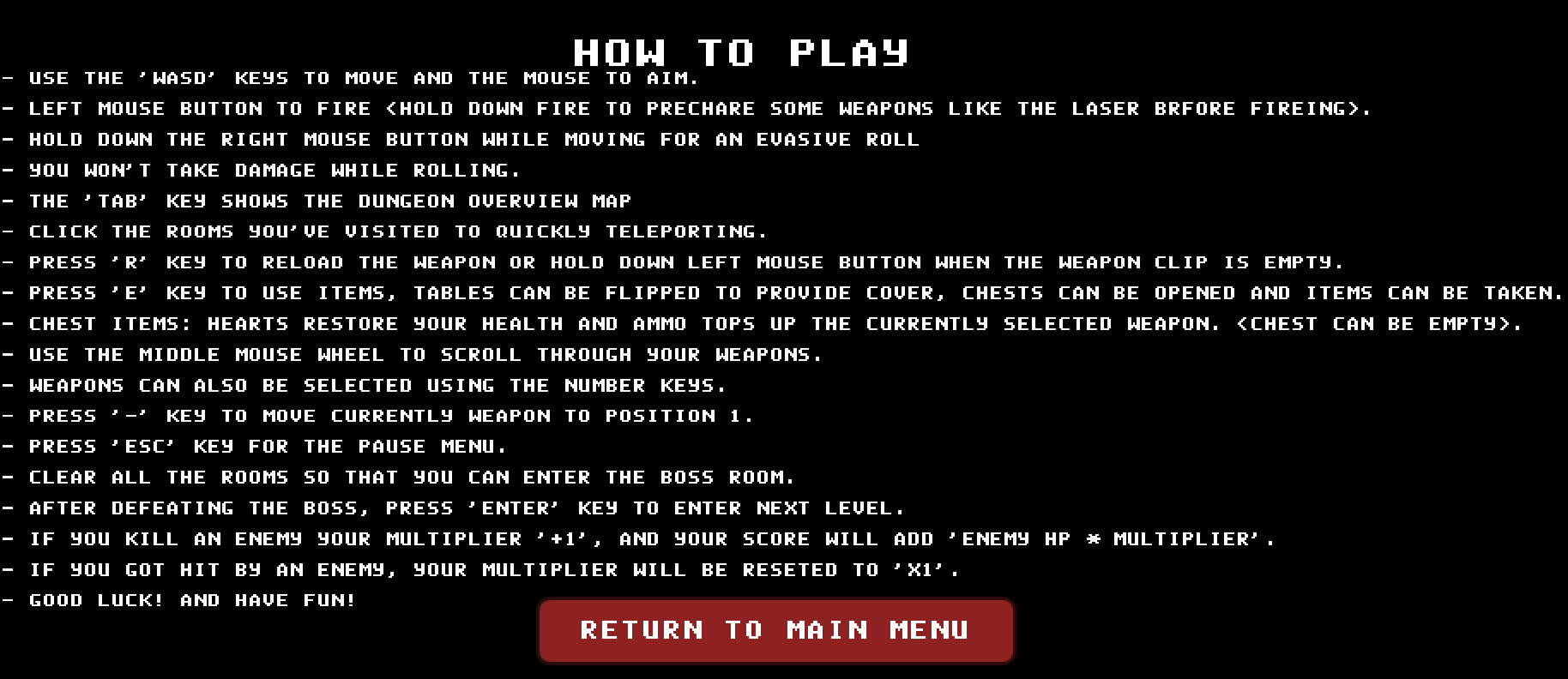


Figure 3 Instructions Scene

### 2.3.3 High Score Scene

This scene is mainly used to show the leaderboard to players. It includes a table, showing the player's ranking, name, level reached, score. Also, this scene contains a button used for returning back to the main menu scene.



Figure 4 High Score Scene

### 2.3.4 Character Selector Scene

This scene is embedded in the main menu scene and is in the middle. It contains two buttons used for switching characters. Players can see the appearance and initial weapon of the characters here.



Figure 5 Character Selector Scene

### 2.3.5 Main Game Scene

This scene is the main scene of the game. All game-related activities take place here. In this scene players can press “Esc” key to bring up the pause menu. In the pause menu, player can change the volume of sounds and music or restart the game. In this menu, I used the PlayerPrefs class, which is similar to the LocalStorage class in JavaScript, provided by unity. It will record the volume value adjusted by the player, and automatically apply the recorded value after starting the game again.

## 2.4 Game Options

The Game Options is the pause menu in the main game scene, which allows the player to change the sound and music volume, or on and off.

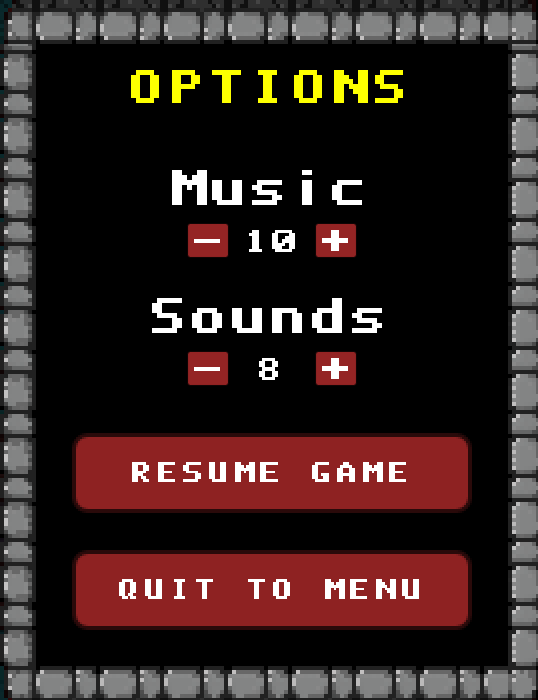


Figure 6 Pause Menu

## 2.5 Replaying and Saving

The save and replay mechanism is not implemented in this game, which was abandoned after a certain trade-off.

The save and replay mechanism of Rogue-like games is mainly implemented based on seeds, which is a string of number. All content in a round will be generated based on the seed of the round. This means that the game itself only needs to record a few things, such as the equipment held by the player, the number of current levels, and the rooms that have been cleared.

But all of this is based on the seed algorithm. Since I don't have professional algorithm knowledge, it takes a lot of time to study this mechanism, which will delay the progress of the project.

At present, if there is a need to implement this mechanism, it is also necessary to refactor the codes such as room generation, enemy generation, and treasure chest generation. Due to the deadline, I chose to give up implementing this mechanism.

## 2.6 Cheats and Easter Eggs

### 2.6.1 Cheats

In the game, I added another character, which has the same appearance as the first character, “The General”. He has unlimited health and a weapon with extreme high damage and firing rate.

### 2.6.2 Easter Eggs

Easter eggs in the game will appear in the form of weapons. Players have a chance to obtain some very interesting weapons, such as mailboxes that emit mail, holy swords that emit sword energy, etc.

# 3 Story, Setting and Character

## 3.1 Story and Narrative

### 3.1.1 Source of Game Inspiration

Rogue-like games are a popular genre of video games that feature procedurally generated levels, permadeath, and high difficulty. One of the most interesting subgenres of Rogue-like games is the dungeon shooter. These games combine the exploration and loot aspects of traditional Rogue-likes with fast-paced shooting action and bullet hell scenarios. Dungeon shooters often have a sci-fi or cyberpunk aesthetic, but some also come from horror or fantasy genres.

I have played "Enter the Gungeon", "The Binding of Isaac", "Soul Knight" which are very popular Rogue-like dungeon shooter games nowadays, and my total time on these three games is even close to 500 hours (Rkieferbaum, 2023) (Botto, 2022). They are the main source of inspiration for my games. What made me decide to choose this type of game as my final year project was that I saw the games made by other students in a class that used game as an example to learn programming a year ago in my Chinese university.

Roguelike games are highly procedurally based on automatic generation. From a programming standpoint, developing a game like this would be a great opportunity to practice what I've learned about programming. And I only used Java and Python to develop some simple games before, this time I chose to use the Unity engine to try more professional game development.

At the same time, this type of game is also compatible with a variety of background stories, no matter how simple or complex the story is. It can be like "The Binding of Isaac" to describe the story of a tragic family through a large number of props, monsters, animations and other details, it can be also like "Soul Knight", which doesn’t contain any plot and just focus on the gameplay.

### 3.1.2 Back Story

I originally built a complex backstory based on Ender's game and Truman's world, but due to the difficulty of making suitable animations, weapons, enemies, etc., I finally chose to give up, but I will show it.

**3.1.2.1 Simple Story**

One day, a dungeon suddenly appeared on the earth. No one knew where it came from, and no one knew its purpose. None of the people who tried to enter the dungeon returned. But as time passed, the area of the dungeon became larger and larger, and some monsters even began to try to go outside the dungeon. People realize that this is an unprecedented crisis for human beings, and they need to find real heroes to conquer this dungeon and wipe out the monsters in it.

**3.1.2.2 Complex Story**

In the complex version of the story, every time the players pass through the dungeon, a strange vortex portal will appear. The player will be force-sucked and wake up at the dungeon entrance. This fits with the Rogue-like’s nature and “Infinite” in the title of the game.

In the process of game, players will gradually encounter more and more abnormal things, such as special magic pieces of paper, implying that this is a weapon in the virtual world, enemies with wrong textures, etc. After certain conditions are met, players can avoid the forced teleportation after clearing the level and enter the truth plot.

After escaping the vortex, the character wakes up from the training pod. Players will be told the truth about the game. The earth has been invaded by aliens, and the process of exploring the dungeon before is a simulation of fighting aliens. In order to resist the invasion, some candidates were selected and sent to the training pod. The character controlled by the player is the best candidate to be the first to wake up from it, and will become the ultimate hero against the aliens.

At this point the title of the game will be converted to Final Dungeoner, and the player will play the final round of the game. If it fails at this stage, the game will fall back to the original version and tell the player "Hero failed. Humanity lost the earth". If successfully cleared, players will still be able to play, but their character will wear a crown.

In the complex version of the story, I tried to imitate "The Binding of Isaac" and combine elements related to the plot with the game, such as rooms with puzzles, weapons that hint at the truth, and so on. But after trying to make relevant materials, due to the huge workload and lack of artistic expertise, I chose to abandon this idea.

## 3.2 Game World

### 3.2.1 General Look and Feel of World

In the selection of map and monster materials, I chose materials with horror elements, so the overall style of the game will look slightly gloomy and fearful. But because it is a 2D game, enemies and maps do not bring too much fear to players.

### 3.2.2 Area#1 Catacomb

**3.2.2.1 General Description**

Here is the shallow part of the dungeon. Although the monsters here are not powerful, they still cannot be ignored. The previous explorers were buried here, and in the depths of the area, there are giant slime kings waiting for players.

**3.2.2.2 Physical Characteristics**

The monsters here are less difficult to deal with, and the number is smaller, and there is only a low probability of dropping powerful weapons in the treasure chest.

**3.2.2.3 Levels that use area#1**

From Level 1 to Level 3.

### 3.2.3 Area#2 Sorcery

**3.2.3.1 General Description**

You have reached the deepest area of the dungeon, which is seriously affected by the dungeon, where the power of witchcraft is rampant. And there are more and more monsters that can use magic. In the deepest part, the giant skeletons manipulated by the power of witchcraft have awakened, waiting for the next challenger to come.

**3.2.3.2 Physical Characteristics**

The monsters here are more difficult to deal with and more in number, and players will get more supplies and more powerful weapons from the chests.

**3.2.3.3 Levels that use area#2**

From Level 4 to Level 6.

## 3.3 Characters

### 3.3.1 Character#1 – The General

**3.3.1.1 Back Story**

He was a general who was chosen as a candidate to save the world after retiring from the military. Rich special military combat experience, making him recognized as the best candidate.

**3.3.1.2 Personality**

Steady, brave, reliable, smart, or some other personalities that a hero should have.

**3.3.1.3 Physical characteristics**

Old man with silver hair in green military uniform and black eye patch.



Figure 7 Character#1 The General

**3.3.1.4 Special Abilities**

Carries an additional starting weapon.

### 3.3.2 Character#2 – The Scientist

**3.3.2.1 Back Story**

An unknown scientist, in the process of his experiment, suddenly a group of people broke into his laboratory and sent him to a strange dungeon.

**3.3.2.2 Personality**

Calm and analytical.

**3.3.2.3 Physical characteristics**

Bald middle-aged man in a white lab coat.



Figure 8 Character#2 The Scientist

**3.3.2.4 Special Abilities**

Carry a self-made laser gun.

### 3.3.3 Character#3 – The Thief

**3.3.3.1 Back Story**

A habitual thief, he was caught on his last theft. But instead of a prison, he was sent to a strange dungeon.

**3.3.3.2 Personality**

Cunning, greedy, cowardly.

**3.3.3.3 Physical characteristics**

Red-haired man in black with a black mask.



Figure 9 Character#3 The Thief

**3.3.3.4 Special Abilities**

Higher movement speed, shorter rolling CD, longer immunity time, less HP.

### 3.3.4 Character#4 – The Cheater

In the game, I added another character, which has the same appearance as the first character, “The General”. He has unlimited health and a weapon with extreme high damage and firing rate.

# 4 Levels

The game contains 6 levels. After defeating the BOSS of each level, the player can press the "Enter" key to enter the next level. The game will become more difficult as the number of levels increases, which will be reflected in many aspects, such as the number of rooms that players need to clear, the total number of enemies generated in the room, the upper limit of the number of enemies that exist in the room at the same time, the blood volume of monsters, and the weapons used by monsters etc.

The current level settings are as follows:

* Level 1 – The Hall of Heroes – Area #1 Catacomb
* Level 2 – The Shadow Crypt – Area #1 Catacomb
* Level 3 – The Deep Catacomb – Area #1 Catacomb
* Level 4 – Dark Magic – Area #2 Sorcery
* Level 5 – Wizards Realm – Area #2 Sorcery
* Level 6 - Sorcery & Survival – Area #2 Sorcery

# 5 Technical

## 5.1 Target Hardware

Minimum requirements to play this game:

* OS: Windows 7 or later
* Processor: Intel Core 2 Duo E6320 (2\*1866) or equivalent
* Memory: 2 GB RAM
* Graphics Card: GeForce 7600 GS (512 MB) or equivalent
* Storage: 2 GB of available space required

## 5.2 Development Hardware and Software

### 5.2.1 Development Hardware

My own laptop – Lenovo Legion Y9000P IAH7H:

* OS: Windows 11
* Processor: 12th Gen Intel(R) Core(TM) i7-12700H 2.30GHz,
* Memory: 16GM DDR5
* Graphics Card: RTX 3060
* Storage: 1TB SSD

### 5.2.2 Development Software

* Game Engine: Unity Engine
* IDE: Visual Studio
* Programming Language: C#
* Version Control: Git

## 5.3 Development Procedures and Standards

### 5.3.1 Development Procedures

Agile methodology by the name of Scrum. Scrum is a process skeleton consisting of a set of practices and predefined roles. In each sprint (a 15-30 days cycle, the length of which is determined by the development team), the development team creates an increment of usable (ready to roll out) software. The features to be implemented in each sprint come from the product order. The product order (product goal) refers to the outline requirements of the work that needs to be completed in order of priority. Which line items (goal items) are included in a sprint is determined by the sprint planning meeting. In the meeting, the product owner tells the development team which line items in the product order he needs to complete. The development team decides how many line items they can commit to in the next sprint. During a sprint, no one can change the sprint backlog, which means that the requirements are frozen during a sprint. (MrOllie, 2022)

In this project, each sprint is defined as 1-2 weeks based on the features to be implemented and the timetable.

### 5.3.2 Development Standards

No warnings or errors left after each commit. Comply with the C# code specification

## 5.4 Game Engine

Unity. The engine can be used to create 3D and 2D games. It is particularly popular for iOS and Android mobile game development, is considered easy to use for beginner developers, and is popular for indie game development. (Guerillero, 2022)

## 5.5 Network

This game is offline.

## 5.6 Scripting Language

C#. Generally, Unity game development uses C# as the main development language. Although Unity itself supports Unity (an extension of Java) and Boo (a language inspired by Python), in normal production practice, C# is still used as the main development language.

# 6 Interface

## 6.1 Visual System

### 6.1.1 HUD

Heads up display will be displayed at the corners of the screen. The upper left corner will display the player's health and score. A minimap is displayed in the upper right corner. The lower right corner displays the weapons and ammo held by the player.

### 6.1.2 Menus

The main menu will implement the following functions: (1) Selecting characters. (2) Start the game. (3) View the high score scene. (4) Quit. (5) View the instructions scene

During the game, players can press the “Esc” key to bring up the pause menu. In the pause menu, players can adjust the volume of sounds effects and background music. Using the playerfabs class provided by Unity, the volume set by the player will be automatically applied the next time the game is launched. Players can also return to the main menu through the pause menu, but this will clear all progress for the current round game.

### 6.1.3 Rendering System

Using universal render pipeline provided by Unity. (Unity, 2021)

### 6.1.4 Camera

Using cinemachine as main camera, the camera's target group includes player and cursor, which allows the game perspective to be locked on the player and also enables the player to look away from a certain distance. (Unity, 2022)

### 6.1.5 Light Models

Using global light 2D provided by Unity. In the game, there are some torches using the light models. (Unity, 2020)

## 6.2 Control System

The player interacts with the game through the keyboard and mouse.

## 6.3 Audio

Using the Audio Listener, Audio Mixer and Audio Source supported by the Unity engine. The audio in the game is mainly divided into background music and various sound effects. (Unity, 2021)

## 6.4 Music

The game will play different background music based on the current scene and game state. Fade in and fade out are also used when switching background music. Currently, the game contains 5 kinds of background music:

* Ambient Music - Play when no enemies are encountered.
* Battle Music - Play when encountering an enemy.
* Boss Music - Play when encountering a boss.
* Melancholy Music - Play when entering a chest room.
* MainMenu Music – Play when in the main menu scene.

From the perspective of code implementation, there are two classes related to sound effects in the game. They are MusicTrackScriptableObject used for storing details of the audio clips, Music Manager used for management of background music.

## 6.5 Sound Effects

All weapons and interactive objects in the game have corresponding sound effects. Sound effects for weapons include firing, reloading sound effects. For interactive objects, it usually includes dragging sound effects when moving items, overturning sound effects when overturning tables, prize sound effects when opening chests, etc.

From the perspective of code implementation, there are three classes related to sound effects in the game. They are SoundEffectScriptableObject class used for storing details of the audio clips, SoundEffectManager class used for management of sound effects, and SoundEffect class as a logical model.

# 7 Artificial Intelligence

Two AIs are implemented in the game, which are used to control the movement of enemies and use weapons respectively.

## 7.1 Enemy Movement AI

This AI is used to control the movement of enemies. In Unity 3D, the paths for enemies can be calculated by using Nav Meshes and Nav Mesh Agents, but in Unity 2D there are not built-in solutions for pathfinding. And considering the game map is based on Tilemap grid. I chose to implement the movement AI based on the A\* pathfinding algorithm.

Since there are some movable objects in the game, such as chests and tables. In the A\* pathfinding algorithm, in addition to the fixed impassable terrain, the variable obstacle is added to dynamically calculate the enemy's movement path.

What’s more, in terms of performance optimization, a certain interval is set for the construction of the enemy's movement path, which avoids consuming a lot of CPU time to prevent the game from freezing.

## 7.2 Enemy Weapon AI

This AI is used to control the weapons of enemies. The AI will automatically attack based on the enemy's weapon parameters. There are two types of enemy attack methods, one is that as long as the player is within the enemy's attack range, they will attack blindly. There are two types of enemy attack methods, one is that as long as the player is within the enemy's attack range, they will attack blindly. The other is that enemies only attack when they can attack the player. Regarding the former, this attack mode is mainly used for enemies that will release a large number of bullets. While regarding the latter, Unity's built-in Raycast class is used to check whether there is an obstacle between the enemy and the player, that is, to determine whether the enemy can attack the player.

Additionally, players outside the enemy's tracking range will not be attacked.

# 8 Game Art

All materials in the game are 2D pixel style. Here are some examples:



Figure 10 Example of Characters



Figure 11 Example of Weapons



Figure 12 Example of Enemies

# 9 Management

## 9.1 Schedule

### 9.1.1 Prototype 1 – October and November, 2022

* Design and decide game concepts
* Complete main design of the game

### 9.1.2 Prototype 2- December, 2022

**Map System:**

* Custom dungeon room editor
* Room templates
* Dungeon room builder
* Game scene setup
* Doors to the rooms

**Character System:**

* Character basic animation setup
* Character movement and control

### 9.1.3 Prototype 3- January, 2022

**Map System:**

* Mini Map

**Weapon System:**

* Object pooling
* 9 types of weapons
* Special Effects of weapons
* Sound Effects of weapons

### 9.1.4 Prototype 4- February, 2022

**Enemy System:**

* Enemy basic animation setup
* Enemy movement AI – A\* algorithm
* Enemy spawner
* Enemy weapons
* Enemy special ammo patterns

**Battle System:**

* Health and damage
* Battle between levels
* Boss room setup

**Map system:**

* Decorative objects
* Moveable objects
* Torches
* Chests and supplies
* Overview Map
* Teleporting

### 9.1.5 Prototype 5 – March, 2022

**Map System:**

* Room templates art assets replacement

**Scene System:**

* Main menu
* Pause menu
* Instructions menu
* High score menu

### 9.1.6 Prototype 6 – April, 2022

* Numerical value balance
* More interesting weapons
* Improving game art assets
* Test and fix bugs

## 9.2 Version Control

The link to GitHub repository: [GitHub - luoyangyuli2001/InfiniteDungeoner](https://github.com/luoyangyuli2001/InfiniteDungeoner)

## 9.3 Test

### 9.3.1 Test Schedule

I will use my laptop to test my game. If possible, I will upload my game to the unity community, some players may come to play my game, and they may find some bugs in the game. At the same time, I will also share the game with my friends, let them help me test the game.

### 9.3.2 Bug Log

**Enemy spawn error:** In some room templates, tables blocked enemy spawn points, which caused enemies to get stuck in unintended places and crashed the game. (Fixed, Adjusted the position of the table to avoid overlapping with the enemy's spawn point.)

**Players can fire bullets outside the map:** If the player holds a weapon that is too long, in some rooms, the weapon can be extended to the outside of the map to fire. (Fixed, added collision detection to the weapons. Weapons now have entities instead of textures, but this will cause the interaction distance with the chests to be too long when holding a large size weapon, requiring the player to open the chest in a specific pose.)

**Too much ammo supplies:**  Chests generate too much ammo supplies. Upon inspection, it was found that the program recognized the weapon with infinite ammo as a lack of ammo, and thus triggered the dynamic difficulty compensation mechanism. (Fixed, when checking the player's ammo status, it now skips the judgment for weapons with infinite ammo.)

**Wrong map generation:** In the third level, there is a wrong map generation around a treasure chest room. After investigation, it was found that the bug was caused by the second map logic model, and the cause of the bug has not been found yet. But after redoing the logical model of the map, the bug has been fixed. **(**The bug was discovered by user 1.**)**



Figure 13 Chest room with bug



Figure 14 Chaotic corridor around chest room

### 9.3.3 Tests and Users Feedback

I shared my game link with my friends, they will play and experience the whole game process, and try to help me find bugs in the game, and comment on the game itself.

#### 9.3.3.1 User 1 Test Result and Feedback

**Test result:** The third level found a wrongly generated room.

**Feedback:** The game has a high degree of completion, but there is still a big gap with the same type of games currently on the market. These are areas where I feel the game needs improvement:

* The guidance of the game is weak, although the instructions clearly explain the various operations in the game and the problems that need attention. But considering that the vast majority of players ignore instructions when playing games, not adding enough guidance to the game will be an easy point to get a bad review.
* The function of overturning the table to avoid bullets is very unfriendly for new players. Interacting with the table requires a certain level of proficiency
* Weapons have collision detection which feels odd and tends to get stuck in tight places when using weapons with larger models.

The game also has some highlights:

* There are many types of firearms and they are all very cool, and the sound effects are in place.
* The design of the teleporting is very good, eliminating the tedious process of running through many rooms.

#### 9.3.3.2 User 2 Test Result and Feedback

**Test result:** No bugs

Feedback: The invulnerability time generated by enemies is too long, which makes the battle not very refreshing. The guidance after clearing the Boss room is not very good. I didn’t notice how to enter the next level when I played for the first time. The weapon is too long and gets stuck in the corridor, but it's kind of fun.

#### 9.3.3.3 User 3 Test Result and Feedback

**Test result:** No bugs

Feedback: The game is highly completed, especially when there is only one developer and time is limited. Cheating characters is a very good setting. The light effects and music in the game are very good

#### 9.3.3.4 User 4 Test Result and Feedback

**Test result:** No bugs

Feedback: I encountered a lot of strange problems when I played it for the first time, but after reading the instructions, there were no problems at all. This game contains all the content it should have, the only shortcoming is that there is no sound effect for killing monsters, and it lacks the sense of attack. It is enough for one person to do this. When it comes to cheating characters, the exhilaration of not having to reload ammo is unparalleled.

# 10 Reflection

## 10.1 What went right and what went wrong

If I were asked to describe the most correct thing I did during the entire development process of my Final Year Project, I would say that it was setting my project as a game. Although I had previously attempted to create some simple games during my university studies and submitted them as assignments for some courses, and had achieved high scores, dedicating nearly half a year of time and energy to the development of a larger game posed a challenge for me and required me to learn a lot of new knowledge that the simple games I made before could not match.

Before this, I always looked at every game I played from a player's perspective or as a video game enthusiast, and although I studied computer-related courses, I sometimes speculated about the logic behind the games. However, when I truly put myself in the position of a game developer, I realized that there were many things in games that were not as simple as I had imagined. After completing my Final Year Project, when I looked back at the games I had played before, I finally understood the hard work that game developers put into even the most insignificant details of a game, such as creating materials and models for small items in a scene, just like they would do for larger components.

During the development process, I referred to many game development tutorials, but encountering problems was inevitable. However, the majority of the problems were solved, and some of them were easily solved through components provided by the Unity engine. For the remaining issues, I was able to find preliminary solutions through Stack Overflow or by consulting with ChatGPT.

## 10.2 What I learned from the process

During the entire development process, I learned a lot. Firstly, Unity engine, as a famous game engine, provides many convenient components such as the Light component for making light sources, the Rigidbody component for physical calculations, the Collider component for collision detection, and the Particle Effect component for making special effects. Learning how to use these components allowed me to further understand the implementation details of various game functionalities.

In addition to that, developing a relatively large game independently also allowed me to experience the difficulties of managing a large-scale project. I was able to practice the theoretical knowledge frequently mentioned in software engineering courses, such as keeping the code loosely coupled, scalable, and readable. These are all important factors for maintaining a high-quality codebase and a successful project.

Moreover, the project also taught me about the importance of teamwork and communication. While I was the sole developer of the project, I needed to communicate effectively with my supervisor and peers to receive feedback and suggestions. This helped me to improve my project and also prepared me for working on collaborative projects in the future.

Overall, developing my Final Year Project as a game was a great decision as it allowed me to gain valuable practical knowledge and experiences that will be beneficial for my future career as a game developer or software engineer.

## 10.3 What I would do differently if I had to do it all over again

If I had to start over again and had enough time, I would devote a lot of effort to creating game assets. Throughout the development process, I have been constrained by the lack of game assets. At the beginning, I tried to introduce a rich storyline, like the game "The Binding of Isaac," which has both satisfying gameplay and storyline. However, I soon realized that games that can create excellent storylines not only have well-crafted storylines, but also rich game assets to support the characterization of the storyline, just like a book and its movie adaptation. Nowadays, games have surpassed technical limitations, and there are countless games with excellent graphic design. Without sufficient and appropriate game assets, even with an excellent script, most players today are not interested.

With sufficient game assets, perhaps I can implement a twist-filled storyline like the “Ender's Game” mentioned in the initial design plan. At the same time, I can also add more features to the game, such as introducing puzzle elements. If I had enough time to create quality game assets, I would be able to build a more engaging and immersive game experience.

## 10.4 What I would do if I had more time to develop this project further

If I had more time to develop this game, the first thing I would do is to expand its content by adding more monsters, levels, and weapons. A good rogue-like game should have a vast amount of game content to support its high level of randomness. Besides expanding the game content, I believe further enhancing the game's graphics and visual elements could be a good idea.

However, these additions would only be incremental improvements to the existing game. If possible, I would still like to reconsider the entire game design, as I mentioned in the previous section.

## 10.5 What were the most challenging features to implement

The most challenging feature to implement, in my opinion, was the enemy AI movement in the game. First, I attempted to implement the A\* pathfinding algorithm, which took me a lot of time since I had not previously delved deep into various algorithms. I spent many days learning how to apply the A\* pathfinding algorithm in the game. Then, when I tried to combine the implemented A\* pathfinding algorithm with my game code, I encountered more problems, such as how to dynamically update the path used for enemy movement when there are movable obstacles in the game that can be moved by the player. In addition to this, the initial AI controlling enemy movement consumed a lot of CPU time, especially when the player was far from the enemy, and there were many enemies on the screen, which caused noticeable game lag. Therefore, I spent a lot of time researching how to optimize the algorithm. Although the final optimization measure I adopted was simple, it was effective - I just added an interval for path building.

If I had more time, I would have liked to explore other AI algorithms that could further enhance the enemy's movements in the game. There are many potential areas for improvement, such as implementing more complex pathfinding algorithms or developing better methods for handling obstacles. I would also consider ways to optimize the code to ensure that the game runs more smoothly even when there are many enemies on the screen. By improving the enemy AI, I can create a more immersive and challenging gameplay experience for the player, which would enhance the overall quality of the game.

# 11 Appendices

## 11.1 Material Source

The materials used in the game come from the following sources

* [Unity 2D Dungeon Gunner Roguelike Development Course | Udemy](https://www.udemy.com/course/unity-2d-dungeon-gunner-roguelike-development-course/learn/lecture/30926182#announcements)
* [PC / Computer - Enter the Gungeon - The Spriters Resource (spriters-resource.com)](https://www.spriters-resource.com/pc_computer/enterthegungeon/)
* [Characters | Soul Knight Wiki | Fandom](https://soul-knight.fandom.com/wiki/Characters)
* [Nintendo Switch - Enter the Gungeon - The Sounds Resource (sounds-resource.com)](https://www.sounds-resource.com/nintendo_switch/enterthegungeon/?source=genre)

This game is for educational purposes only and not for any commercial use.

## **11.2 References**

Botto. (2022, December 21). *The Binding of Isaac (video game) - Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/The\_Binding\_of\_Isaac\_(video\_game)

Guerillero. (2022, October 21). *Unity (game engine) - Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Unity\_(game\_engine)

MrOllie. (2022, December 29). *Scrum (software development) - Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Scrum\_(software\_development)

Rkieferbaum. (2023, March 28). *Enter the Gungeon - Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Enter\_the\_Gungeon

Unity. (2020, January). *Introduction to Lights 2D | Universal RP | 7.1.8*. Retrieved from Unity Manual:https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@7.1/manual/Lights-2D-intro.html

Unity. (2020, March). *Unity - Manual: Rigidbody 2D*. Retrieved from Unity Manual: https://docs.unity.cn/2020.3/Documentation/Manual/class-Rigidbody2D.html

Unity. (2021, March). *Unity - Manual: Audio Listener*. Retrieved from Unity Manual: https://docs.unity3d.com/Manual/class-AudioListener.html

Unity. (2021, March). *Unity - Manual: Box Collider 2D*. Retrieved from Unity Manual: https://docs.unity3d.com/Manual/class-BoxCollider2D.html

Unity. (2021, March). *Unity - Manual: Polygon Collider 2D reference*. Retrieved from Untiy Manual: https://docs.unity3d.com/Manual/class-PolygonCollider2D.html

Unity. (2021, March). *Unity - Manual: Universal RP*. Retrieved from Unity Manual: https://docs.unity3d.com/Manual/com.unity.render-pipelines.universal.html

Unity. (2022, January). *Unity - Manual: Cinemachine*. Retrieved from Unity Manual: https://docs.unity.cn/2022.1/Documentation/Manual/com.unity.cinemachine.html

Vozul. (2023, February 27). *A\* search algorithm*. Retrieved from Wikipedia: https://en.wikipedia.org/w/index.php?title=A\*\_search\_algorithm&action=history

XoLo921. (2023, March 8). *Video game graphics - Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Video\_game\_graphics#Top-down\_perspective

## 11.3 Bibliography

### 11.3.1 Udemy Courses

* [Unity 2D Random Dungeon Generator for a Roguelike Video Game | Udemy](https://www.udemy.com/course/unity-2d-random-dungeon-generator-for-a-roguelike-video-game/)
* [Learn To Create A Roguelike Game In Unity | Udemy](https://www.udemy.com/course/unityroguelike/)

### 11.3.2 YouTube Videos

* [2D Top Down Pixel Art RPG Game Dev in Unity 2022 ~ Crash Course Tutorial for Beginners - YouTube](https://www.youtube.com/watch?v=7iYWpzL9GkM)
* [Unity Procedural Generation of a 2D Dungeon - Introduction - YouTube](https://www.youtube.com/watch?v=-QOCX6SVFsk&list=PLcRSafycjWFenI87z7uZHFv6cUG2Tzu9v)
* [How to make an EDITOR WINDOW in Unity - YouTube](https://www.youtube.com/watch?v=491TSNwXTIg)

### 11.3.3 Bilibili Videos (A Chinese video platform)

* How to use unity to make a top-down shooting game: https://www.bilibili.com/video/ BV1xb4y1D7PZ
* How to implement weapons in Unity https://www.bilibili.com/video/BV1Mh411v7PU

## 11.4 Project Video Log

Semester 1 Demo: <https://youtu.be/juH6IerigGE>

Semester 2 Demo: [FYP-InfiniteDungeoner - YouTube](https://www.youtube.com/watch?v=V_uWVoMy70g)

## 11.5 User Manual

* Use the 'WASD' keys to move and the mouse to aim.
* Left mouse button to fire (hold down fire to charge some weapons like the laser before firing).
* Hold down the right mouse button while moving for an evasive roll. You won't take damage while rolling.
* The 'Tab' key shows the dungeon overview map. Click the rooms you've visited to quickly teleport.
* Press the 'R' key to reload the weapon or hold down the left mouse button when the weapon clip is empty.
* Press the 'E' key to use items. Tables can be flipped to provide cover, chests can be opened, and items can be taken.
* Chest items: Hearts restore your health and ammo tops up the currently selected weapon. (Chests can be empty).
* Use the middle mouse wheel to scroll through your weapons. Weapons can also be selected using the number keys.
* Press the '-' key to move the currently selected weapon to position 1.
* Press the 'Esc' key for the pause menu.
* Clear all the rooms so that you can enter the boss room.
* After defeating the boss, press the 'Enter' key to enter the next level.
* If you kill an enemy, your multiplier increases by 1, and your score will add (enemy HP \* multiplier).
* If you get hit by an enemy, your multiplier will be reset to 'x1'.Good luck and have fun!

## 11.6 Game Online Deployment Link

I deployed the game on my friend's server, you can access it through the following link: [Unity WebGL Player | InfiniteDungeoner (iocky.com)](https://infinitedungeoner.iocky.com/)

## 11.7 Version Control

The link to GitHub repository: [GitHub - luoyangyuli2001/InfiniteDungeoner](https://github.com/luoyangyuli2001/InfiniteDungeoner)

## 11.8 Balance of the value in the game

During the game development process, I created the following table for enemy health and the strength of various weapons that the player can possess. This table will also be used for balancing the game's values.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Enemy Name | LV0 | LV1 | LV2 | LV3 | LV4 | LV5 |
| Grimonk | X | X | 50 | 60 | 60 | 70 |
| Hedusa-B | X | X | 35 | 40 | 40 | 50 |
| Hedusa-G | 15 | 25 | 35 | 35 | 35 | 35 |
| Hedusa-R | X | 25 | 35 | 40 | 40 | 50 |
| MudRock | X | X | 50 | 60 | 60 | 70 |
| Orc | X | X | 50 | 60 | 70 | 70 |
| Slime-B | X | 25 | 35 | 35 | 35 | 40 |
| Slime-G | 20 | 25 | 35 | 35 | 35 | 40 |
| Slime-R | X | X | 35 | 35 | 35 | 40 |
| Slizzard | X | X | 50 | 60 | 60 | 70 |
| Boss | 300 | 400 | 400 | 400 | 400 | 400 |

Table: Enemies’ HP in each level

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Weapon Name | Damage | Ammo Number | Fire Rate | Level | DPS |
| Pistol | 6 | 1 | 0.25 | 0 | 24 |
| Plasma | 8 | 1 | 0.4 | 0 | 20 |
| Revolver | 10 | 1 | 0.3 | 1 | 33.33333333 |
| DungeonEagle | 15 | 1 | 0.3 | 1 | 50 |
| Excalibur | 8 | 1 | 0.2 | 1 | 40 |
| FaceMelter | 3 | 23 | 1 | 1 | 69 |
| FlashRay | 15 | 1 | 0.35 | 1 | 42.85714286 |
| JK47 | 12 | 1 | 0.3 | 1 | 40 |
| MailBox | 12 | 1 | 0.3 | 1 | 40 |
| PeaShooter | 12 | 1 | 0.3 | 1 | 40 |
| RustySidearm | 11 | 1 | 0.3 | 1 | 36.66666667 |
| SawedOff | 7 | 7 | 0.4 | 1 | 122.5 |
| Shotgun | 4 | 7 | 0.4 | 1 | 70 |
| N22Laser | 25 | 1 | 0.5 | 2 | 50 |
| RPG | 200 | 1 | 2 | 2 | 100 |
| SniperRifile | 35 | 1 | 0.85 | 2 | 41.17647059 |
| AK-47 | 14 | 1 | 0.25 | 3 | 56 |
| BigShotgun | 6 | 15 | 0.4 | 3 | 225 |
| QBZ | 10 | 1 | 0.2 | 3 | 50 |
| Railgun | 35 | 1 | 0.4 | 3 | 87.5 |
| GM6Sniper | 60 | 1 | 0.7 | 4 | 85.71428571 |
| MP7 | 10 | 1 | 0.1 | 4 | 100 |
| AKEY-47 | 20 | 1 | 0.15 | 5 | 133.3333333 |
| AWP | 200 | 1 | 1 | 5 | 200 |
| Com4nd0 | 250 | 4 | 2 | 5 | 500 |
| MassShotGun | 10 | 25 | 0.5 | 5 | 500 |
| Rocketlauncher | 500 | 1 | 2 | 5 | 250 |

Table: Weapons’ Attributes