# ***Game design report***

# ***Game name: Seek to survive***

# ***Long Liu 20104729***

# ***Software system practice Year 4***

# ***Date:24/4/2024***

contents

[1. Game Overview 1](#_Toc20268)

[1.1 Game Concept 1](#_Toc16435)

[1.2 Genre 1](#_Toc27561)

[1.3 Target Audience 2](#_Toc5415)

[1.4 Game Flow Summary 2](#_Toc31337)

[1.5 Look and Feel 2](#_Toc19153)

[1.6 Project Scope 3](#_Toc8382)

[1.6.1 Number of Levels 3](#_Toc12919)

[1.6.2. Number of Location 3](#_Toc18430)

[1.6.3. Number of Enemies 3](#_Toc22470)

[1.6.4. Number of Buildings 4](#_Toc26122)

[1.6.5. Description of Player 4](#_Toc13912)

[2. Gameplay and Mechanics 5](#_Toc30797)

[2.1. Gameplay 5](#_Toc27970)

[2.1.1. Game Progression 5](#_Toc9106)

[2.1.2. challenge Structure 5](#_Toc12294)

[2.1.3. Objectives 5](#_Toc28592)

[2.1.4. Play Flow 6](#_Toc25999)

[2.2. Mechanics 6](#_Toc3714)

[2.2.1. Physics 6](#_Toc18574)

[2.2.2. Movement 7](#_Toc15201)

[2.2.3. Objects 8](#_Toc4052)

[2.2.4. Actions 8](#_Toc1721)

[2.2.5. Combat 9](#_Toc7928)

[2.2.6. Economy 10](#_Toc13607)

[2.3. Screen Flow 11](#_Toc8453)

[2.3.1. Screen Flow Chart 11](#_Toc377)

[2.3.2. Screen Descriptions 12](#_Toc553)

[2.4. Replaying and Saving 16](#_Toc12216)

[3. Story, Setting and Character 16](#_Toc7188)

[3.1 Story 16](#_Toc20483)

[3.2 Character and Animation 17](#_Toc14838)

[4.Interface 19](#_Toc23188)

[4.1. Visual System 19](#_Toc18521)

[4.1.1. HUD (Heads up display) 19](#_Toc28832)

[4.1.4. Camera 20](#_Toc3226)

[4.2. Control System 20](#_Toc358)

[4.3. Audio 21](#_Toc23650)

[5. Technical 21](#_Toc6028)

[5.1. Target Hardware 21](#_Toc3690)

[5.2. Development hardware and software 21](#_Toc25388)

[5.2.1 Hardware 21](#_Toc10437)

[5.2.2 Software 22](#_Toc31888)

[5.3. Game Engine 22](#_Toc734)

[5.4. Scripting Language 23](#_Toc23698)

[6. Project Management 23](#_Toc15495)

[6.1. Project Methodology 23](#_Toc30529)

[6.3. Version control 24](#_Toc2002)

[6.4. Risk Analysis 24](#_Toc27096)

[7. Appendices 25](#_Toc18120)

[7.1 References 25](#_Toc11361)

# 1. Game Overview

## 1.1 Game Concept

This game is a side-scrolling video game. The player can move left or right. Players have a base to upgrade abilities. The enemy will appear at some fixed locations,the nest, and attack the player. As time goes on, the player can collect materials to make tools and update buildings. When the game enters some specific days, some enemies will actively attack the player's base. The player needs to guard the base to avoid failure of the game. Player can gain materials by killing enemies. Materials can be used to upgrade base and player. On the base, player can build defense buildings.

## 1.2 Genre

* **side-scroller**: A side-scrolling video game (alternatively side-scroller) is a [game](https://en.wikipedia.org/wiki/Game" \o "Game) viewed from a side-view camera angle where the screen follows the [player](https://en.wikipedia.org/wiki/Player_character" \o "Player character) as they move left or right. [1]
* **[Tower Defense](https://store.steampowered.com/tags/en/Tower%20Defense/?snr=1_5_9__410)**: **Tower defense** (**TD**) is a subgenre of strategy games where the goal is to defend a player's territories or possessions by obstructing the enemy attackers or by stopping enemies from reaching the exits, usually achieved by placing defensive structures on or along their path of attack.[2]
* **Base Building**: The player will build base to improve their strength, such as more buildings and people.
* **Single player**: The game has only one player.

## 1.3 Target Audience

My target audience will be people who have interests in Tower defense games. It mainly refers to players who can accept pixel style, combat with enemies and build base.

## 1.4 Game Flow Summary

The game will load the main menu menu, where the player can select new game,continue game, options and quit game. In the game, player needs to upgrade base to the highest level, and hold the base during an attack triggered by the upgrade.

## 1.5 Look and Feel

The aim of the project was to develop a pixel-style tower defense construction game with a side view, similar to the popular side-scrolling game Terraria. The player's main task is to build defensive structures on the map to defend against enemy attacks and destroy enemy nest.

Figure1.1 Game screen

## 1.6 Project Scope

### 1.6.1 Number of Levels

There is only one difficulty level.

### 1.6.2. Number of Location

Players and enemies will only belong to one map. The player will appear on one end of the map and the enemy's lair will appear on the other. Some broken buildings or hidden buildings will appear in the space between the base and the nest. Players can repair these broken buildings to get them working again. These buildings can support some additional functions.

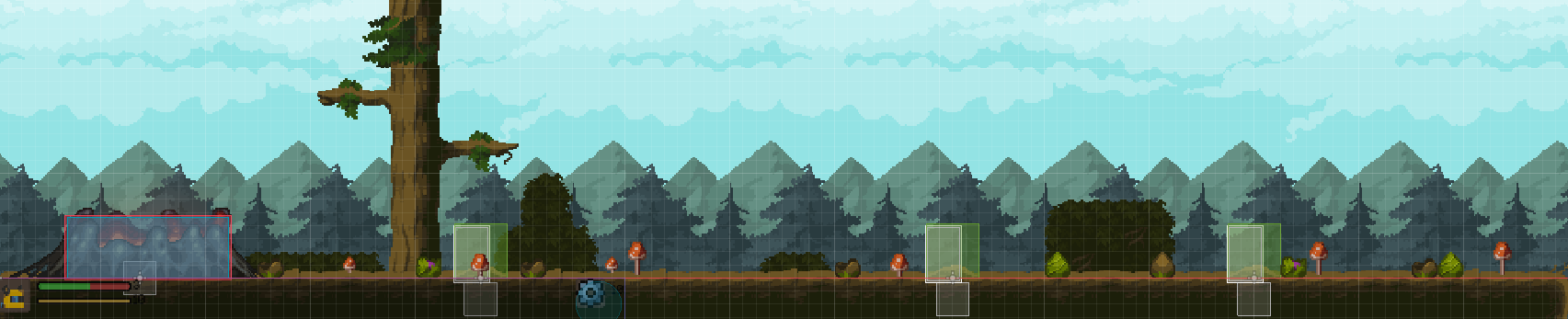


Fig1.2 map

### 1.6.3. Number of Enemies

There will be one kind of enemy.

* Attacker: attack from frontal, medium heath, medium speed ,medium power

All enemies come from nest.

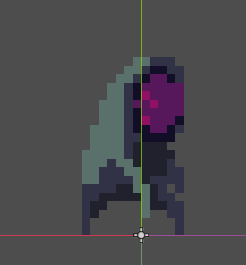


Fig1.3 attacker

### 1.6.4. Number of Buildings

Base location: nest

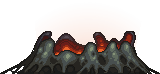


Fig1.4 nest

Annexe types:

* wall building points

wall_1

Fig1.5 wall

#### 1.6.4.1 Details of Buildings

* **Wall**: defense attack, has 3 levels
* **Nest**: enemies will appear there, has more health points.

### 1.6.5. Description of Player

* Range attack,
* has a bag to save materials,
* medium attack power,
* high health,
* medium speed,
* use power increase speed in a short time,
* If player dies, the game will end.

# Gameplay and Mechanics

## 2.1. Gameplay

### 2.1.1. Game Progression

In the beginning, the player will show up at the base with some initial materials. These materials can be used to build in buildings. At the same time, the lair periodically generates enemies to attack the base. Players can obtain materials by destroying enemies. As the game progresses, the player will be able to build more buildings, and the nest will generate more and more enemies. The player's goal is to destroy the nest before being taken down by the enemy.

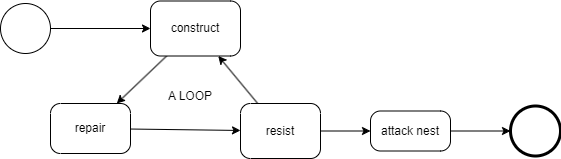
### 2.1.2. challenge Structure

Every once in a while, these lair will produce some enemies to attack the base. The player must destroy all enemies. When no enemies are created, the player can use this time to explore the map.

2.1.3. Objectives

The player should protect themselves , and aim to destroy the nests.

### 2.1.4. Play Flow

****

**Fig4.1 play flow**

Players can use materials to build walls. Over time, enemies will attack the player, and the player can use the walls to defend against these attacks, eventually eliminating the enemy. After the battle, players can repair damaged buildings with materials. Through a cycle of development, defense, and repair, players will continually cut the nest's life. The player can attack the lair. If the player's health reaches zero, the game ends in failure. If the nest is destroyed, the player wins.

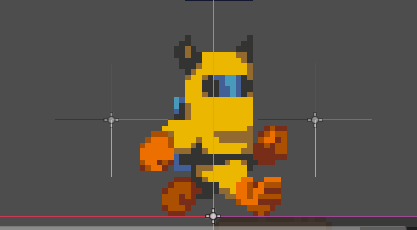
## 2.2. Mechanics

### 2.2.1. Physics

The physics of this 2d game is not real physics, but a simulation based on collision detection. There are four main types. Area2D can detect when objects overlap and signal when an object enters or leaves. StaticBody2D is the body of a physical engine that does not move. It is involved in collision detection, but does not move in response to a collision. RigidBody2D simulates 2D physics. It cannot be controlled directly, but rather by applying a force to it and then calculating the resulting motion by the physics engine. Every object that requires a physical effect has a collision shape to determine the size of the collider. Each object can specify layers to determine which layers it will physically collide with.

### 2.2.2. Movement

#### 2.2.2.1. General Movement



**Fig2.1 movement**

Players use the 'A' and 'D' keys on the keyboard to move and the left mouse button to control attacks. The perspective is always centered around the character controlled by the player. Players move at a fixed speed.

Pressing 'Shift' allows players to expend energy for quick movements for a short period. After running out of stamina, the player will stop their fast movement and move at a slower speed than usual. Stamina will slowly recover. If the player stops moving, their stamina will recover more quickly.

On Android devices, players move through virtual joysticks. And interact with buttons on the screen.

#### 2.2.2.2. Other Movement

* **Enemies**: They attack any buildings and player.

### 2.2.3. Objects

#### 2.2.3.1. Picking Up Objects

Sometimes enemies will drop some materials, players can get them.

### 2.2.4. Actions

#### 2.2.4.1. Switches and Buttons

When the game is paused, the player can change the game’s settings and save the game and back ti title screen.

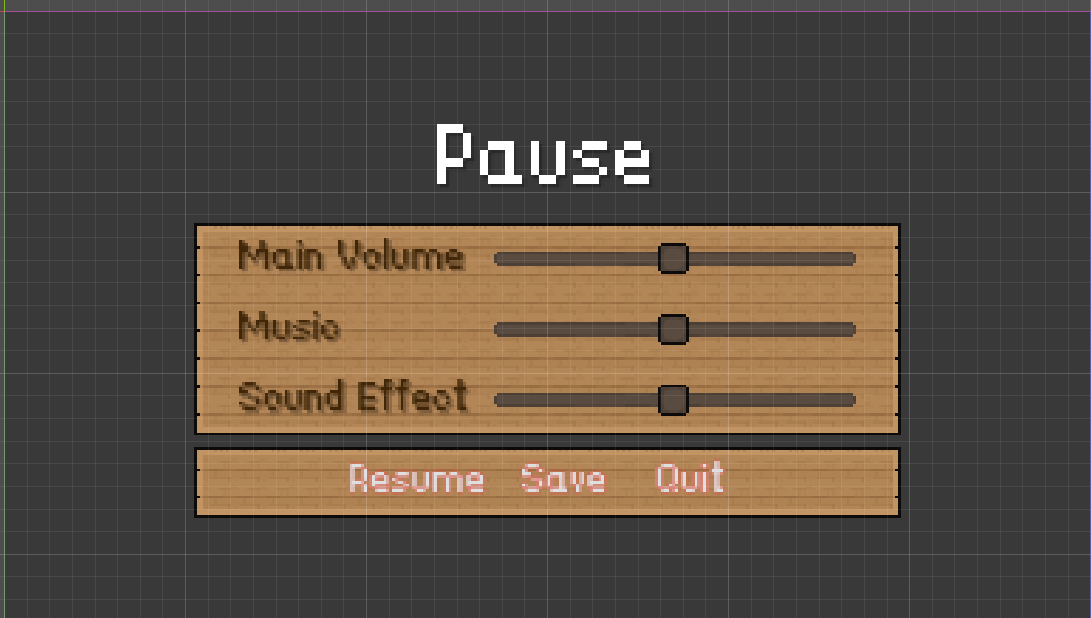


Fig2.1

#### 2.2.4.2. Aiming and Shooting



**Fig4.3 shooting**

The player does not need aim to shoot. Bullets will be shot in the direction facing the player. The Normal Attack is auto aim.

### 2.2.5. Combat



Fig2.2.1

**Bullet**:

Players and some enemies can shoot, usually the enemy's bullets are green and the player's bullets are orange. This allows players to distinguish which bullets belong to themselves and which belong to enemies. Our bullets can pass walls and shields to attack enemies.

**Barrier**:

Shield and wall can resist bullets. But if they are hit multiple times, they will be destroyed. The shield needs time to be repaired. If the wall is completely destroyed, it will require the expenditure of materials for reconstruction. Otherwise, only the worker needs to fix it. The wall can be upgraded, and the upgraded wall can withstand multiple attacks.

**Enemy**:

**Attacker**:

The enemy will approach and attack.

### 2.2.6. Economy

The player can get materials from enemies. The player can spend materials to build buildings , upgrade buildings , and repair and rebuild buildings.

## 2.3. Screen Flow

### 2.3.1. Screen Flow Chart

### screen flow.drawio

**Fig4.7 screen flow**

### 2.3.2. Screen Descriptions

#### 2.3.2.1. Main Menu Screen



**Fig4.8 main menu screen**

It will show the game’s title. And the screen will supply 3 options.

* **New**: Start a new game.
* **Load**: Continue a game from a game save.
* **Exit**: Kill the game process.

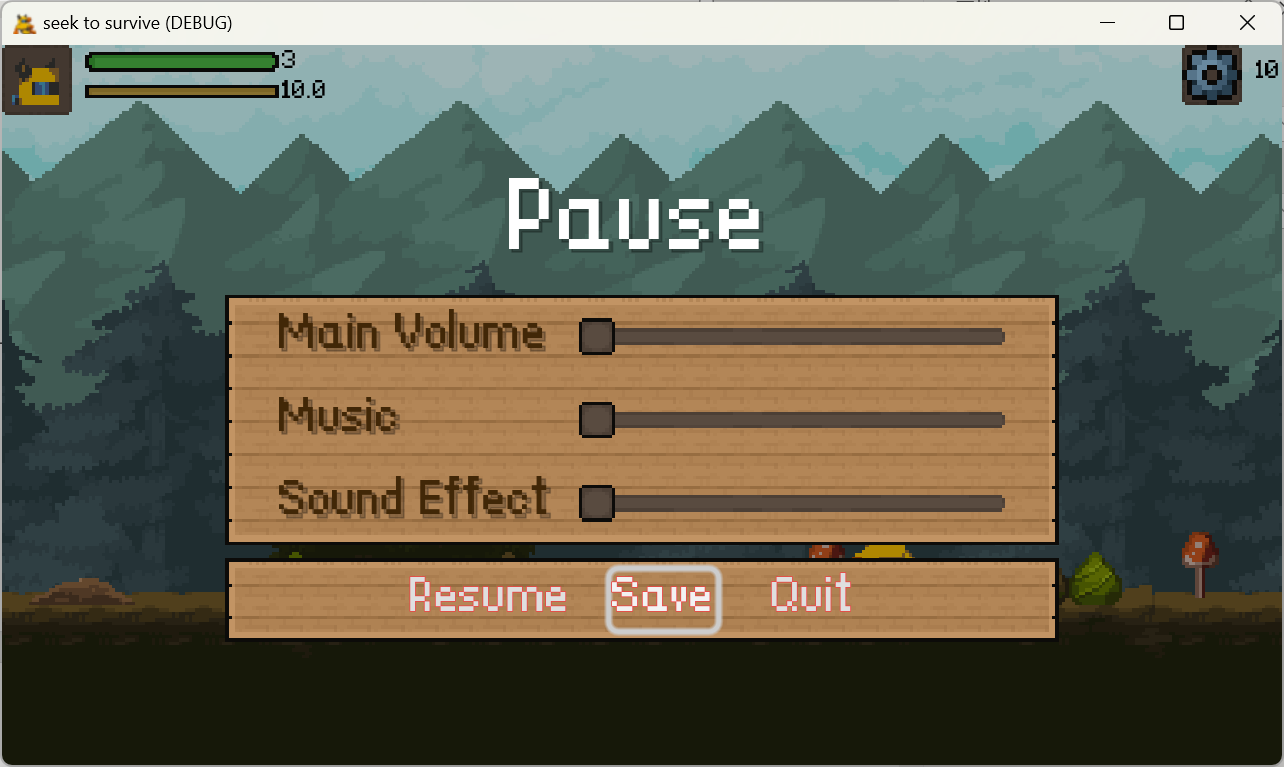
#### 2.3.2.3. Gameplay Screen



**Fig4.10 gameplay screen**

It will show the game’s content. The player can pause the game and open menu screen.

#### 2.3.2.4.Pause Menu Screen

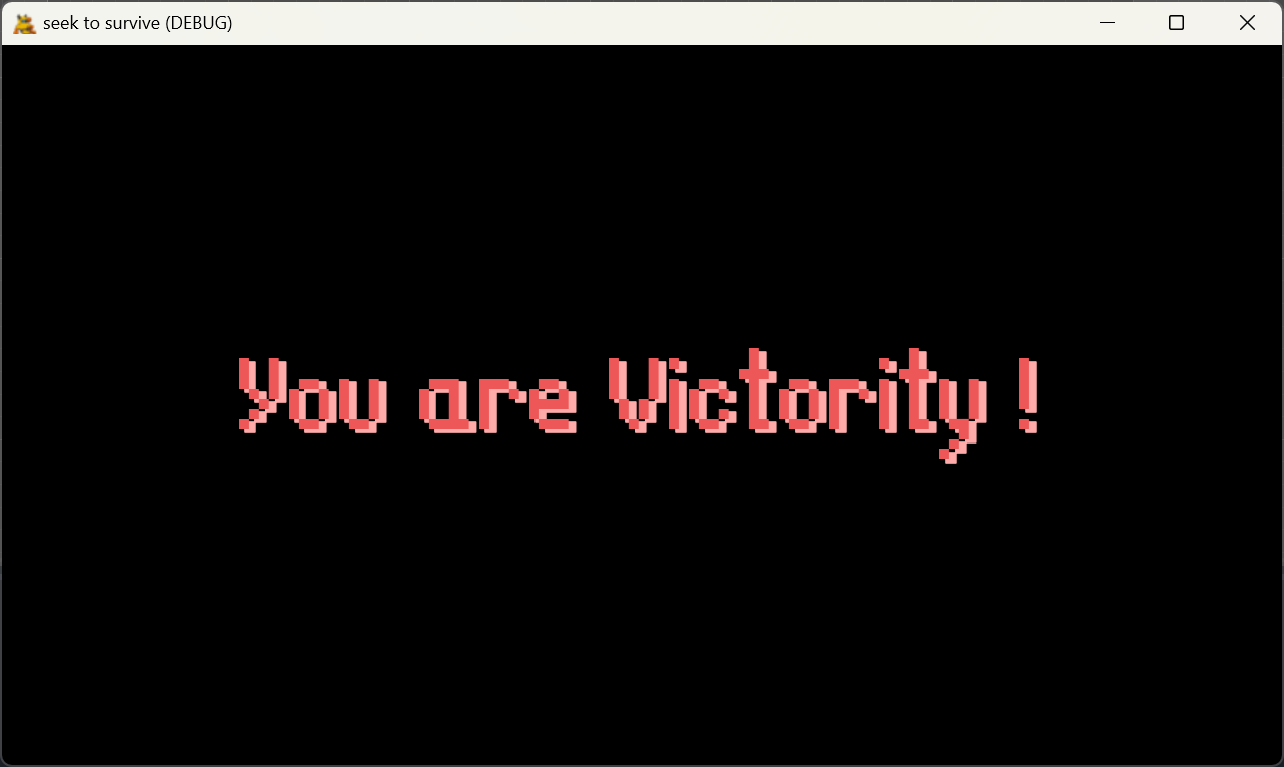


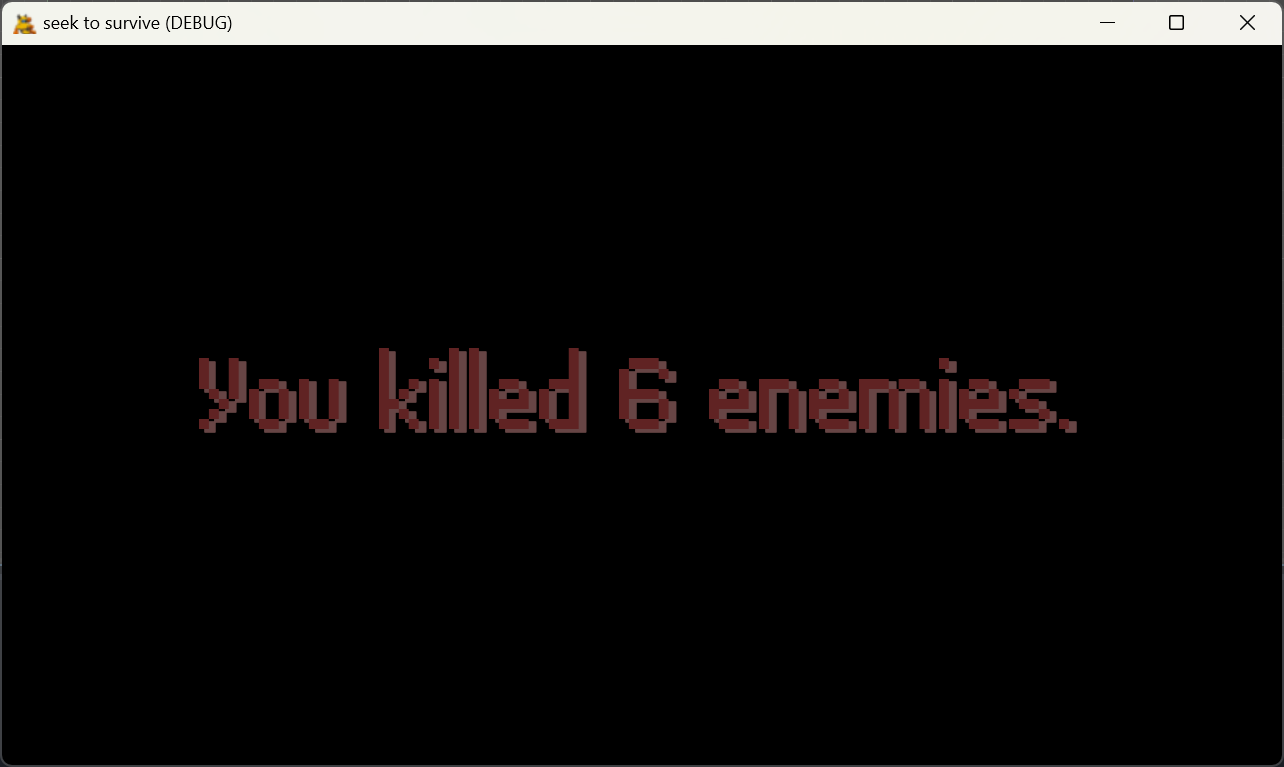
**Fig4.11 menu**

The game will be paused. The player can save the game on the screen. At the same time, the player can quit the game on the screen.

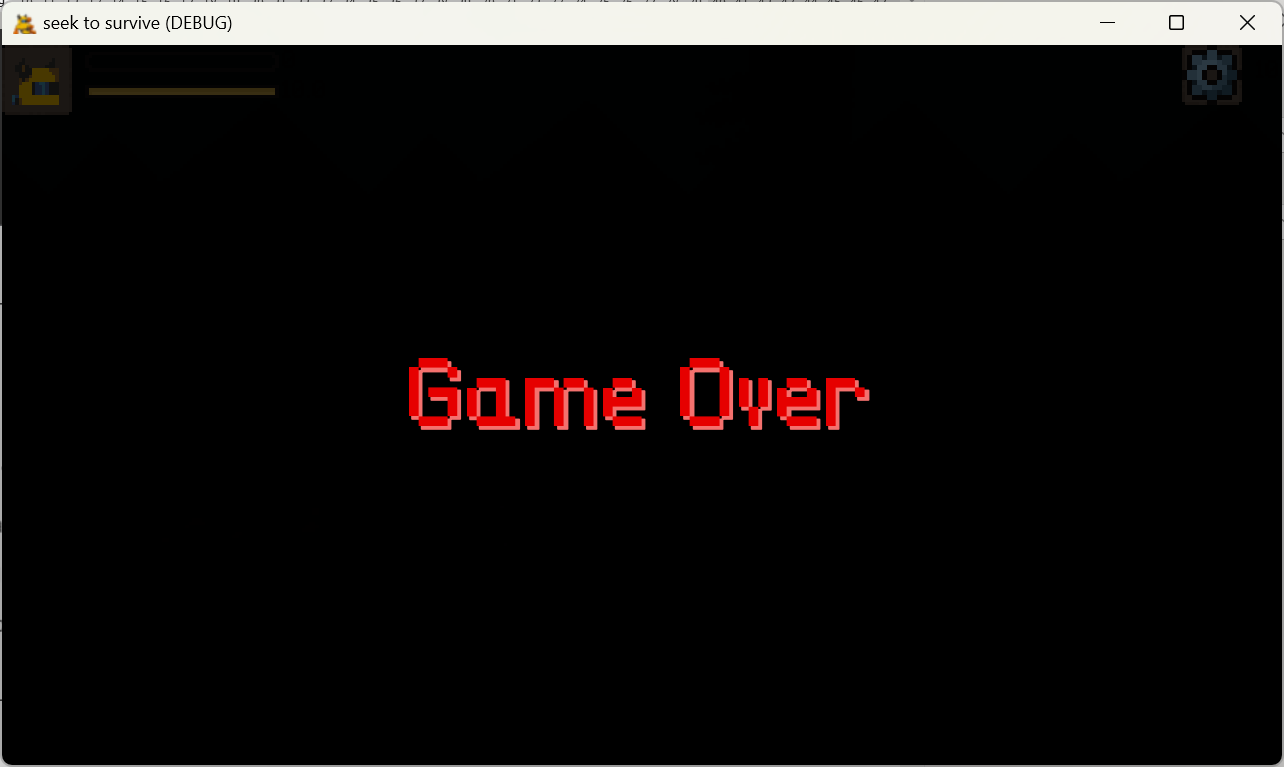
The player can change game’s settings, such as language, screen size, music volume and key settings.

#### 2.3.2.4. Game End Screen

****



**Fig4.12 game end of victory**



**Fig4.13 game end of failure**

The scree will show some statistical data about this game, such as number of enemies killed by the player. It will not take up too long time.Then, the game will jump to main menu screen.

## 2.4. Replaying and Saving

The player can save the game at any time. The game save will save the information of all characters, contains the player,bullets, enemies, and buildings. If the player continue game from a game save, the program will load all from the save file.

The game archive file format is json. The file contains the location and status of all nodes in the scene.

In addition, player changes to the game's Settings are saved in the same folder. The file format is ini. The user's folder location is %AppData%\Roaming\Godot\app\_userdata\seek to survive.

# Story, Setting and Character

## 3.1 Story

The meteorite that falls from the sky brings a kind of unknown mutated creature, and the humans have to retreat to the underground tunnels. The player's base is the underground part of a subway station, and the player must hold their ground against the attack of mutated creatures.

## 3.2 Character and Animation

#### 3.2.1 Player

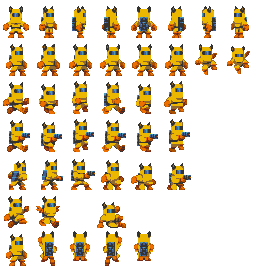


Fig3.2.1 player

This is the player's animation sequence frame, including standby, running, shooting, etc.

#### 3.2.2 Enemy

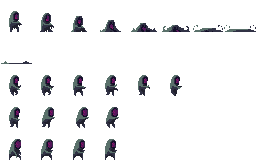


Fig3.2.2 attacker

This is the animated sequence frame of the enemy, including standby, running, death, etc.

#### 3.2.3 Nest

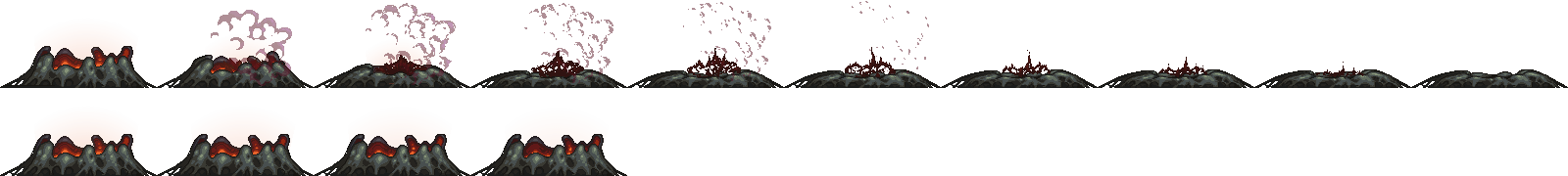
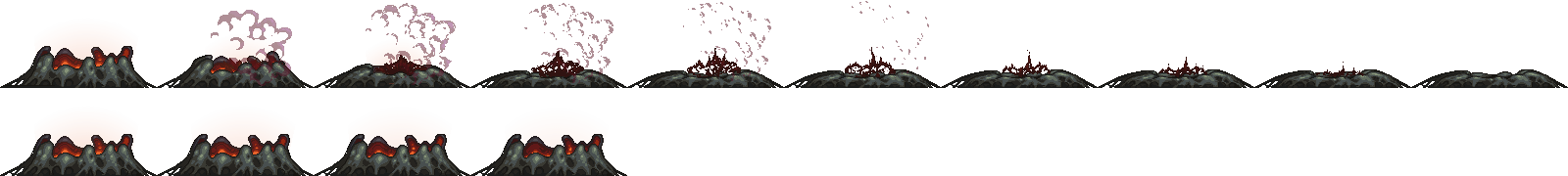


Fig3.2.3 nest

This is the animated sequence frame of the enemy, including standby, running, death, etc.\

#### 3.2.4 Transition animation

The game will have a fade in and fade out effect when switching scenes

#### 3.2.5 Title animation

On the title screen, the game title and buttons are animated with movement.

#### 3.2.6 Background movement

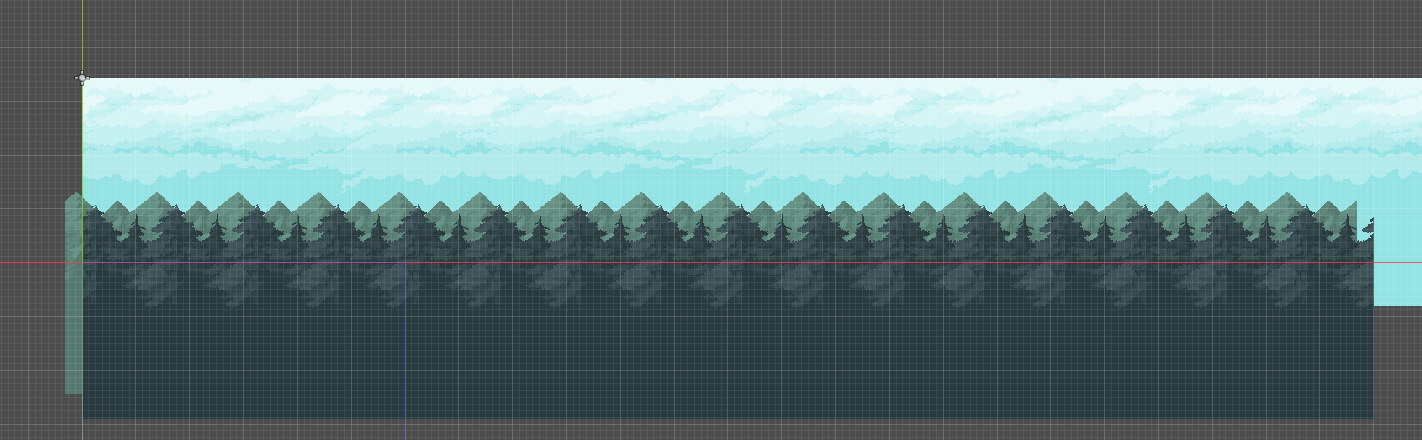


Fig3.2.6.1 background

The background of this game has three layers, made up of sky, mountains and forests. When the player's camera moves, the effect of stereo is simulated by the different relative movement speeds of the three layers. The sky layer moves the slowest, followed by the mountain layer and the forest layer the fastest. But they all move slower than the player's camera moves. And the image is stitched together before the camera moves to the edge of the image to achieve the effect of continuous background.

#### 3.2.7 Screen shake

When the player is injured, the game's screen vibrates to enhance the impact.

# 4.Interface

## 4.1. Visual System

### 4.1.1. HUD (Heads up display)



**Fig 5.1 HUD**

There are two long rectangles on the left top to show health points and stamina.

On the right top, the screen will show the specific number of materials and the button of menu.

### 4.1.4. Camera

The camera will focus on the player. And it moves with the player.

At the same time, the camera is restricted from going beyond the map.

## 4.2. Control System

The player use ‘A’ and ‘D’ to control the character’s movement. Clicking ’F’ to shoot bullets. The player using ‘E’ to interact with buildings. If the player want to open the menu, he should click the menu button or click ‘Esc’ button on keyboard.



Fig4.2.1

On touch devices, the game controls the character in a different way. The virtual controller controls the movement of the character, and E controls the interaction function. S controls the firing function. R Control Controls the player's sprinting ability.

## 4.3. Audio

The interactive focus of the game moves to and from buttons with sound effects. In the main interface of the game, the game scene, the game victory interface has different background music.

# 5. Technical

## 5.1. Target Hardware

MINIMUM:

|  |  |
| --- | --- |
| OS: | Windows 7 |
| Processor: | Intel 4th Gen Dual Core 2.0Ghz |
| Memory: | 2 GB RAM |
| Graphics: | Nvida GTX Series 8 |
| Storage: | 4 GB available space |
| Additional Notes: | The game can likely run on lower rated hardware, but I can't guarantee the performance or provide support. |

## 5.2. Development hardware and software

### 5.2.1 Hardware

|  |  |
| --- | --- |
| OS: | Windows 11(x64) |
| Processor: | 13th Gen Intel(R) Core (TM) i9-13980HX 2.20 GHz |
| Memory: | 32.0 GB |
| Graphics: | Nvida RTX 4090 laptop |
| Storage: | 1 TB available space |

### 5.2.2 Software

Godot v4.12 is the official version when I started developing the prototype.

Git,Asepirte.etc.

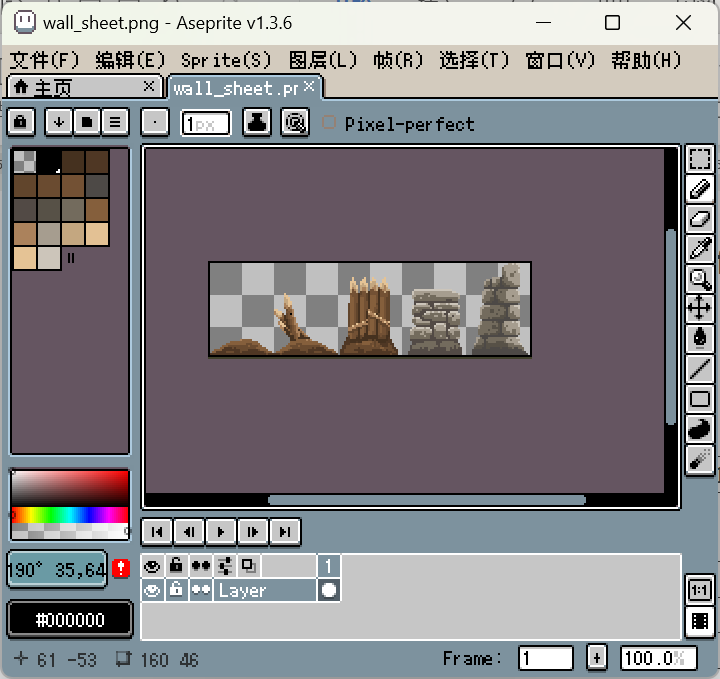


Fig5.2.1 Asepirte

## 5.3. Game Engine

**Godot** is a cross-platform, free and open-source game engine released under the permissive MIT license. It was initially developed by Argentine software developers Juan Linietsky and Ariel Manzur for several companies in Latin America prior to its public release in 2014. The development environment runs on many platforms, and can export to several more. It is designed to create both 2D and 3D games targeting PC, mobile, and web platforms and can also be used to develop non-game software, including editors.

Godot allows video game developers to create 3D and 2D games using multiple programming languages, such as C#, C++, GDscript.[3]

## 5.4. Scripting Language

The scripting language that will be used in this project is GDScripts. Because it is one of the main preferred languages for scripting in the Godot game engine.

# Project Management

## 6.1. Project Methodology

In this game development project, I chose to adopt an agile development method. Agile methodologies are known for their high adaptability to rapid change and uncertainty and are well suited to meet the ever-changing needs and challenges of game development.

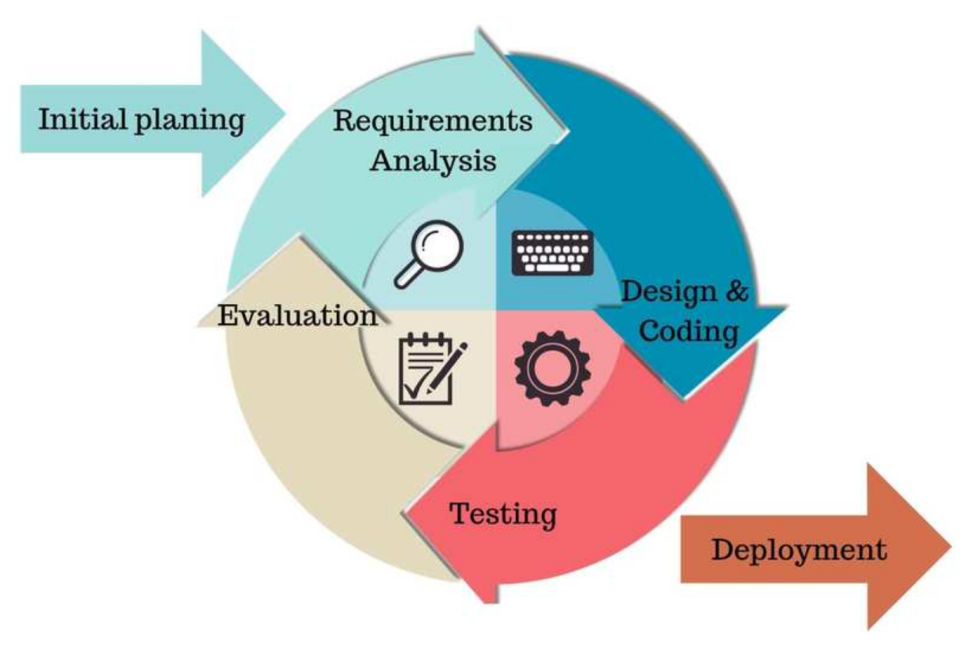


Fig 6.1 agile methodologies

## 6.3. Version control

GitHub is an online software source code hosting service platform that uses Git as a version control software. As of June 2022, GitHub has more than 57 million registered users and 190 million codebases (including at least 28 million open source codebases), making it in fact the largest code-hosting site and open source community in the world. As of January 26, 2023, GitHub is already used by over 100 million developers.[4] So I'll use Github to keep pushing code after every sprint.

However, once the project exceeds 100MB, Github will refuse to upload it. So I only uploaded the part that designed the key code, not the whole project.

## 6.4. Risk Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Risk Description | Risk Probability | Risk Impact | Strategy |
| Underestimating engineering difficulty | High | High |  |
| Contracting illness | Low | Low | Avoiding Sickness |
| Inability to meet deadlines | Medium | Medium |  |
| Lose of Report/Documentation | Low | High | Documents are stored on Baidu Cloud and regular copies are stored locally on several devices. |
| Lose of Project/Workspace | Med | High | Project are stored on Baidu cloud disk, and regular copies are stored on multiple local devices. |

# Appendices

## 7.1 References

1. [Side-scrolling video game - Wikipedia](https://en.wikipedia.org/wiki/Side-scrolling_video_game)
2. [Tower defense - Wikipedia](https://en.wikipedia.org/wiki/Tower_defense)
3. [Godot (game engine) - Wikipedia](https://en.wikipedia.org/wiki/Godot_(game_engine))
4. [GitHub - 维基百科，自由的百科全书 (wikipedia.org)](https://zh.wikipedia.org/wiki/GitHub)
5. [Godot Docs – 4.2 branch — Godot Engine (stable) documentation in English](https://docs.godotengine.org/en/stable/)