Game design report

Game name: Seek to survive

Long Liu 20104729

Software system practice Year 4

Date:

Supervisor: Mairead Meagher

1. Game Overview

1.1 Game Concept

This game is Horizon version adventure game. Players can move, left and right. Players have a base to upgrade abilities. The enemy will appear at a fixed range and attack the player. As time goes on, the player can collect materials to make tools and update modes. By these objects, payers can increase the maximum life value. When the game enters some specific days? Should this be 'places', some enemies will actively attack the player's base. Player needs to guard the base to avoid failure of the game. Player can gain materials by killing enemies and collecting 'collecting what?". 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 viewed from a side-view camera angle where the screen follows the player as they move left or right. [1]
- **Tower Defense**: **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 adventure games. It mainly refers to players who can accept pixel style, combat with enemies and discover new things. Targeting the market for high school students and college students.

1.4 Game Flow Summary

The game will load the main menu, where players can select new games, options, or archives. In the game itself, 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

I want the game to look and feel like a pixel adventure game. Similar to "Kingdom: New Lands(fig1)" or "Until We Die(fig2)". Here are some relevant games pictures:

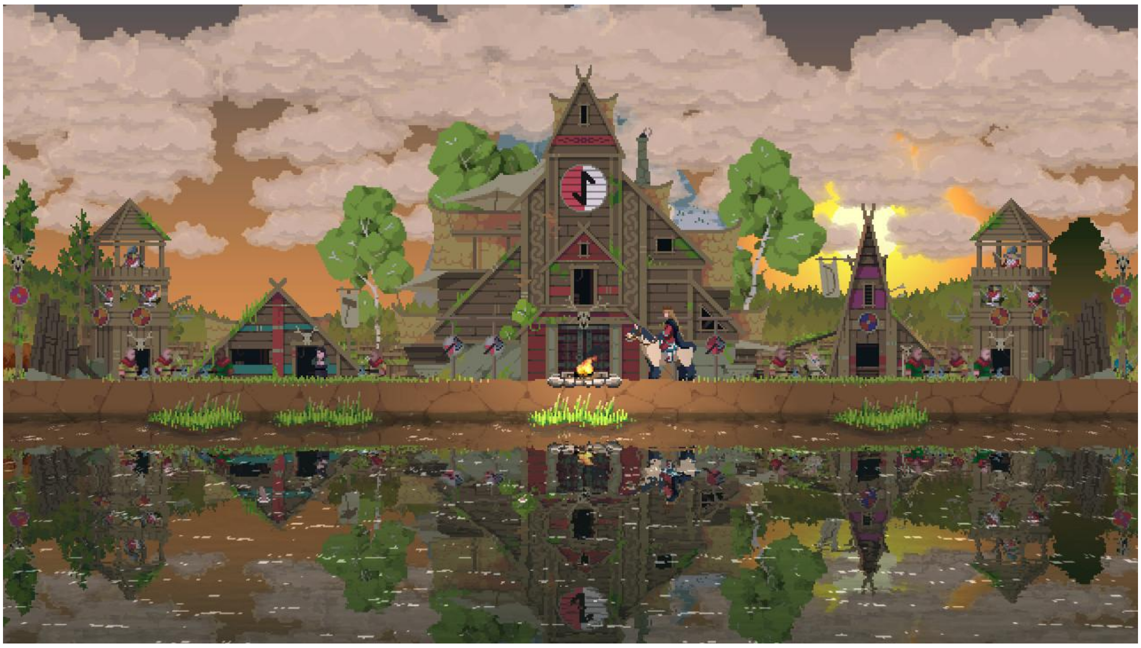


Figure1.1 Kingdom: New Lands

Figure1.2 Until We Die

1.6 Project Scope

1.6.1. Number of Location

Player and enemies will belong to an only one map. Player and base will appear on the centre of the map, and enemy nests will appear on the two ends of the map. Some broken buildings or hidden buildings will appear in the space between the base and the nests. Player can repair these broken buildings to get some annexes. These annexes can support some extra functions.

Annexe：a building that is added to, or is near, a larger one and that provides extra living or work space.

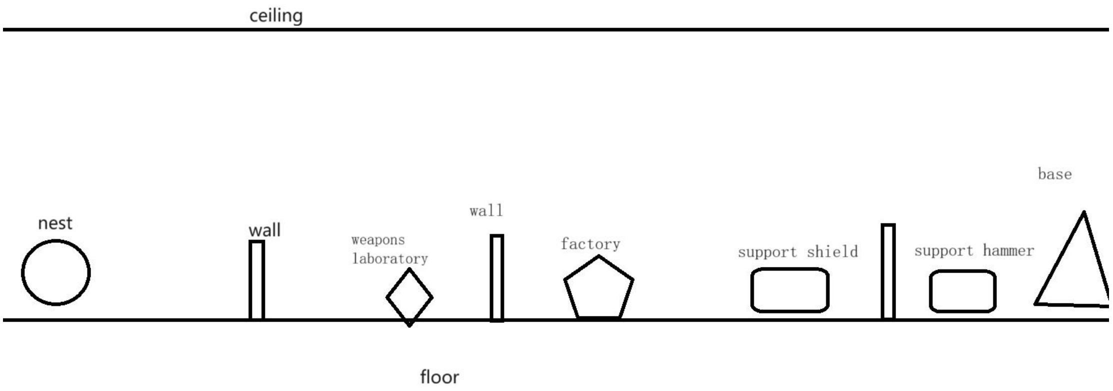


Fig1.3 Left half map

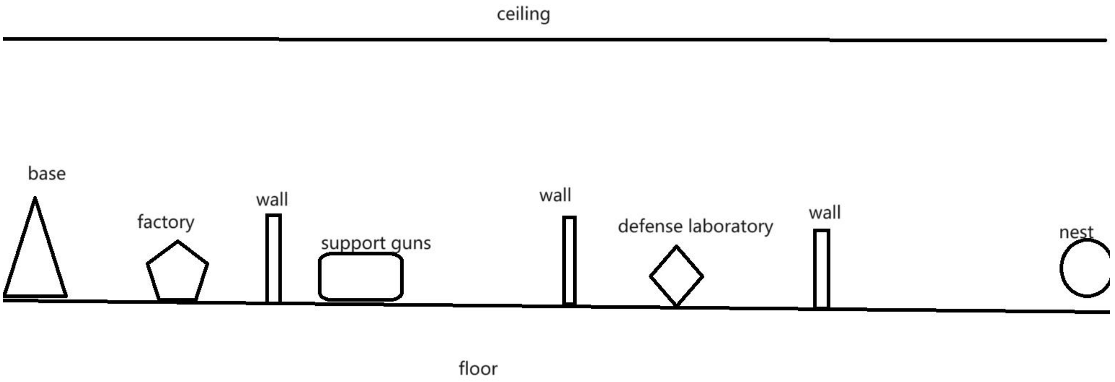


Fig1.4 Right half map

1.6.2. Number of Enemies

There will be three kinds of enemies.

- Crawler: attack from above, low health, fast speed , medium attack power
- Attacker: attack from frontal, medium heath, medium speed ,medium power
- Thrower: range attack, low heath medium speed, medium attack power

All enemies come from nest.

1.6.3. Number of Buildings

Base location: 1 base, 2 nests
Annexe types:

- 6 wall building points
- 3 kinds of tool rooms
- 1 weapons laboratory
- 1 defense laboratory
- 2 factories

Base: improve other buildings' max level, attract new refugees interval time, have 3 levels.
If base is destroyed, the game will end.
Wall: defense attack, has 3 levels

- **Weapons laboratory**: improve attack power, has 2 levels, need workers to finish update projects

- **Defense laboratory**: improve wall health, has 2 levels, need workers to finish update projects
- **Factory**: produce materials, has 2 levels, need workers to work
- **Tool room**: support hammers, guns, shield, need workers to work

Nest: enemies will appear there, have 3 levels.

1.6.4. Types of non-Player Character (NPC)

There are 5 kinds of NPC.

- **Refugee**: the primary level people.
- **Civilian**: refugee will be civilian after they get one piece of material.
- **Worker**: they build buildings, repair buildings, and need a hammer
- **Soldier**: they can range attack enemies. Civilian use a gun to be a Soldier.
- **Defender**: they have shields for blocking attack. Defender will use lances to attack, when they are behind the wall. Civilian use a shield to be a Soldier.

There are some neutral organisms.

- **Huge-mouse**: It will provide some materials after it dies.

1.6.5. Description of Player

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

2. Gameplay and Mechanics

2.1. Gameplay

2.1.1. Game Progression

Starting out, the player will appear at the base with some initial materials. These materials can be used to hire civilians. Subsequently, the player can create tools like guns or hammers by constructing corresponding buildings. This will change the jobs of the civilians: workers will operate the factory to gather more materials, while soldiers will hunt neutral organisms for additional resources. Meanwhile, nests will periodically produce enemies that attack the base. As the game progresses, the player will construct more buildings, and the nests will generate increasingly more enemies. The player's objective is to destroy the two nests before being overwhelmed by the enemies.

2.1.2. challenge Structure

Every night, the nests will generate a few enemies to attack the base. Every five days, they will produce a large number of enemies. The player needs to defend the base and eliminate all enemies; otherwise, the game will end. On the following night, when no enemies are generated, the player can take advantage of this time to explore the map.

2.1.3. Objectives

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

2.1.4. Play Flow

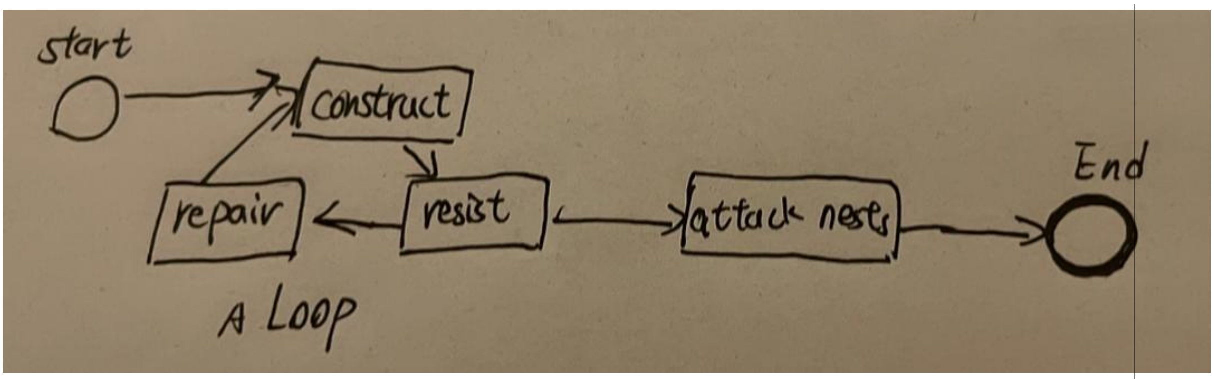


Fig4.1 play flow

The player can use materials to make refugees into civilians. They will leverage these materials to produce more tools, transforming civilians into workers and soldiers. Workers can build more structures after the materials for these buildings are provided. The player can construct additional walls and annexes. Once the base has enough buildings, the player can produce shields to turn civilians into defenders.

As time progresses, enemies will attack the base, and soldiers will defend against these attacks using the walls, ultimately eliminating the enemies. After the battle ends, workers can repair damaged buildings with materials. Through the cycle of development, defense, and repair, the player will accumulate enough soldiers and defenders to attack the nests. The player can summon combatants to launch an attack. The game will end in failure if the player's health points reach zero or if the base is completely destroyed. Victory is achieved when the two nests are destroyed.

2.2. Mechanics

2.2.1. Physics

Since this is a 2D game, the character can only move left and right. It does not account for height differences, so bullets will not fall. However, the bullet will still disappear after reaching its range. The game map consists only of x and y planes, so all physical calculations will be two-dimensional.

2.2.2. Movement

2.2.2.1. General Movement

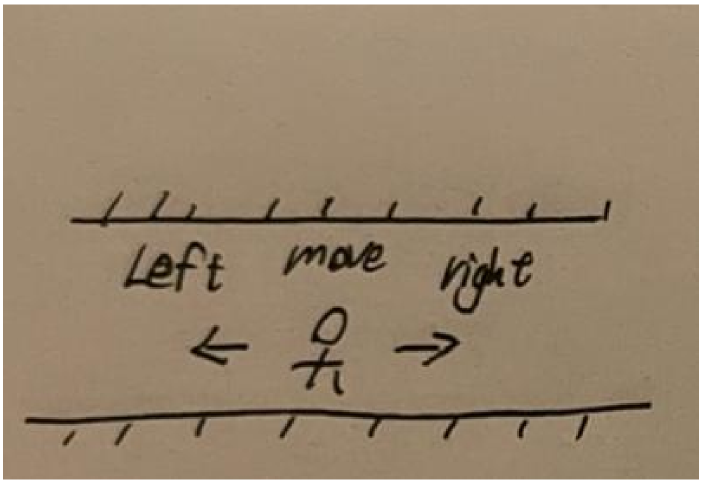


Fig4.2 movement

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.

2.2.2.2. Other Movement

- **Soldier/Defender**: They will stand after behind/on front? the wall. If the player buys some materials, they will move with the player.
- **Worker**: They will move to any places that need them.
- **Civilian**: They try to move the places that exist tools.
- **Refugee**: They randomly move in a fixed area.
- **Enemies**: They attack any buildings.

2.2.3. Objects

2.2.3.1. Picking Up Objects

Sometimes enemies will drop some materials, players need to get close to them to 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.

2.2.4.2. Aiming and Shooting

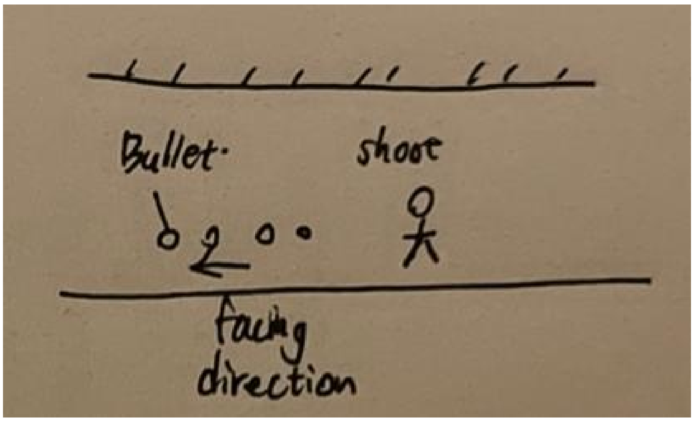


Fig4.3 shooting

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

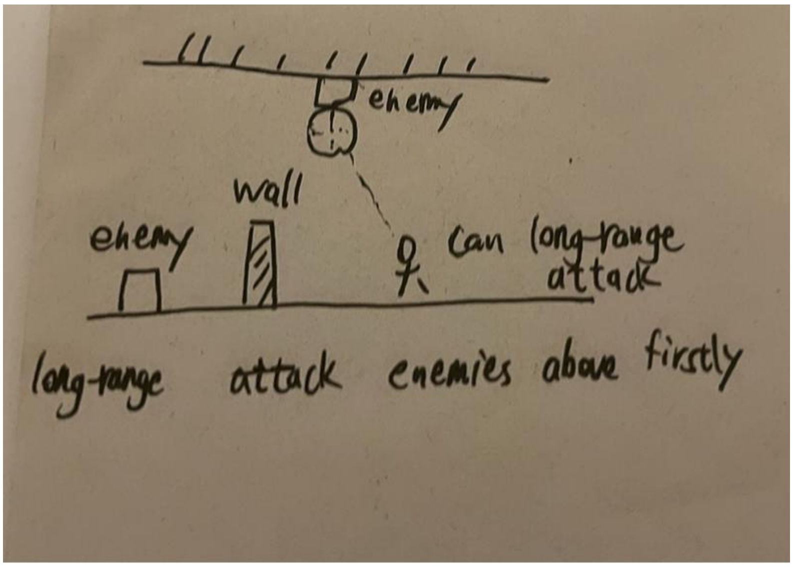


Fig 4.4 aiming

Remote attacks prioritize targeting enemies above.

4.2.5. Combat

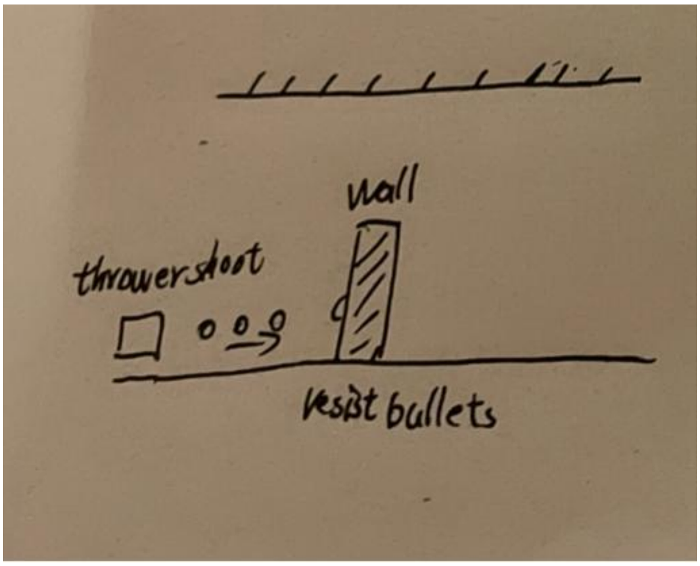


Fig 4.5 thrower attack

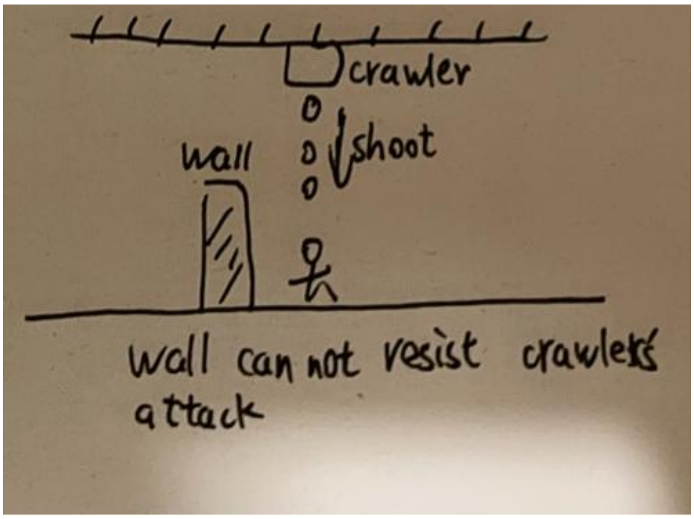


Fig 4.6 crawler attack

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:
Crawler: The enemy will attack from above. This allows their attacks to ignore the wall. But shields can still block their attacks. Only long-range attacks can harm them.

Attacker:
The enemy will approach and attack.

Thrower: The enemy will maintain a distance from the attacking target and shoot bullets for attack.

2.2.6. Economy

The player can get materials from factories and enemies. The player can Players can spend materials to build buildings and walls, produce tools, upgrade buildings and bases, and repair and rebuild buildings.

2.3. Screen Flow

2.3.1. Screen Flow Chart

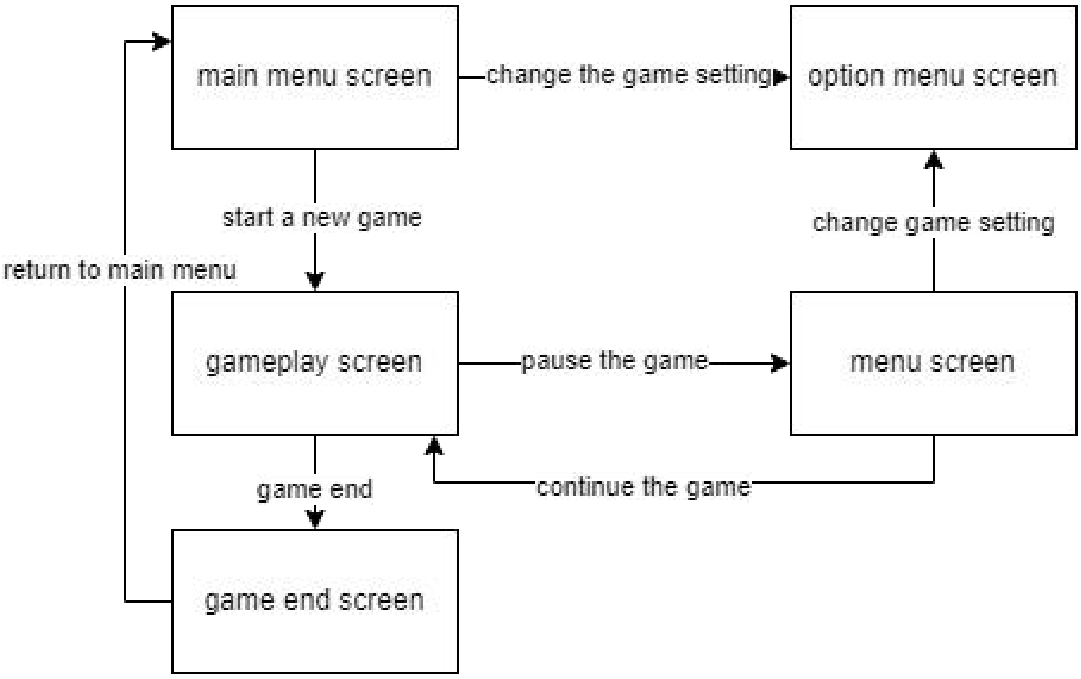


Fig4.7 screen flow

2.3.2. Screen Descriptions

2.3.2.1. Main Menu Screen

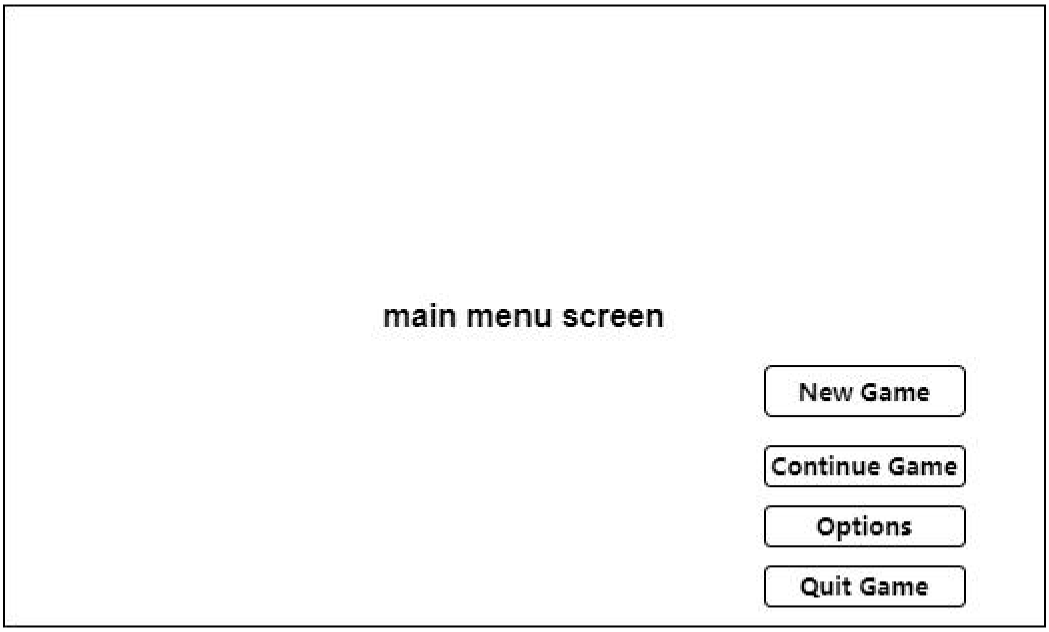


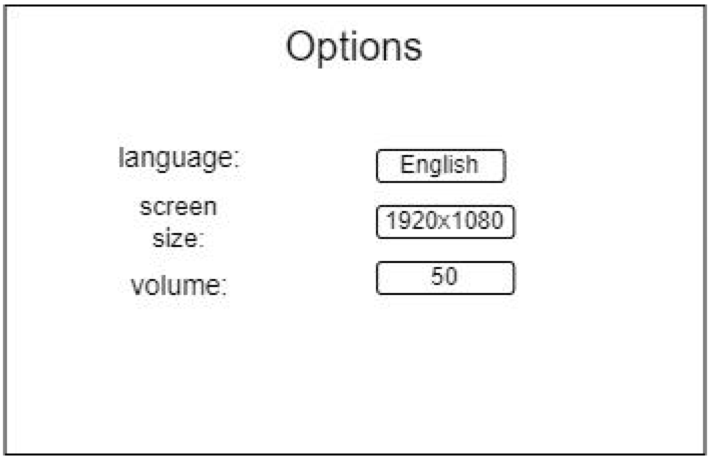
Fig4.8 main menu screen

It will show the game's title. And the screen will supply 4 options.

- **New Game**: Start a new game.
- **Continue Game**: Continue a game from a game save.

- **Options**: Open options menu.
- **Quit Game**: Kill the game process.

2.3.2.2. Options Menu Screen



Options

language: English

screen size: 1920x1080

volume: 50

Fig4.9 options menu

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

2.3.2.3. Gameplay Screen

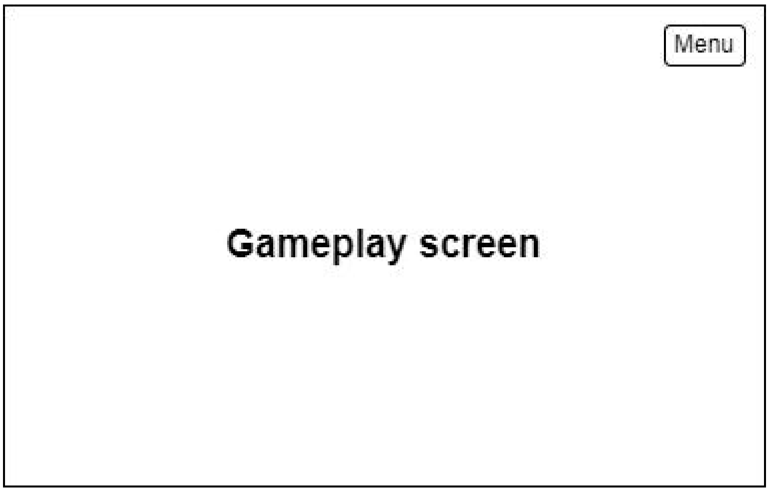


Fig4.10 gameplay screen

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

2.3.2.4. Menu Screen

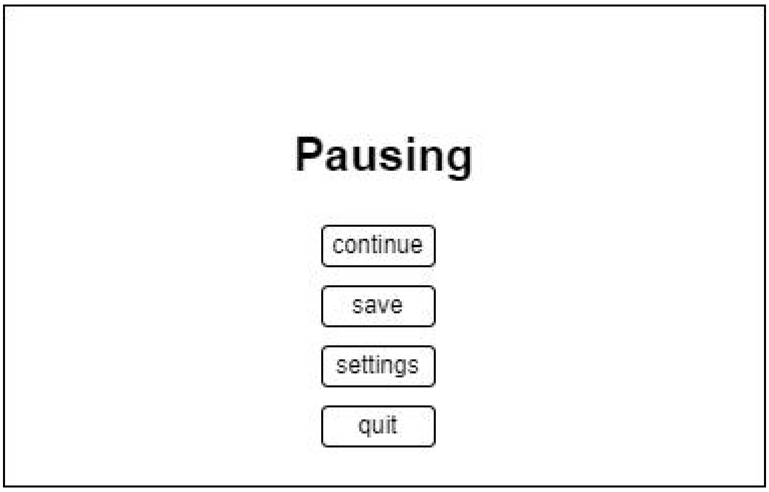


Fig4.11 menu

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.

2.3.2.4. Game End Screen

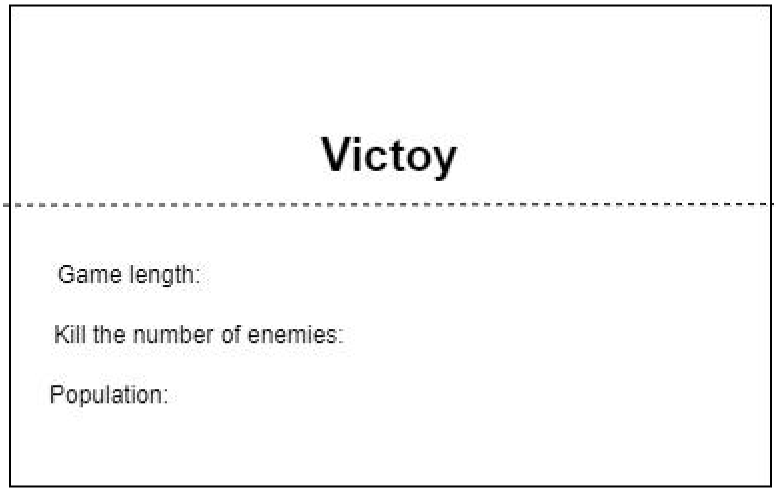
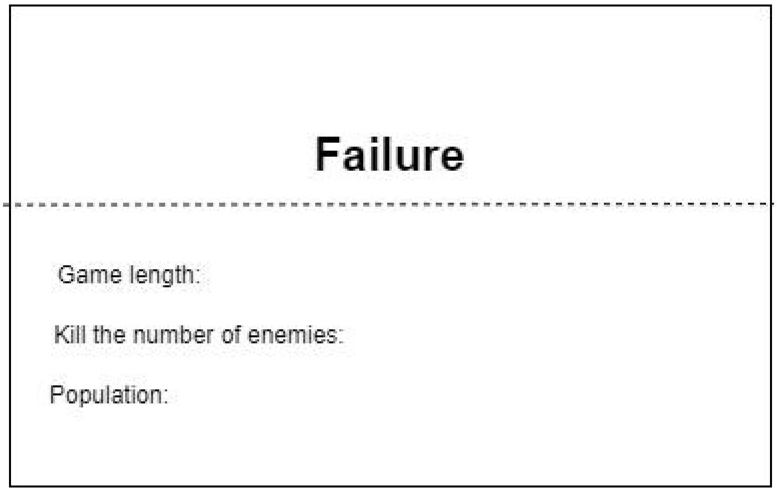


Fig4.12 game end of victory



Failure

Game length:

Kill the number of enemies:

Population:

Fig4.13 game end of failure

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

2.5. 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, enemies, and buildings. If the player continue game from a game save, the program will load all from the save file. But the game will not save actions, just save status. For example, if the thrower,a kind of enemy, is shooting when saving game, the thrower need re-shooting after the program load the game save. So, the information of bullets will not be saved.

3. Story, Setting and Character

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.

4. Interface

4.1. Visual System

4.1.1. HUD (Heads up display)

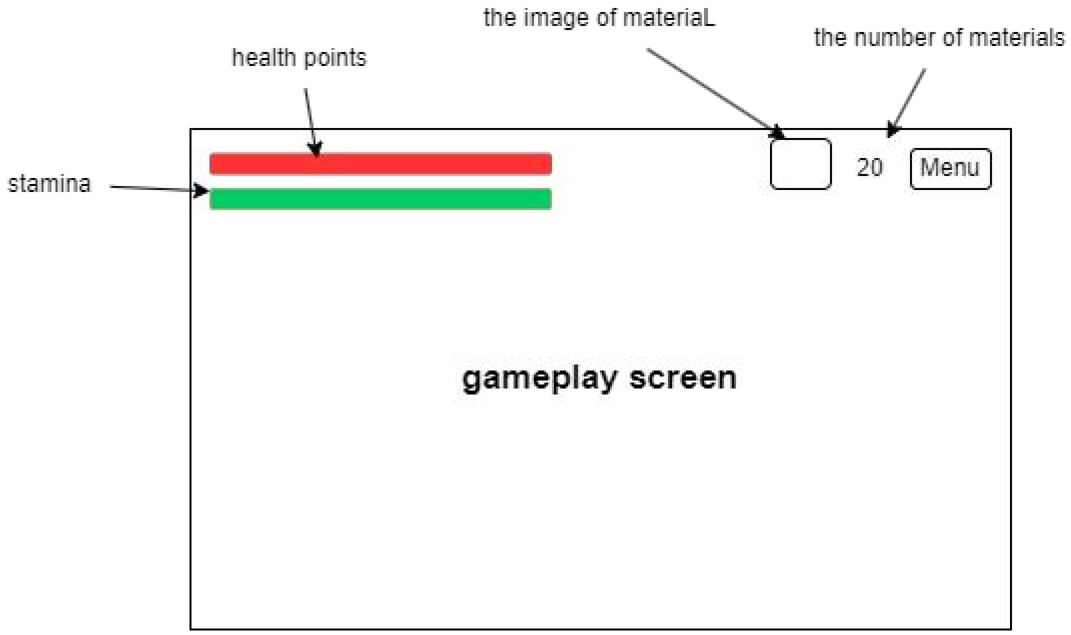


Fig 5.1 HUD

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.

4.2. Control System

The player use ‘A’ and ‘D’ to control the character’s movement. Clicking the left mouse button 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.

4.3. Audio

I will download free audio from internet libraries.

4.4. Help System

If the player enters a specific part of the game, the game will provide a short text notification in the upper right corner, such as a large number of enemies are coming. At the same time, some ICONS will also be displayed in the upper right corner to remind the player.

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 latest official version when I started developing the prototype.
Git,Microsoft Visual Studio,etc.

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 for this project is C#. Because it is one of the main languages of choice for scripting in the Godot game engine.

6. 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. In contrast to the waterfall model, agile development divides the development process into shorter iterative increments. At the end of each iteration, a final usable product is produced. If requirements change, the method can be developed in the next iteration cycle.

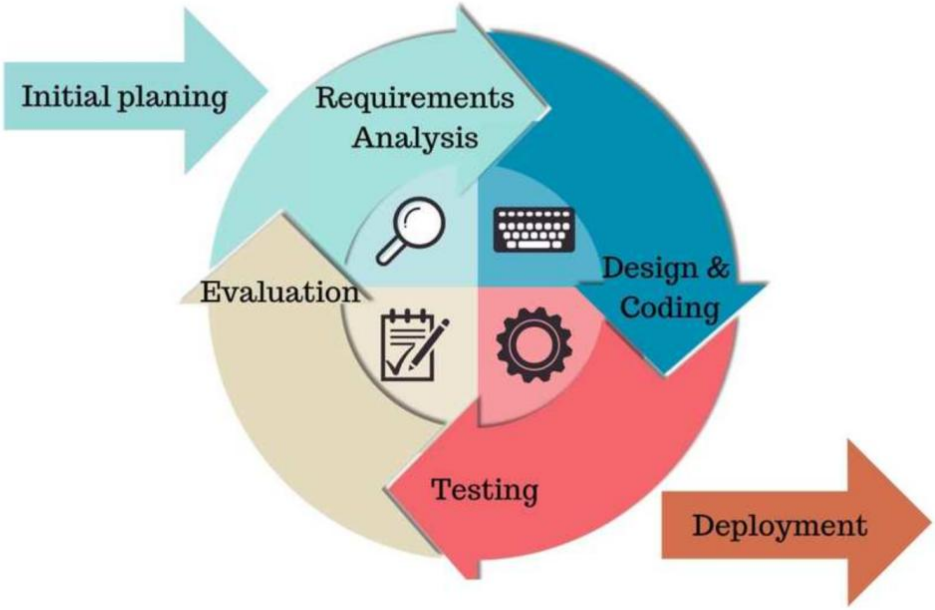


Fig 6.1 agile development

6.2. Detailed Schedule

	date	Sprint(2 weeks)	Goals	Tasks
1	2024.1.22	Prototype development and basic framework	Establish the basic framework and prototype of the game.	Develop the most basic game mechanics and a simple user interface.
2	2024.2.5	Core Game Gameplay Development	To achieve the core gameplay of the game	Encode the main game features, such as motion control and basic interaction.
3	2024.2.19	Map Design and Development	Design and implement preliminary game map.	Create map of the game and integrate them into the game
4	2024.3.4	Advanced features and interactivity	Add advanced gaming features and interactivity.	Implement special effects, complex game logic, etc.
5	2024.3.18	Testing and Optimization	To test the game and resolve any issues discovered	Conduct system testing, optimize performance, and improve user interface.
6	2024.4.1	Preparing for Beta Release	Prepare for the beta version of the game.	Make final adjustments and optimizations

6.3. Version control

Github is a hosting platform for open source and proprietary software projects. As an open source repository and version control system, Github has more than 9 million developer users. As more and more applications move to the cloud, Github has become the preferred method for managing software development and discovering existing code. 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	Risk Mitigation Strategy
Underestimating engineering difficulty	High	High	
Contracting illness	Low	Low	Avoiding Sickness
Inability to meet deadlines	Med	Med	
Lose of Report/Documentation	Low	High	Documentation stored on Baidu Yun Drive with regular Copies stored on several devices locally.
Lose of Project/Workspace	Med	High	Project stored on Baidu Yun Drive with regular Copies stored on several devices locally.kept on local devices.

6.5. Test Plan

	Test Plan
When?	At the end of every sprint.
Who?	Members of available game groups
Where?	Send testers a test version of the game via the game forum. The lack of connection between developers and testers facilitates critical analysis.
What?	Specific problems found during development.
How?	The developer will not be present and will provide playtesters with the game executable and a short questionnaire. Outside of the questionnaire, they can also make any complaints or suggestions they may have.

7. Appendices

7.1. References

1. Side-scrolling video game - Wikipedia
2. Tower defense - Wikipedia
3. Godot (game engine) - Wikipedia