System Requirements

Miremare

January 31, 2019

Abstract

This system requirements document covers the functional and non-functional requirements of our game.

Contents

1 Introduction

1.1 Aim

The aim of the project is to create a new Battle Royale + Funny 2D Game.

1.2 Overview of Idea

1.2.1 Battle Royale

Meanwhile, there are also numbers of Party Games released by companies which are more suitable for playing with friends together in any parties or while having group calls. Some good example of party games are **The Jackbox Party**, **Ultimate Chicken Horse** and **Human: Fall Flat**. The essential of the game is to let player play with or against their small group of friends and the purpose of the game is to let everyone felt entertained and less competitive.

1.2.2 Concept of New Idea

As we could observe that most of the Battle Royale game only limit to maximum of 4 players to play together and they don't have a dedicated mode to let small group of friends play against each other. Although they allow player to create a custom game that allow players to fight against each other but the original map would be too big for the small amount of players, thus not really suitable for the purpose. The new idea of our team is to develop a Battle Royale game that designed for a group of 9 friends or with other random person to play against each other in a party. The players will start off the game on a map with nothing. They will need to search around the map for weapons or armory. At the meantime, player can kill each other with weapon or bare hand. Instead of having shrinking circle on the map like other Battle Royale games do, we decide to use a new concept where we spawn a melee attack only AI every minute to kill the nearest player from the AI. Each minute the AI spawned will have longer hitpoint and faster movement speed. The last man standing will be the winner. Since the game is designed for small groups of friends or players, we decide to design the game more towards funny than competitive.

2 Functional Requirements

2.1 Player Object

- Multi-players (up to 8, at least 2) will be able to play this game through network. If one player disconnected from the network, the player will considered as left the game and result as lose. Other player will not affected by it.
- Players will be able to control a character walking in four directions by using the direction keys on the keyboard, which are up, down, left, and right. Players can also press up and left, down and left, up and right, down and right to move another four directions.
- Each character is randomly spawned in the map with 100 HP `[U+FF08]`100 Hp is the maxiumu health`[U+FF09]` at the begin of the game .
- After characters die, their game will stop and there is an image show that tell player that they lost the game. They are given options to leave the game.
- Players can quit in the middle of the game by closing the game window.
- Players have all the view of the map during the game.
- The last survived player is the winner. Screen will show an image that tell player they won the game.

2.2 Map Layout

- The size of the map is 1000px * 700px.
- The map will have three main objects:
  1. Ground.
  2. Trees.
  3. Obstacles.
- These objects are fixed on the map.
- The character can only walk on the ground.
- The character cannot walk through the walls.
- The bullet cannot shoot through the wall.
- Players will stay hidden when they hide in the shade of trees. However, players can be seen by AI players even when they are hidden.
- The map is not surrounded by walls.

2.3 Attack

- The players will not be able to collect weapons for this version.
- The number of bullets is unlimited.
- There is only one kind of weapon in this version.

Each player will be assign a weapon when the game start.

One character can only hold at most one weapon.

Players can not press the G key to drop the current weapon in this version.

The damage of the weapon is 10 for every hit shot.

Players can not switch their weapons in this version

Players are able to shoot in eight directions, which are up, down, left, right, left-up, right-up, left-down and right-down, by using the W, A, S, D keys on the keyboard. If two directions are contradictory(player press both up and down to shoot), it has no effects.

2.4 Defence

1. Medicines
   - The system will not generate medicines randomly in the map at the beginning of the game.
   - The system will give each player one medicine when the game start.
   - Each player can press the key "R" to use medicine.
   - Each player can only use medicine once in the game.
   - There is a count tells players that how many chance left to use medicine in current game.
   - Using medicines will help recovering players Health up to the maximum of 100.

2.5 AI (Computer-Controlled) Players

- The game will random spawn melee AI in the map, but only in the single player mode.
- AI can see all players on the map and will attack the closest player.
- AIs will not attack each other
- AI use the same weapon as players.
- AI players cannot use medicines.
- Player can eliminate AI by one shot.
- AI deals the damage the player 10hp for one hit shot.
- There AI will be spawned at the begin of the game(Single player mode).
    - All AIs have the same intelligence.
    - Only one kind of AI in this version.
    - AI will has an initial HP of 10 at the beginning of the game.
    - AI can be killed by the players.

3 Non-functional Requirements

3.1 Platform

- The game will able to run in Windows, MacOs and Linux system, test the game in both Systems.
- We can test this requirement to verify if the game is working properly.

3.2 Maintainability

- On a design perspective, we will develop the game in java.
- The code written for the game must be maintainable.
- Adding documentation will improve the maintainability scale of the system.
- The game will recover in 30 minutes if there is an unexpected issue happen.

3.3 Resources

- The game must be able to run with minimum 1024MB of RAM.
- The game must consume less than 300MB of the hard disk space.

3.4 Dependability

- The system shall be available 24/7 except certain times of maintenance.
- The unavailability of the system due to maintenance shall not exceed more than 30 minutes a day in total and 10 minutes at each time.
- The total time of unavailability or crush in one year can not exceed 3 hours.

3.5 Performance

- The network handler responses time shall not exceed 2 seconds after receiving the request from the user.
- The average response time between click and reaction must be less than 0.5 seconds.
- The maximum response time between click and reaction must be less than 2 seconds.
- After receiving the request from the user. The average response time between click and reaction must be less than 0.5 seconds.

3.6 Game Balance

- Have some basic tips and instructions in the game.
- The system shall be user friendly.
- Players with little experience of playing computer will be adept the game quickly with instructions provided.

3.7 Security

- The system should avoid any memory leaks.
- The system need to prevent any kind of unauthorized tampering or hacking.
- There should have an anti-cheat program to prevent some players to cheat or change the internal data.

3.8 Develop-ability

- This is the beta 1.0 version.
- The code need to be reusable for the future versions.
- Add the new features in the future.

[a4paper]article

4 Risk Analysis

Risk Category	Description
Technical	Complexity of the project develop, Requirements, Function, Interface, Quality, etc.
Organizational	Code integrate, debug, etc.
Planning	Estimation and scheduling for each part of the project.
External	The work efficiency of each members depend on the workload of different part. Affect the weekly team meeting or work by the personal events or plan, etc.

4.1 Risk level

- Low - Small issues on the program, will not cause critical problems.
- Medium - Some recoverable or predictable issues, bugs during the develop processing, it may affect the function or the quality of the program.
- High - Some critical problems, bugs. It may hard to fix. It may seriously break the quality of whole project.

4.2 Frequency of Risks

- Low - Very unlikely that this will occur during the development.
- Medium - There are 50-50 chance that this will occur during the development.
- High - Very likely that this will occur during the development.

4.3 Technical Risks

4.3.1 Programming

- Risk level: Medium
- Frequency: High
- The program needs to modify during the develop processing, add a new feature or a function may cause unpredictable issues (bugs, errors program breakdown).

4.3.2 Compromising on program design

- Risk level: Medium
- Frequency: Medium
- During the phase of coding, the requirement might conflict, or find that the final specification is unclear or even unreachable.

4.4 Organizational Risks

4.4.1 Integration

- Risk level: High
- Frequency: Medium
- There may have some issues, problems need to face when integrate the code from different members. or the different code standard lack of the code annotations. It may hard to integrate, cause bugs, errors, etc. Therefore, the program may not work probably.

4.5 Planning Risks

4.5.1 Scheduling

- Risk level: Low
- Frequency: Medium
- The schedule of the program develop may influence by the unexpected workload, issues, complexity of coding. Therefore, the progress may delay.

4.6 External Risks

4.6.1 External productivity issues

- Risk level: Low
- Frequency: Medium
- A group member may get ill, or has to face some unpredictable event during the develop processing, they may absent from the weekly team meeting, therefore, a team may not get the newest information or planning on time, and there work may lately submit, it will also affect the whole project progress.

5 Evaluation

5.1 Strengths

Good graphics, delicate interface, audio. It is a competitive game. Competitiveness.

5.2 Weaknesses

Just have basic function, need add more features. (items, new model, etc.).

5.3 Overall

From this project, it is extremely important to allocate the time and the work probably. And have a precise idea of the program from the beginning. Therefore, we will have a clear way to go, and will not stuck at the middle part of the progress and try to find another new way. And communication between each group members is a important role in a teamwork. Frequently communicate with other team members will help you to keep going and will enhance the quality of the work.