Course project: Requirements & Team Policies

Instructions

Produce the first version of your living document with the following sections.

1. Team info & policies

- List each team member and their role in the project.
  Wei Zheyang☐Programmer

  Li Xiang☐Game designer , Responsible for designing game scenes and characters, determining the rules and difficulty level of the game

  Lu Zhenghang☐ Test controler , Responsible for testing the game program, identifying and reporting bugs and other problems, and ensuring the stability and quality of the game program

  Gu Qian☐Art designer , Responsible for drawing game materials, including characters, maps, props, etc., as well as beautifying the game UI and visual effects

  Li Xinyu☐Copywriting editor , responsible for writing game introductions, instructions, help documents, etc., as well as text content required in the game

- Link to each project relevant artifact such as your Git repo (this can be empty for now).
  https://github.com/ENDVEN/Pac-Man/commits?author=ENDVEN
- List communication channels/tools with corresponding use policies :
  WeChat☐Baidu Cloud Disk

2. Product description

major features

- 1☐Character control

Character controls use x and y coordinates to operate. When the key on the keyboard is pressed and there is no obstacle above the current position of the character (x, y-1), the character will change the direction of the character's movement upwards and rotate the texture of the character; similarly, "down", " Both left" and "right" perform corresponding operations.

- 2. Eating and being eaten

When the elf meets a ghost, the elf can only be eaten by the ghost. At the same time, the elf loses blood, and the blood loss animation is played, and the blood volume is reduced by one. And when the elf eats the powerful pill before meeting the ghost, the elf can eat the ghost, making the ghost return to the haunted house to be reborn.
If the elf meets the bean and eats the bean, the score increases. If the elf eats the powerful pill, the state of the ghost will be changed so that the elf can eat the ghost.Secondly, it is necessary to control the movement of the elf so that it advances automatically. Also use the method of judgment, when it is judged that there is no obstacle in the direction of movement, the elf moves forward by one unit.

- 3. Animation effect

Define a variable, judge its value, and then choose to play the corresponding picture, and play the picture in a loop to achieve an animation effect. Similarly, matching the textures in different states enables the character actions and scenes to be played in an appropriate order.

stretch goals

- 1. Random movement of ghosts

For the random movement of the ghost, it is first necessary to judge the current direction of the ghost, and then judge the direction it can go, and finally randomly generate a direction in the direction that can go forward.

First determine the direction the ghost can move forward. Only when there are no obstacles in front, the ghost can move forward; judge the specific directions that the ghost can move forward according to variables and functions, one, two or three of the four directions of up, down, left, and right. If there are two or more possible directions, one direction will be randomly generated, and if there is only one direction, this direction will be selected.

- 2. Mapping generation

First draw by hand, make the map of the first level, and test the feasibility of playing the game. Subsequent level design will try to generate levels and mazes through random number seeds, and verify the level design quality and clearance rate of random number seeds.

Goals hoped to achieve

Provide an interesting and enjoyable game experience: It is hoped that through this project, an interesting and enjoyable game can be created, which can attract players' attention and allow them to enjoy the joy and excitement during the game.

3. Use Cases (Functional Requirements)

1. Wei Zheyang--Programmer

• Actors: players and enemies

• Triggers: The player eats all the beans and passes the current level

• Preconditions: There are beans and enemies in the current level

• Postconditions: The player passes the current level and enters the next level

• List of steps:

The player controls the movement of Pac-Man and gets points by eating Pac-Man

If Pac-Man touches an enemy, the player loses the game

When all the beans are eaten, the player passes the current level

The system prompts the player to pass the current level and enter the next level

The system generates the map and bean distribution of the next level

• Extensions/variations of the success

scenario: When a player passes the current level, the system will give an additional reward if they earn a certain number of points

• Exceptions: If the player loses the game, the game ends and returns to the start screen

2. Li Xiang -- Game designer

• Actors: Players and game programs

• Triggers: A new level of the game needs to be designed

• Preconditions: I have gathered enough information and materials, and I know about game engines

• Postconditions: A new level was successfully designed and submitted to the developers for development

• List of steps:

Determine the theme and difficulty of the new level based on the overall design and difficulty requirements of the game.

According to the theme and difficulty requirements, design elements such as maps, items, enemies, player abilities, and rules of the new level.

Using the game engine, combine and adjust the designed maps, props, enemies and other elements.

Write and adapt new game rules to keep players challenged and entertainded.

Adjustments and optimizations are made until the design of the new level is as expected.

Submit design documents and materials to developers for development.

3.Gu Qian -- Art Designer

- Actors: Players and games
- Triggers: The player enters the game and starts the game
- Preconditions: The game interface is loaded
- Postconditions: The player sucessfully starts the game and the art designer designs the game graphics to delight the player
- List of steps:

  The player opens the game app and waits for the game screen to load.

  The player clicks the "Start Game" button.

  The game enters the official game interface, and the beautiful game graphics begin to be displayed to the player.

  The player starts playing the game.

- Extensions/variations of the success: Players can adjust different game backgrounds and themes
- Exceptions: The game screen failed to load and the player could not start the game

4.Lu Zhenghang -- Test controler

- Actors: Players, games, and testers
- Triggers: The tester receives a test task and needs to test the game
- Preconditions: The game has been developed and installed in a test environment
- Postconditions: Testers were able to complete the test, and the test results proved that all the features of the game worked
- List of steps:

  The tester logs in to the test environment.

  The tester opens the game app and waits for the game screen to load.

  Testers test various features of the game, including starting the game, actions during gameplay, game over, and more.

  Testers record test results and generate test reports

- Extensions/variations of the success

  scenario:Testers can test game modes of varying difficulty

- Exceptions:The game had a serious bug that caused the test to fail

5.Li Xinyu -- Copying Editor

- Actors: Gamers and copywriters

- Triggers:Copywriting editors need to prepare introductory copy for the game

- Preconditions: The development and art design of the game has been completed, and the game graphics have been produced

- Postconditions: The copy editor successfully wrote the game introduction copy for the game, and the copy was able to attract more players to try the game

- List of steps:

  Receive game introduction materials and game screen design drafts provided by the game development team.

  Read the game's introduction materials to understand how the game is played and what it features.

  Write game introduction copy based on game introduction materials and game screen design drafts.

  Submit the game introduction copy to the game development team.

  The game development team reviews and adopts the game introduction copy

- Extensions/variations of the success

  scenario:Copy editors can optimize existing game intro copy to make it more engaging and readable

- Copywriting editors can write relevant copy for games, such as game guides and playbooks

- Exceptions:The game introduction copy contains errors or inaccuracies, resulting in a bias in the player's understanding of the game.

4. Non-functional Requirements

• Scalability: The project needed to support different levels of game difficulty to meet the needs of different users

• Usability: Games should be easy to understand and use, with simple controls and an intuitive interface to get started quickly. The game should also have a good feedback mechanism and hints to help users better understand the rules of the game and how to do it

• Security and privacy needs: The project should ensure the security and privacy of user data. User data should be stored and transmitted encrypted to prevent data breaches

5. Team process description

- □□□□□□□□ Specify and **justify** the software toolset you will use.

We will implement the game using the C++ programming language, the OpenGL graphics library, and the SDL game development library. We will also use graphic design software to design the game's graphical interface. and we will use Visual Studio as the development environment and Git for version control and collaborative development.

- □□□□□□□□ Define and **justify** team members' roles.

Our team is made up of a group of developers who share a common philosophy and passion, and we all have a wealth of experience and skills. Our team is very united, everyone is very capable and motivated, and is committed to developing high-quality and interesting game products.

- □□□□□□□□ Provide a rough schedule for each member (or sub-group).

Phase 1: Game Design (1 week)

Determination of game ideas and concepts

Design of game mechanics and flow

Game screen and sound design

Game level and storyline design

Phase 2: Game Development (2-3 weeks)

Implementation of game logic and algorithms

Production of game interface and sound effects

Production of game levels

Phase 3: Playtesting and Optimization (1-2 weeks)

Playtesting and Feedback Gathering

Game bugs and performance optimization

difficulty setting

The total time required is about 6 weeks, and the specific time depends on the actual development difficulty. At the same time, we should also pay attention to the arrangement of time to ensure the progress and quality of each stage.

- □□□□□□□□ Specify and explain at least three major risks that could prevent you from completing your project.

1、The map generation algorithm is more difficult

Automatic map generation is a relatively complex algorithmic problem, and there are many factors to be considered. Such as the size of the map, the distribution of beans on the map, the location of walls, the location of enemies, and so on. To design an efficient algorithm to generate a map, it is necessary to consider the interaction and influence of these factors, and at the same time ensure that the generated map can ensure the playability and fun of the game.

2、There may be bugs in the map generation algorithm

Due to the high complexity of the map generation algorithm, some unexpected bugs may appear. For example, there are walls on the map that shouldn't appear, or enemies can't reach certain areas, etc. These problems need to be discovered and fixed in time, otherwise it will affect the playability and player experience of the game.

3、Collision detection algorithm complexity is high

The Pac-Man game requires a lot of collision detection, such as the collision between beans and Pac-Man, the collision between enemies and Pac-Man, and so on. These collision detection algorithms need to be designed to be efficient and accurate, otherwise it will affect the fluency and experience of the game.

- □□□□□□□□ Describe at what point in your process external feedback will be most useful and how you will get that feedback.

The most useful feedback is during the game testing phase, because at this time the game has a certain degree of playability and integrity, and testers can play the game and provide detailed feedback.