Gypsophino

Group 27

Contents

1 Group Number

Group 27

2 Group Members

Student Name	Student ID
Haoxiang Fei	519370910099
Qin Huang	519021910485
Simin Wang	519370910102

3 Project Name

Gypsophino

4 Intended Language

C++

5 Summary

Our project is a kind of music game that can be played on the computer.

6 Motivation

We want to design our own music game because all members in our group love music. And we'd like to have some new with music games in general, such as pixel blocks with a major tone and a new storyline.

7 Tentative Design

7.1 Features

7.1.1 Note Hitting

Our software is designed to provide player a basic, intuitive music game playing experience. And the core function of a music game is note hitting. Colorful notes will fall down from above, and player need to press down certian keys on the keyboard at certain instant of time to get points.

7.1.2 Music Map Management System

Our software is designed to be capable to handle multiple music maps. Player is allow to select the music map and play. Players can also compose their own maps and share it with friends.

7.1.3 Score System

Our score system is designed to be able to calculate score for any single valid hit of note and give the player instant feedback. The score system is also able to store the history score of the player.

7.1.4 Settings

Default settings of the falling speed of the notes and the offset may not fits everyone well, so player can modify these basic settings and volume of the effect sound and the music in our software.

7.1.5 Different Difficulties

One difficulties of a piece of music is not enough. Our software should be friendly to the new music gamer, and be challenging to experts. A auto difficulties multiplexer may be also include to reduce the workload of map writers.

7.1.6 Story Line

Pure music game might be too hardcore. A music game with story might be oriented to a larger group of people. Our story system can have multiple branch according to players choices and performance in playing. And it is also has multiple ends waiting for players to explore.

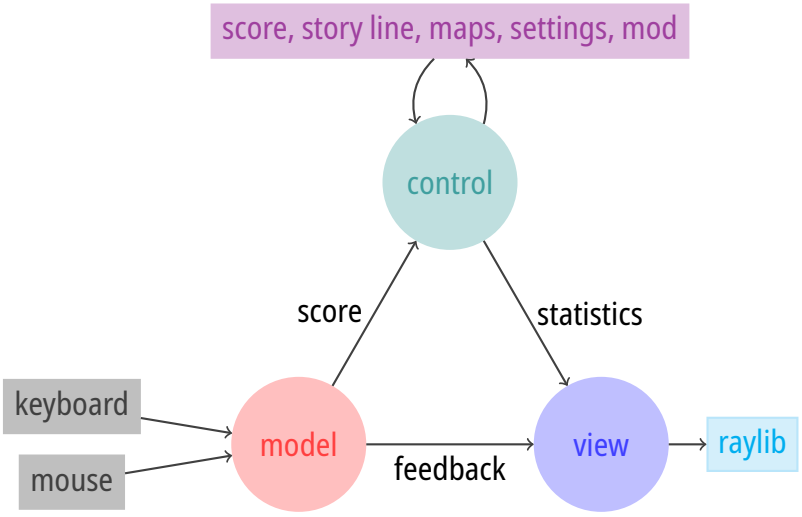
7.1.7 Mod System

Our game allows player to develop and use mod to create additional content of the story, or re-organize the note arrangement of maps to more challenging or more easy, etc.

7.1.8 Multiple Player System

Our game allows player to play this game with their friends locally. Two players can also cooperate to finish a song.

7.2 Diagram of Our Software



8 Expected Outcome

8.1 Bottom-line

1. basic operating of a rhythm game
2. score system
3. basic settings (volume)
4. ability to import pre-composed map (manually)
5. cover picture

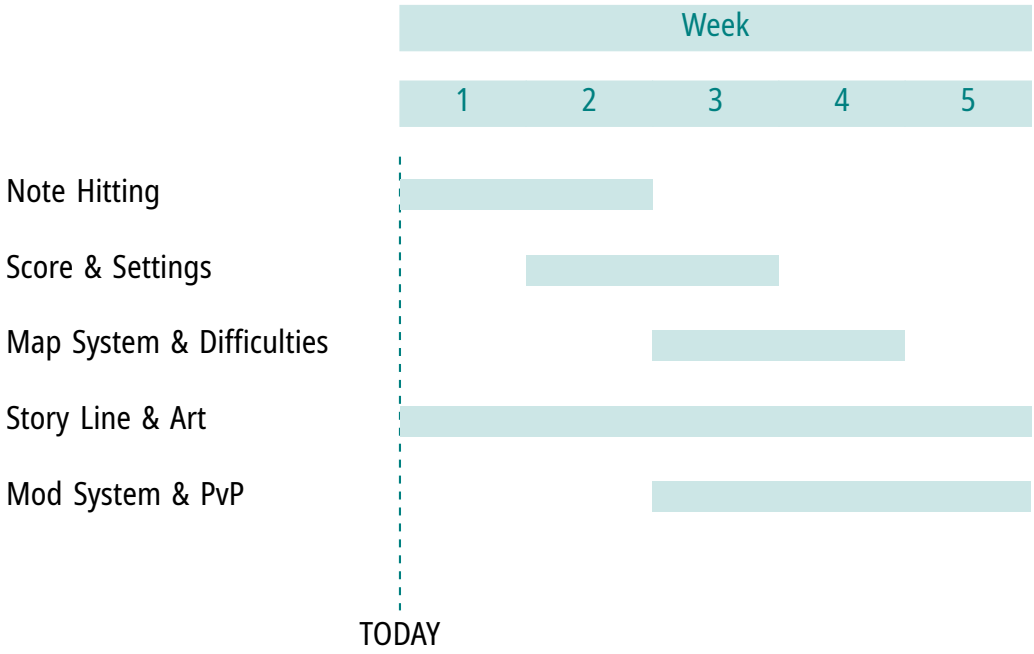
8.2 Expected

1. story line
2. plot pictures (CGs)
3. difficulty level
4. advanced settings (user can customize the offset between the music and graphics according to their experience)

8.3 Potential

1. MODs (like invincible mode)
2. Versus mode

9 Timetable



10 Extra Perparations

To process the incoming keystorks while drawing, we need to learn the knowledge about the asynchronous communication and multi-threads.

To write well-structured code, we need to learn some software engineering method like OOP.

To draw graphics, we need to learn that how to use OpenGL or other computer graphics library/framework.

Since our members don't share the same operating system, we need to have a basic understanding of the platform compatibility.

As we need to manager the maps and track the game score, we need to learn the organization and the basic usage of the database.

11 Task Assignment

Fei HaoXiang

1. Note hitting
2. Advanced settings
3. Versus mode

Wang Simin

1. Write story
2. Basic settings and maps
3. Maps system
4. MODs

Huang Qin

1. Draw cover and plot pictures
2. Score system
3. Set the beats
4. UI (including integration of stories and images)