COMP7604 Game Design & Development

Monster Secrets

Group 7: Li Xinwei (3035454949) Zhang Zihao (3035454951)

1. INTRODUCTION

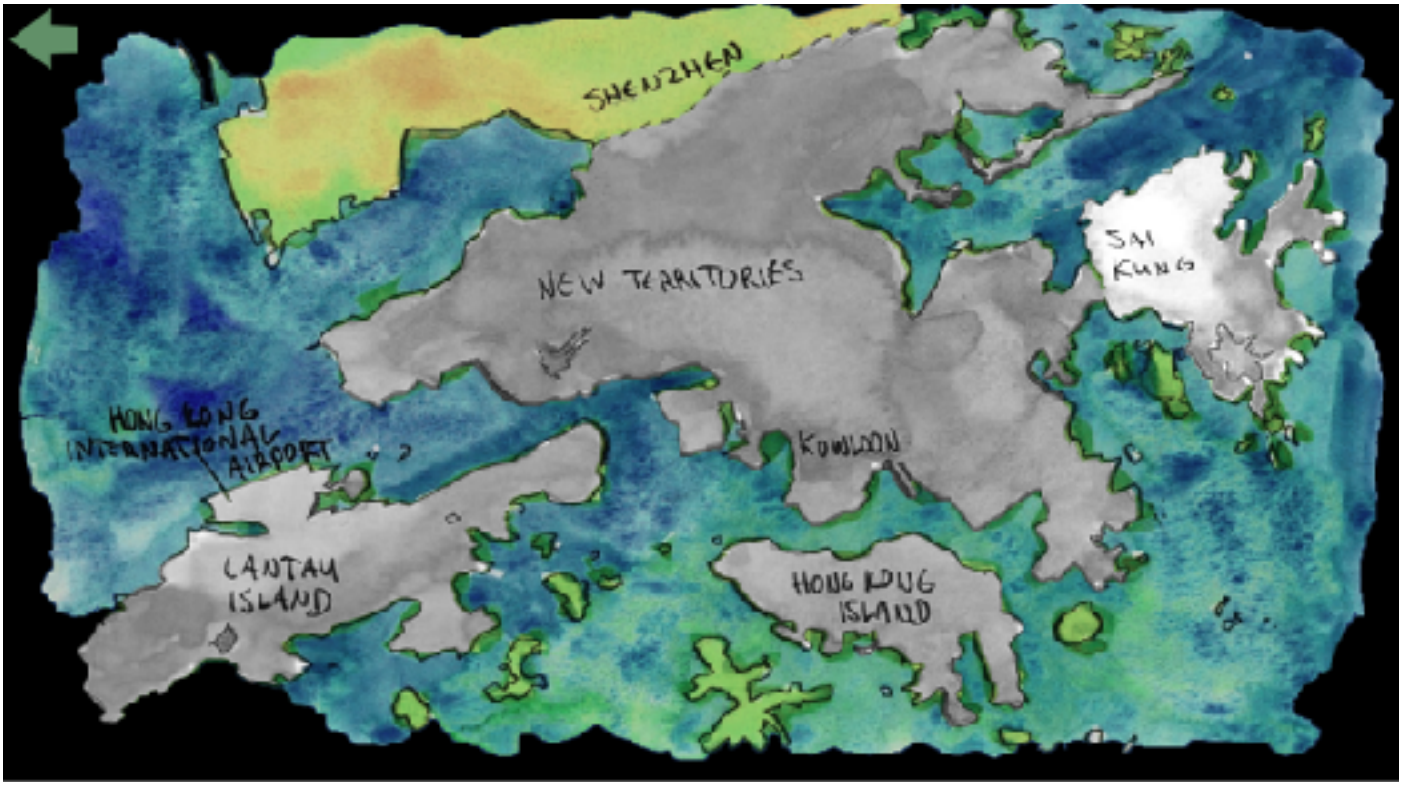
This report is about a video game. It contains the idea and plan at the very beginning, the development platform and resources that the game used and the division of labor of each member. The game has been completed and is planed to publish on APP Store and Google Play.

2. GAME DESCRIPTION

2.1 Game background

Once upon a time, Hong Kong was a place full of magic and without human beings.

A flock of magical creatures lived happily. They looked cute and clumsy but they had extremely strong power of magic. Later they were called little monsters by human after being found, but that was a thousand latter. They suffered being ruled for a period of time just before human discovered them, by a giant, evil monster, who coveted their power and tried to steal from them.



The evil monster used an old but powerful spell to curse the little monsters, they would lost themselves if they were cursed for a long time and their power would be used to serve the dark.

A human wizard wanted to free the little monsters. He had to figure out the old spells and used the same spells to free the poor little monsters.

The player will be the human wizard in the game to set those cute creatures free and helps them to expel the dark, evil monster.

2.2 Overview

This game is a mobile leisure game which involves map exploration and elimination. The map is a cartoon HongKong map which is divided into four parts, Hong Kong Island, Kowloon, Sai Kung and Lantan Island.

Players are free to select any part of the island to start a new game with different game scenes.

There will be a tutorial for beginners if the player is the first time to play this game.

2.3 How to play

During the game, the player need to draw the corresponding symbols above the bouncing little little monsters. There is a line of dots below the indicates the number of symbols the player need to draw to free the monsters.

Each time the player draw a correct symbol, he will gain a score.

To make the game more fun and challenge, the generation speed of the little monsters is growing with time, on the other hand, after free a number of little monsters, the evil monster will appear as a boss to stop the player to recuse little monsters, carrying with a large amount of spells.

The player need to beat the boss to save the rest of monsters or the game will end. If the player missed more than five monsters to rescue, the game will be ended as well.

During gaming, the player could click the button at the top left of the screen to quit the game, then, the player could resume the game or end the game to the map scene.

3. GAME FEATURES

3.1 Gestures recognition

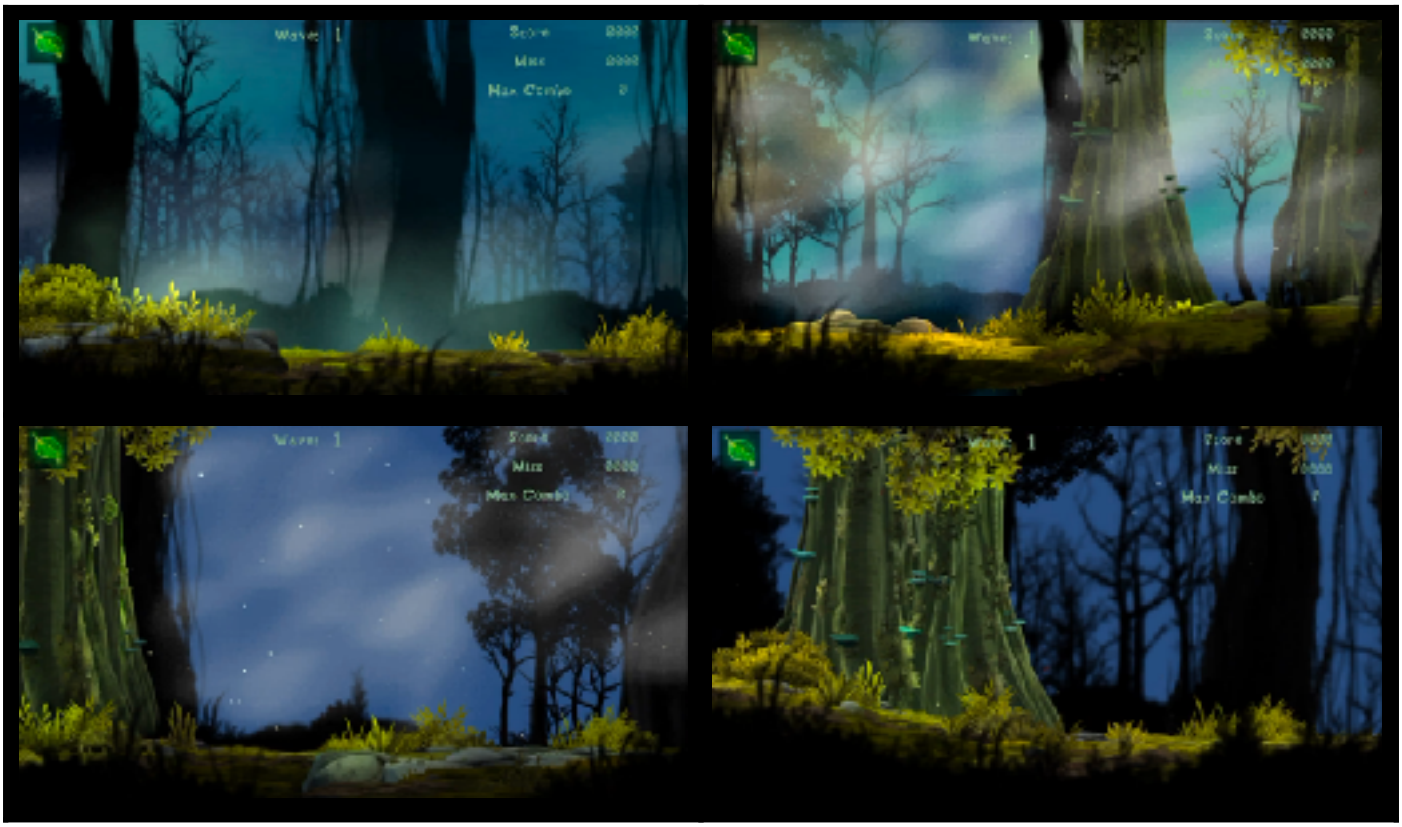
Gesture recognition is used to detect the spell player draws.

When a monster comes with a symbol, player should draw the corresponding symbols. If the symbol player dawn matched the monsters, player will get 100 score. If player correctly drawn symbols within 3 second, the action will be considered as a combo which will give extra marks.

14 symbols will be shown in the game.

3.2 Game scenes

Built with a resource from unity asset store, which includes basic sprites of backgrounds, foregrounds, trees, plants and floors. The raw sprites were separated by sprite editor, and be used to make up the four different scenes. Since the whole style of the game scenes were designed as a mysterious forest, the game scenes surrounded with moving fogs which makes it more mysterious.



3.3 Monsters generation

The bouncing monsters automatically appeared from left hand side of the map. There are 8 kinds of monsters combined with the former symbols.

During each period of generation, we need to dynamically load a monster, symbol and a little dot which indicate how many symbols should be drawn to dismiss the specific monster. Those monsters will also automatically embedded with a animator to control its bouncing and dismissing animation.

4.GAME TECHNIQUES

4.1 Gestures recognition technique

Gesture recognition technology is one of the most important part of our game. Fortunately we found a existing unity assets have already did the same thing as we planed to do, we just use the library as our gesture recognition system. This gesture recognition library is called Gesture Recogniser and downloadable from unity assets store.

This library allows us to use their symbol patterns or generate our own gesture files which is very useful for our game. Basically this library is very easy to use, we can build our recognition system as follows:

1) Create gesture files using the menu “Assets/Create/GestureRecognizer/GesturePattern” , config the id property, add lines (simply click on the grid to connect lines), set other properties if needed.

2) Create a canvas image using the menu “GameObject/UI/Image“, add a DrawDetector component, add Line Object, fill other properties.

3) Add EventSystem Object

4) Write scripts to deal with ‘OnRecognize’ event.

4.2 Dynamic Resource Loading

Dynamic resource loading can be achieved by unity prefabs. To do this, we have to convert our monster into a prefab resource and put it into the Resource folder. Anything in the Resource folder can be dynamically loaded during the game. Because of these, we can successfully load the monsters.

5.DIFFICULTIES

5.1 Make the recognition algorithm works better

It is very hard to made a algorithm with high performance which means it can detect symbols drawn by different people. At the very first time, we are tried to train our own model use TensorFlow.

Our plan is as follows:

1) When player draw a line on screen, record the points player touched. 2) sample the points and make it 96 points at total.(using padding if points less than 96).

2) Collect training data with labels and split part of training data as test set.

3) Build a RNN model to fit the data.

Later we found a library did almost the same thing except it need a large training set to perform better. However the more symbols come, the more difficult we have to get the huge amount of training data.

So we later decided to use alternative libraries. Fortunately there is another library do not need that much training data to have a good performance. After a quick research, we use this new library to achieve recognition.

5.2 Tune the monster & boss generation time

After the game play testing, we realized the game is too easy for players to get a high score. So the first idea is to get the monster generated faster and faster. But is still difficult to find a function to get the game balanced.

Finally we got a function: monster_generation_interval = times_of _generation ^ (-0.2) and boss_generation_interval = 1.9 * wave^2.1+5 satisfied our need.

6.TASK COMPLETED

1. Gesture recognition (Variables tuning, pattern drawing, symbol designing)
2. Monster prefabs
3. Maps and Scenes
4. Buttons and Game logic
5. Monster Animations
6. Monster Dynamic Generation (with symbols)
7. Single Monster with Muti-Symbols
8. Boss Generation
9. Scoring System (with combo computing)

10. Game variables tuning

11. Game Ending

12. Game Tutorial

13. BGM

14. Logo

15. iPhone & Android Deployment

7.REFERENCE AND USE OF EXISTING RESOURCE

- BGM:
  - The poison forest - maple story
  - https://soundcloud.com/mmomusic/maplestory-poison-forest-1
- Monster character (dark-monsters-mega-pack) :
  - https://assetstore.unity.com/packages/2d/characters/dark-monsters-mega-pack-74035
- Font:
  - JazzCreateBubble
- Gesture Recognition Library (gesture-recognizer)
  - https://assetstore.unity.com/packages/tools/input-management/gesture-recognizer-86410