Assignment Report

Game Name

Back to Nature

Student Names

Name	Class
咸博文	1603105
肖松	1603108

New features and Implementation Methods

1. New game name and backstory

We design a novel name and a little story for the game. This will make our game more attractive. You can see the story in the start menu.

2. All-sided prompts and hints in game play

- You can see hints of game UI and operations in start menu clearly.

- You can see section descriptions and encouragement at the beginning of each section.

- Background music messages are showed in the end interface.

We write these messages in two ways:

1. Write messages on the background image directly, this way is easy but useful. Messages in the start menu and end interface are showed in this way.

2. Welcome and introduction messages to every section are showed by onGUI() fuction in a C# script when a scene is loaded. This way can control the time messages continue easier.

3. Clearer and integrated UI operation

- In the start menu, you can see 'Start' and 'Quit' buttons clearly.

- When you are playing the game, you can leave game play directly to the start menu by clicking the button 'Menu' on the top of interface.

- In the end interface, there are three buttons:'Menu','Quit','Retry'. You can do anything you want.

  We add the 'Menu' button in each game scene because we think people should have a way to quit when they are playing the game.

4. Graceful game scene with beautiful images and rhythmic audios

- There are four different background musics for menu and three chapters. Both the rhythm and the name are suitable for the game.

- There will be a hit sound when player knocks into an obstacle. This sound can give a hint to player that he has knocked into an obstacle.

- We use a little animation to simulate the running action of the player.

5. More levels

There are 3 chapters in this game.

- Chapter 1 is an introduction to this game.

- Chapter 2 is a common scene with faster speed and more game props.

- Chapter 3 is endless with increasing speed and all game props. Players can enjoy and challenge the game in this chapter.

- All 3 chapters are connected by the backstory. Three scenes are continuous over time.

6. More game props and operations

- Besides normal jump, we designed two new operations: big jump and crouch. Players must use all three operations to pass obstacles.

- There are three game props in this game: medical kit, shield and decelerator.

    - Medical kit can add 1 point HP.
    - Shield can defend one obstacle.
    - Decelerator can lower the move speed of player.

7. Pretty score system

We give a real-time score which reflects the length of ground that player has passed in the game. Player can also get his final score in the end interface. We use number images to show the score, which is attractive to players.

8. Random varied obstacles and game props

We generate obstacles and game props randomly. This measure give this game more variablity. Besides, we design three kinds of obstacles to make players using all operations.

Difficulties and Solutions

Difficulty	Solution
In chapter 3, we generate game objects persistently. If the player has played a long time, the game will stutter and become very slow.	We add a script to every generated object. In this script, we destroy the object in OnBecameInvisible() function.
Distance between obstacles shoudn't be too short or too long, and we must keep the randomness of the game at the same time.	We calculate a random scoped length after generating one obstacle and then generate next obstacle by this length. The kind of the obstacle is random, too.
The positions of game props are difficult to determine, we must generate them in positions where players can reach.	In chapter 2, we generate game props neer the ground between two obstacles. But because the speed of chapter 3 increases slowly, generating game props between two obstacles after playing for a while always make the player knocking into an obstacle to obtain the game prop. So we decide to replace obstacles with game props randomly(20%) in chapter 3. This measure makes players easier to obtain game props.
Generate ground blocks persistently in chapter 3.	We generate 50 ground blocks at first, and then generate a trigger with very large height. If the player passes a trigger, we generate a new block.
We use canJump flag to judge whether the player can do an operation at first, but we found that if we click one key fleetly, the player can jump in the air. We didn't found the source of the problem through the WEB.	At last, we judge this by comparing the player's real-time position Y with ground's position Y. If real-time position is higher than ground's position, we ban the jump operations. This method solve the problem perfectly.
In this game, we often change the speed of player, main camera and background. This is troublesome.	We move player, main camera and background into one game object and let the new game object move by the corresponding speed. This measure let the change of speed easier.

Difficulty	Solution
Use of game props	We set some flag variables to judge the state of the player.
Transmit score value between game scene and end scene.	We create an object in prefab and write a C# script for it. In C# script, we call DontDestroyOnLoad() function in Start() function and declare a private int value to remember score. When we change scene from game to end, we generate this object at first, then we can get this object with player score in the end scene.

Contributions

We discussed and made this game together. After our discussion, the detailed contributions are showed here:

Name	Percent	Contributions
肖松	45%	Game design. Generate game props, ground blocks and obstacles randomly. Key control of operations. Game props' effects.
咸博文	55%	Game design. GUI adjustment. Score system. Scene change. Music and image selection. Story author. Report and comments writting. Game parameters adjustment.