CM4021  
Games Development on the iPhone

Redhwan Nacef

1103186

# Table of Requirements

|  |  |  |
| --- | --- | --- |
| Requirement | Summary | How Met |
| 1 | Scenes | The two levels of the game and the level selector are all different scenes. |
| 2 | State | A game state system which handles if the player has died or not, and how much percent they have gone through the level. |
| 3 | PhysX | The players will be implemented as rigid bodies |
| 4 | Interaction | The players will be transported if on a transporter. |
| 5 | Levels | Levels selected when in the level selector scene or when completing a level |
| 6 | Persistence | Game stores the percent complete of each level and whether the next level has been unlocked |
| 7 | Scripts | The following behaviours are implemented:   * When the player moves, the platform moves * When the player is on a teleporter cube, the player is teleported to the destination cube * If the player hits the boundaries or falls off the path, the game is over. * If the player reaches the end of the level, the next level is unlocked and loaded. |
| 8 | Input | Using the arrow keys and WASD move the player and the platforms. |
| 9 | Ray-casting | The players use ray casting to check below them, if the terrain is directly below them, then the game is over. |
| 10 | Audio trigger | When the player goes on a teleporter, sounds are triggered. |

# Discussion of the Game

The game I am developing for this coursework is a very simple but difficult (I hope) game. The user must navigate two balls through two paths independently to reach the finishing line and progress to the next level. The key mechanics of the game are as follows: at the beginning of the game, the user will be presented with the Main Menu screen, after any key is pressed, the user will see the level selector screen. This screen will show all the levels in the game and the highest percentage complete of each level that has been unlocked. A level is only unlocked if the previous level has been complete. The player will be navigated to the respective level by clicking on it. In this screen, the player will see two paths, one on the right and one on the left and a ball at the beginning of each path. The screen will begin to move forward through the paths as soon as the player hits any key. By controlling the balls, the player must navigate their way through the path using the added aidsto help on the way. The player will control the left ball with “WASD” controls and the right ball with the Arrow keys.

There are two ways to die in a round of the game. The first way is if either of the balls fall off the platforms, the second is the bottom of the screen reaches either of the balls. When the round is over, the player will be shown the percentage of the level completed and will be given the option to retry.

As mentioned before there will be a couple of aids to help along the way. These are:

The Teleporter: The teleport is simple, and represented by two cubes, one with glowing coloured balls going into the cube (indicating this is the cube to teleport you) and one with the same coloured balls but going away from the cube (this will be the destination you are teleported to).

The Moving Platform: This is simply a platform that moves back and forth between two points in space and acts as a bridge to each ball. The trick part about this platform is that it is also controlled by the user. For every move the left ball makes, the platform on the right will move and for every move the right ball makes, the platform on the left will move. The user will have to control these platforms in order to help the two balls get across a space in the path.

# Overview of the Accessibility

Throughout the development of the game, accessibility will always be an important factor to consider when making decisions and when coding. Although all these decisions may not be implemented for this coursework, the game will be designed in a way that will allow changes to be easily made in the future. Below is a list with the different accessibility measures and how the game has implemented them or prepared for them.

**Motor:** This game is designed to have simple and familiar controls that people are used to. In order to allow better accessibility, the controls can be allowed to be remapped to the users’ preference. Also allowing multiple input devices would be considered.

**Cognitive:** The User Interface will be made as simple as possible, with minimum navigation. To improve this sections, a “Help” page could be added to allow a full explanation on how to play and how the game works. The speed in the game has been set as a public variable. This will make it easy to adapt it to a menu in the future to give full control to the user.

**Vision/Hearing:**  All writing will be done with care using an easily readable font and with clear simple language. Adding options allowing the font size/colours/brightness to be adjusted would be considered. No essential information is given by the SFX or music in this game. If there were any instruction given by a voice, subtitles would be provided and background music would be quiet to not interfere.

# User-Testing Plan

There are many different ways to test a game. User testing is one of the most effective and reliable ways. In order to test my game, I would firstly run some tests of my own first. These would be functionality tests. I would create a test plan using my games discussion. I would go through this and test out every aspect and mechanic I planned to have in the first place. For example, the Menu navigation, the controls, test each way the play can get game over, check the scores are saved, the audio and the visuals.

After this, I would create a testing plan for user testing. In order to do this, I will create a document explaining exactly how the game should work. The key mechanics behind it (similar to the first section of this document) and how all the gameplay elements work. As well as this, I would add a questionnaire for the user to fill in after they have played the game. This would be a short questionnaire asking the user what they enjoyed about playing the game, if they noticed any bugs and general feedback. I would then have to find the right group of testers. I would ensure to include testers that play a lot of games as well as those who only occasionally play. The reason for this is I would like to get the opinion of every type of person. My first thought on this would be family and friends as they are likely to be willing to perform the test, my second thought would be the internet. I would post a test version of my game on a popular site (such as Reddit/Facebook etc..), I find that users over the internet will be a lot more critical than family and friends which may prove to be useful when looking for feedback in the game. After finding the groups, I will just let the testers play the game for as long as possible. The more they play, the more chance of noticing bugs as well as retrieving more feedback. Taking the average feedback from all the testers would give a good idea of the progress of the game and if there are any problems with it.