Anthony Tiongson

Professor Mubbasir

CS 448 Computer Graphics

September 15, 2020

I. https://tonypwny.github.io/CS448B0/

II. The game follows the project's specifications.

Player 1 controls: W, S, A, D keys to move; spacebar to jump.

Player 2 controls: Arrow keys to move; right shift to jump.

I used metallic coloring along with free textures and free skybox assets to liven the look of the game. I display some helpful information on the game's starting lobby screen for people who do not know the rules of the game. I used a high risk, high reward design for the distribution of pick up items in the level. For player collisions, I used a simple offset so that a player has to be slightly more higher than the other rather than literally higher to allow more tolerance in collisions. The offset still allows a player add a higher speed to penalize another at a lower without jumping, so players cannot rely solely on never jumping. To make the game more exciting and aggressive, I use the distance in height between players to calculate how many points get deducted (with the max height collision being around a whopping 55 point reduction). I tried my best to follow an object orientated approach when coding the game.

III. The first extra credit involving implementing announcements was an easy task after accomplishing some of the basics for the project. It was essentially combining the tasks of displaying information for a given time (such as player score) and the

usage of the passage of time as a source of information (such as displaying game time). I put a variety of announcements based on different things, such as giving a warning a when the half way point is reached and poking fun at a play a player if they received a point penalty with a current score of 0. With more time, I would like to implement more than one location where announcements occur and also at variable locations.

Regarding experimenting with multiple control interfaces, I found the mouse input (via my trackpad) provided a very interesting and fun way to control the players. I think a game using this kind of input would be great. The problem, however, is that this kind of input is only available to one player for most machines. It is a severe disadvantage to have a player controlling via keys play one with a mouse/trackpad, so I set my game for both players to only use keys to be fair. Unity's InputManager made experimentation and setting controls very easy/intuitive.

For the local/online multiplayer, I started to follow tutorials on Unity's deprecated Multiplayer packages. Unfortunately, I was not able to compile a working WebGL or macOS build at all with the package in use. This game exists as a shared screen experience as of now.