Daniel Yan Wu

1631295

Programing Techniques and Applications

420-LCW-MS section 1

**Final Project: Video Game**

13 pages

Work presented to

Robert Vincent

May 16, 2018

The game is made on *Unity* version 2017.3.1f1 and the scripts are coded in C#. The game building starts off in a *Scene* where the assets can be added, the camera can be moved to wherever they should be. There is an inspector menu that does most of the work. After selecting an object in the scene, the inspector can be used to add scripts, certain physics and many more Unity supported actions. There is also an *Asset Store* which has both free and paid assets that other users can use. One of which is *TextMesh Pro* that makes the texts more adjustable instead of the basic fonts and options given by Unity. In the *Assets* window, everything is sorted by their type. The scenes can be swapped between by opening them; they have a Unity icon.

The scripts made are a bit confusing from the learning process. The scripts called *camera1,* *CameraC* and *PlayerController* are not being used. The actual player controlling script is called: *testcontroller* because it turned to be a lot closer to what was intended. The goal was to get the camera to rotate the player, but that proved to be a lot more difficult than expected. Instead, the camera rotates on its own and so does the player. In the *testcontroller* script, the first thing is movement, the script takes advantage of the built in *Character Controller* to take into account collisions and a vector is made for the y-axis to implement and easy jump function. The rest of the movements are done using *transform* which basically moves the object in the scene. There is an attack function that currently only prints out that the attack connected, no damage is done.

The next important script is *cameraController* which deals will camera movement, it uses Cursor.visible to toggle between the visibility of the cursor, Quaternions for rotations and LookAt to keep the camera facing the player.

The *Stat* script within the *Stats* folder is a base class for the *CharacterStats* script and is serialized, meaning the fields in this class will show up in the inspector; it takes care of the damaging taking that is currently not implemented. The original goal was to add equipment that would have altered the damage values. The *CharacterStats* script uses the damage for the health bar and dictates whether the player is dead or not.

The last 2 scripts that are used in this *test* scene is first of all *PauseMenu*, which uses *SceneManagement* to be able to go back to the main menu, to pull out the dead screen, to pause the game by setting the time scale to 0 and to retry. The final one is *Animations* which uses the *Animator* window in *Unity* to play certain animations when the conditions are met.

In the *Menu* scene, the *MainMenu* script allows the buttons to either play the game or leave the game, the option button is dealt with in the inspector. The *OptionsMenu* script uses an AudioMixer to be able to change the volume of the program and a Dropdown to be able to change the resolution of the window.

In its current state, the game is not playable. The opening screen is a menu to adjust settings like volume, which is implemented but there is no music, and to play or to quit the game. After clicking play, the program jumps to the next scene, which is the game itself, the original goal was to have a sort of character selection screen or an equipment page. In the game, the damage does not do anything and the animations are not synced with the attack cooldown, otherwise by waiting long enough, one after the other, the animations will play when the left mouse button is pressed. By pressing the “escape” key, the pause menu pops up and the game can be resumed or quit. Currently when the “t” key is pressed, the player will take damage and the health will go down. When the player dies, the dead menu will appear.

**Works Cited**

<https://youtu.be/Ta7v27yySKs>

<https://youtu.be/S2mK6KFdv0I>

<https://docs.unity3d.com/ScriptReference/Transform-position.html>

<https://docs.unity3d.com/ScriptReference/Transform-eulerAngles.html>

<https://docs.unity3d.com/ScriptReference/Transform.RotateAround.html>

<https://youtu.be/FhAdkLC-mSg>

<https://youtu.be/YOaYQrN1oYQ>

<https://youtu.be/zc8ac_qUXQY>

<https://youtu.be/JivuXdrIHK0>

<https://youtu.be/9W0xLonwbLo>

<https://youtu.be/wdOk5QXYC6Y>