Dennis Mesina

Assignment 1

CS485 Game Programming

Assignment 1

The game I created uses free 3D models and animations from the Unity Asset store. The player controls a unit on the screen using the WASD + spacebar for manipulating the character and enemies. W is for forward movement, A/D are for turning the character, and S is for backward movement. The spacebar is used to “attack” the enemies. The player starts out with 10 hit points which are displayed on the screen. The number of enemies remaining is also displayed on the screen. Once all enemies are destroyed, a message is displayed that the player has won the game. If the player runs out of health before all enemies are defeated, then the player loses the game and a message is displayed.

The assets are all free from the Unity Asset store, and I can’t really say I copied any other resources, because I mostly just googled things for a more in depth understanding and wrote the code myself. The only real tutorial I followed was the one assigned for the Roll-a-Ball portion of the assignment. When I accessed some of the free resources, I imported a 3D humanoid and decided that I would use the mouse as input for the controls and design a game where you simply destroy monsters. After implementing a mouse controller, I realised that the choice was not very intuitive so I switched over to keyboard input as the interface. When choosing the combat system, I figured projectiles would be a more complicated aspect and so I decided to stick with melee combat. That ended up proving difficult as well, as I had to provide plenty of if statements regarding the placement of the units in proximity to each other when regarding their ability to attack. The animations were a new challenge to me as well as I’ve never been on the production side of a game before and have always taken it for granted being a consumer.

As a result of not following a tutorial or simply modifying someone else’s game, I feel like I’ve learned a lot. The trickiest part was learning how to reference certain aspects of the game that were built separately. For example, if I designed a class, but wanted to call that class from a different object, I would first have to access the object itself, and then call the datamember or function of the class I’m attempting to reach. Unity’s GUI is very handy when modifying the public variables so that we don’t need to keep constantly going back to the code when testing graphical aspects of the game.