Mock Fighting Game

This is a mock fighting built using Unity and C#. The point of the project it to gain experience and test out possible ways to run a game. Features that are currently in the game and that will be added in the future will not be the most efficient solutions. My goals are to get it running and see if the current features work before modifying the code and making it run faster and smoother.

Currently Running

Currently working features are movement, block detection, attack inputs, and character stats. All inputs are currently programmed to run off of a controller/arcade stick. Input checks for special move motions. Example the classic Street Fighter motions are implemented. Quarter Circle Forward motion are detectable and a Boolean is currently displayed to show if the inputs were correct and will be removed once proper animations are added in.

Features to be Added

Animations will be added alongside proper hit boxes and damage/meter calculations. A more lenient input detection; currently you need to press (PlayStation controller) Square and X at the same time to perform a grab. Some leniency would be added as to not make it so exact/difficult. Different meters to show off health, super moves, and stuns. Dashing will be added and a set jump trajectory. Inverting movement once the players switch sides. Displayed inputs. Lastly over all improvements to the code to make it run more efficiently. Last thing would be to program an AI to fight against.

March 18, 2021 Update

This update adds in the damage and super meter calculations along with the graphical display of the meter and health bars in the HUD. Both increase and decrease accordingly when the opponent takes damage. More changes were made to the Universal script. Reducing the amount of code needed for it and not needlessly setting stats, i.e. jabs or kicks that are already set up in their individual character scripts.

The Next Update

The next update will consist of adding more moves, animations, and a combo system working.

March 9, 2021 Update

This update is to reduce the amount of repeated code as possible. I moved controls i.e., movement, button inputs, save, adjust, and read through the directional list. With this code moved to the universal script I will not need to re-write the controls for each new character added. The controls will check the input list to verify if a special move should be animated if not it would go to a standard attack animation. If the special move involves a projectile it will return a string to the character’s script so that it may spawn the correct projectile.

February 25, 2021 Update

In this update I added in an xbot model with animations acquired from <https://www.mixamo.com/>

A simple idle motion was added with walk animations and attack animations. I added a hit box in the attack animation that is active in a number of frames before deactivating. The attack button calls a method that checks the directional inputs of the player. If a Quarter Circle Forward motion is inputted before the attack button is pressed then a simple sphere object is created instead of going to the jab animation. The sphere travels and when it hits the opponent it destroys itself. Both the sphere and the jab animation have a hit detection and currently display on the console that they have made contact with the dummy model. The code was cleaned up and some unnecessary variables were removed.   
In the universal.cs file recovery, start, active, reset, jumpRecovery, isJumping, and isAttacking were removed. isAttacking was a test to make sure the attack input was correctly working. Now that the model has an attack animation playing properly it was unnecessary to have it. Start, active, recovery was originally there to help determine the animation and when to activate the hitboxes of each animation. These were redundant once I learned that you can simply add them and activate them whenever you wish in Unity’s animator tool. In addition, made the code more organized and added in proper getters and setters.