The component I am creating is a first person player controller, including player movement, jumping, and camera control.

The first thing to tackle is movement. To do this I create a function that detects directional input, and clamps the input so it can then be scaled by a speed which can be inputted/edited in the Unity editor. This is all contained in an if statement that checks if the bool “canMove” is true (which is set to be the case on Start) so that we can easily add elements that restrict movement in the future.

The next part of this script will create the camera control. For this, I have made the camera a child of the player so they move together. Then for the look direction the script reads the cursor location and moves the camera in the corresponding location. This function is tied to a sensitivity value which like the movement speed can be edited in Unity. The Y value of the camera's movement is clamped into a certain range to avoid unruly camera behaviours such as spinning or turning upside down. This is also all contained in an if statement checking that the bool “canLook” is true in the same way and for the same reason as the “canMove” if statement. Outside of this statement, on Start the cursor mode is set to Locked so the player can’t move the cursor outside of the game window, creating undesired camera behaviour.

The final part of the script enables the character to jump. The first part of this section checks if the player is on the ground. To do this a raycast is used to check if there is an object with the “groudLayer” tag close under the player. The exact distance that it checks is governed by “groundCheckLength”, another value which can be edited in Unity. Then, if the script returns that the player is grounded and the space key is pressed, the script adds velocity to the Y axis of the player using the “jumpPower” value, which again can be changed in the Unity editor. If the player is registered as not grounded then pressing space will do nothing, which avoids unintended double jumps or similar issues. Once again this is all contained in an if statement that checks for the value of the “canJump” bool to be true.

This script is added to a capsule to act as the player, with the camera set inside the capsule for a first person view. The capsule needs a rigidbody component attached to it to function correctly. Finally, a plane is added to the scene with the layer set to “groundLayer” so it can be checked by the jumping portion of the script.

Log:

Researching this component took about 1 day

Creating and debugging this component took about 2 hours