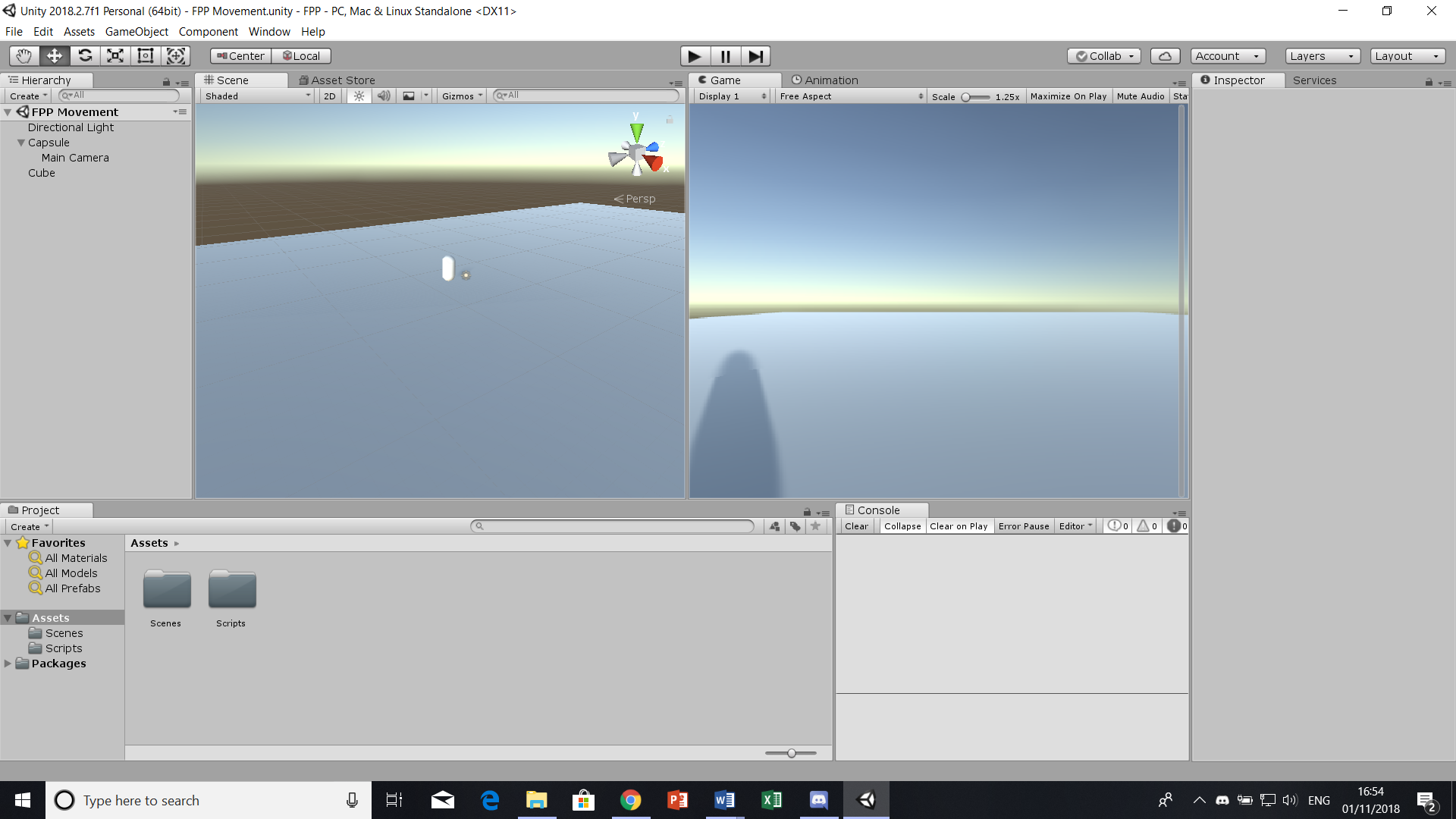
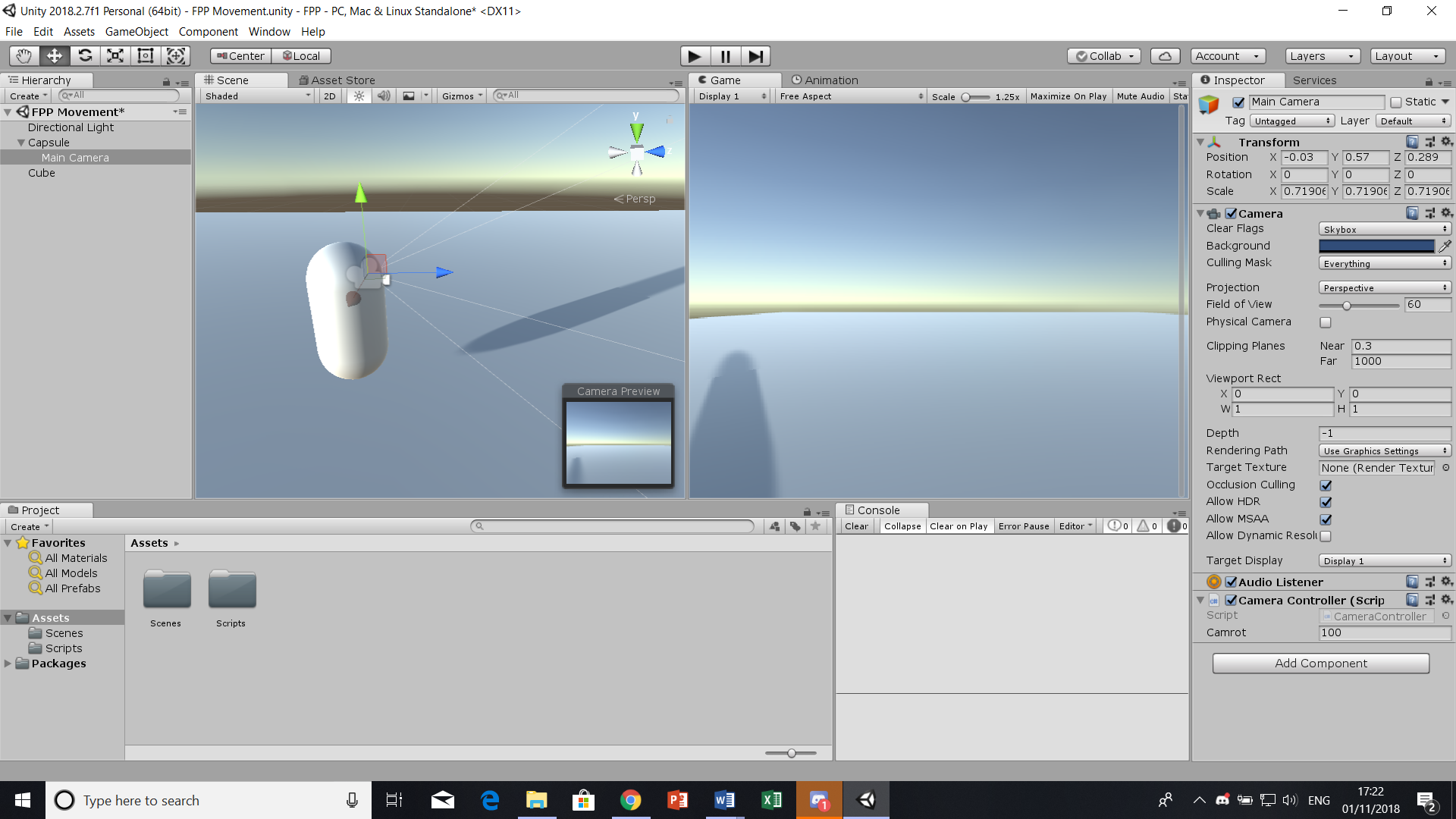
This script will demonstrate how to set up a first-person perspective camera and player movement

**Step 1 – Creating the scene**

To begin, you must first make the scene. This relatively simple scene only requires the assets seen on the hierarchy located on the left side of the following screenshot:



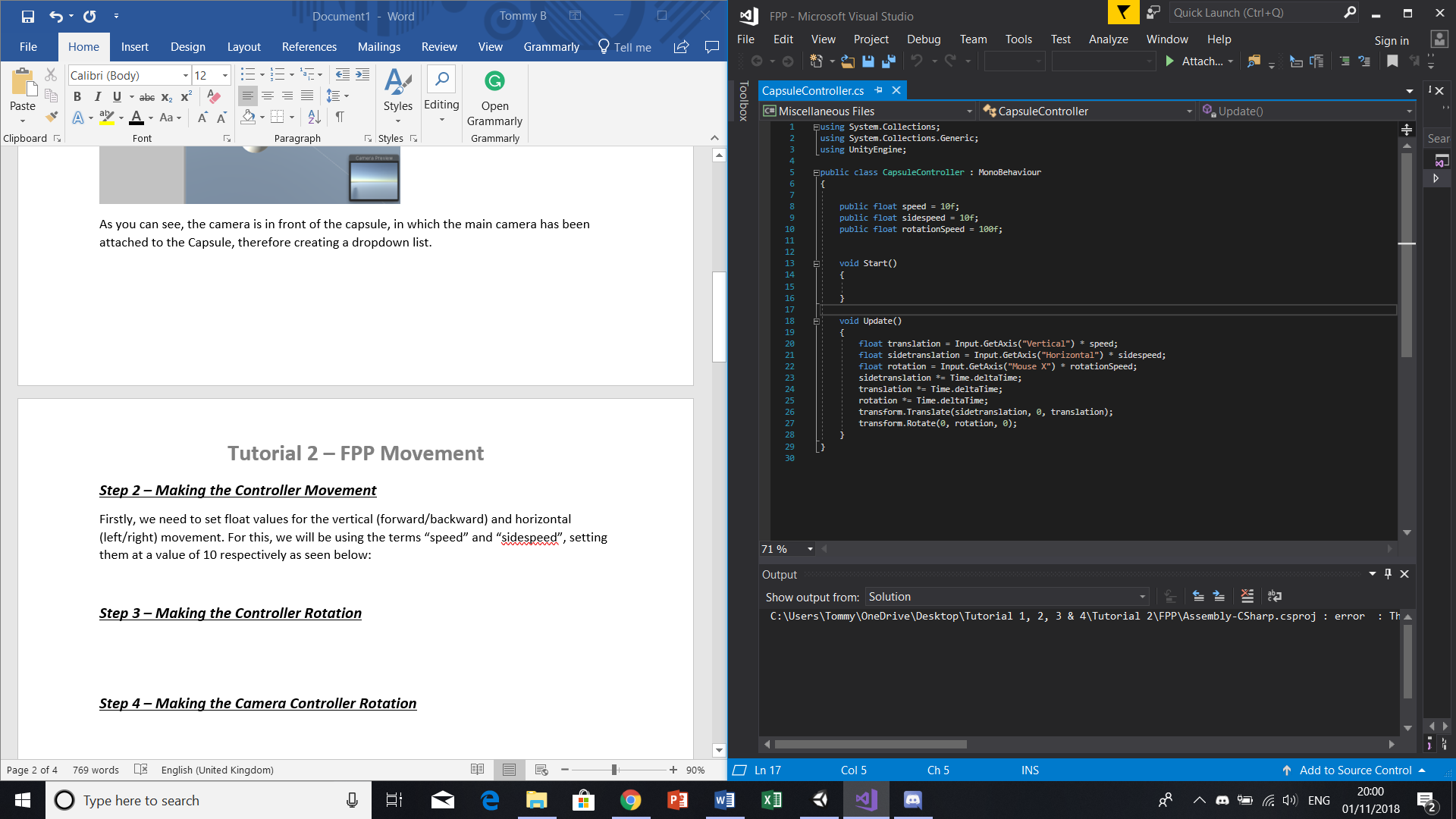
It is important that the capsule has a rigidbody, with the x and y constraints rotation constraint boxes tick to prevent the capsule from falling over. Another important thing that you **must** do is position the main camera in front of the capsule and drag-and-drop the camera onto the Capsule into the hierarchy. This will make the camera a “child” of the capsule, meaning wherever the capsule moves, the camera will go with it. It will look like the following:



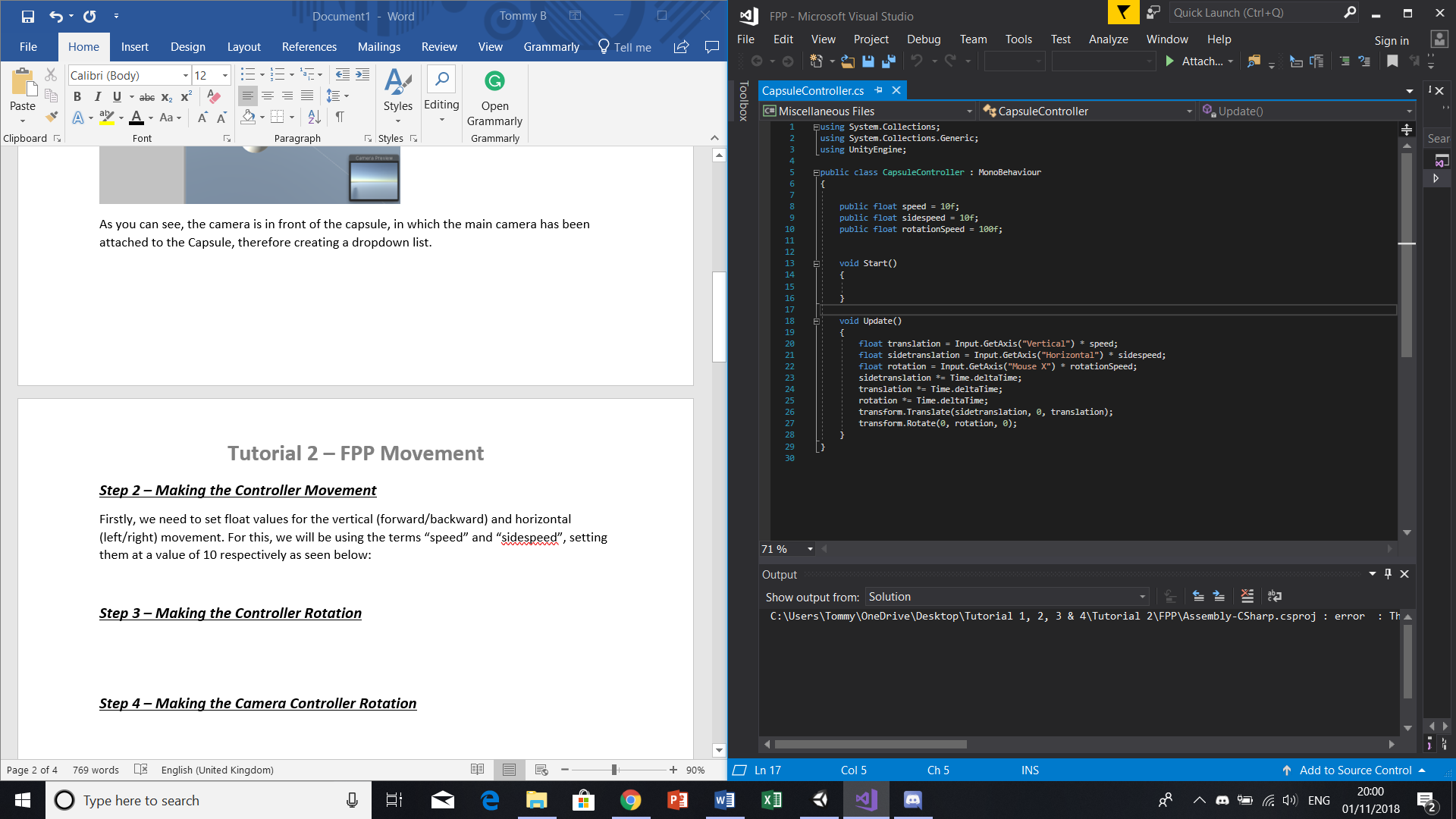
As you can see, the camera is in front of the capsule, in which the main camera has been attached to the Capsule, therefore creating a dropdown list.

**Step 2 – Making the Controller Movement**

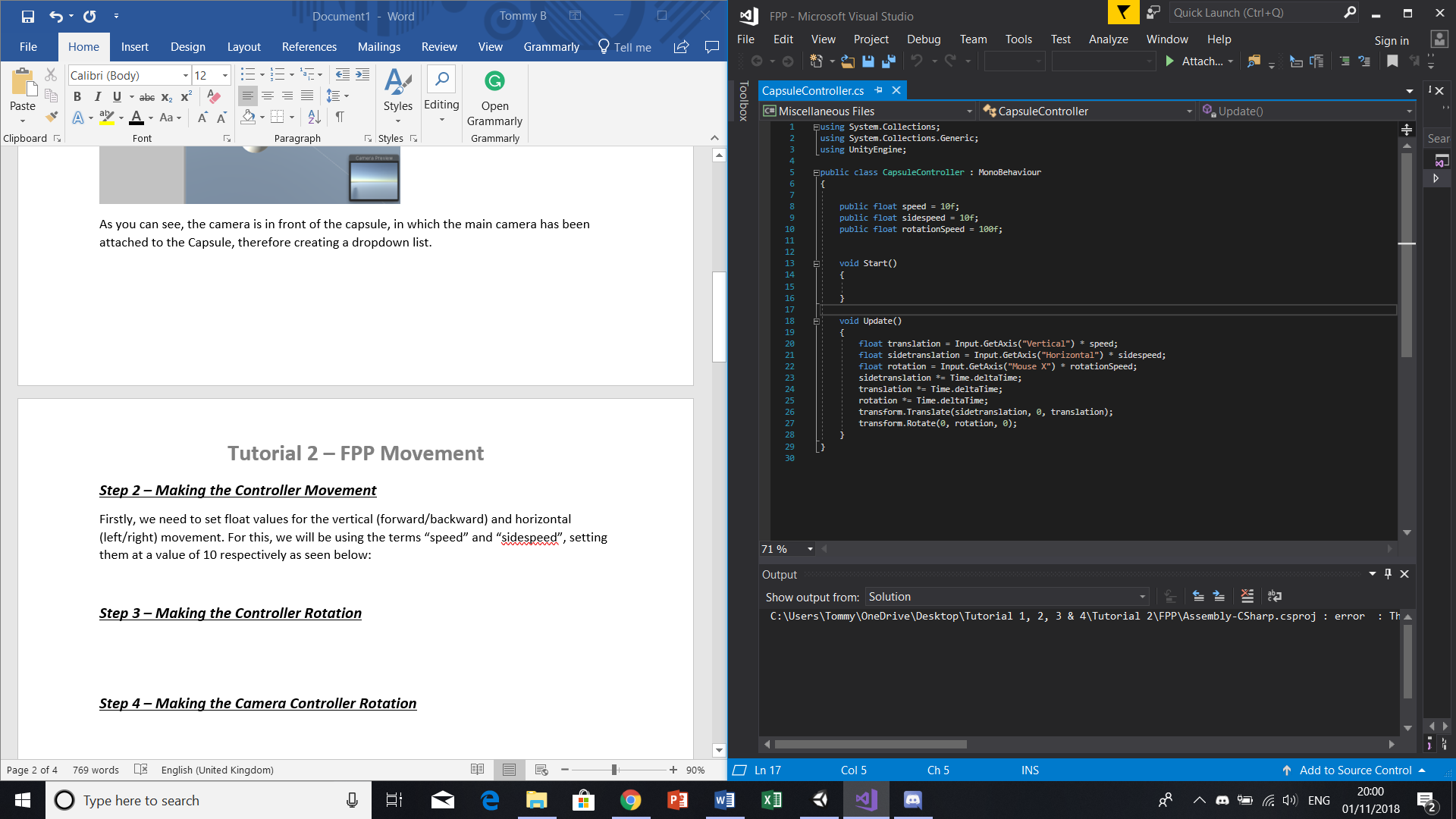
Firstly, we need to set float values for the vertical (forward/backward) and horizontal (left/right) movement. For this, we will be using the terms “speed” and “sidespeed”, setting them at a value of 10 respectively as seen below:



You will now need to refer to the axis and the axis orientation, of which you will apply the previous float values defined to these axes. It should look like this:

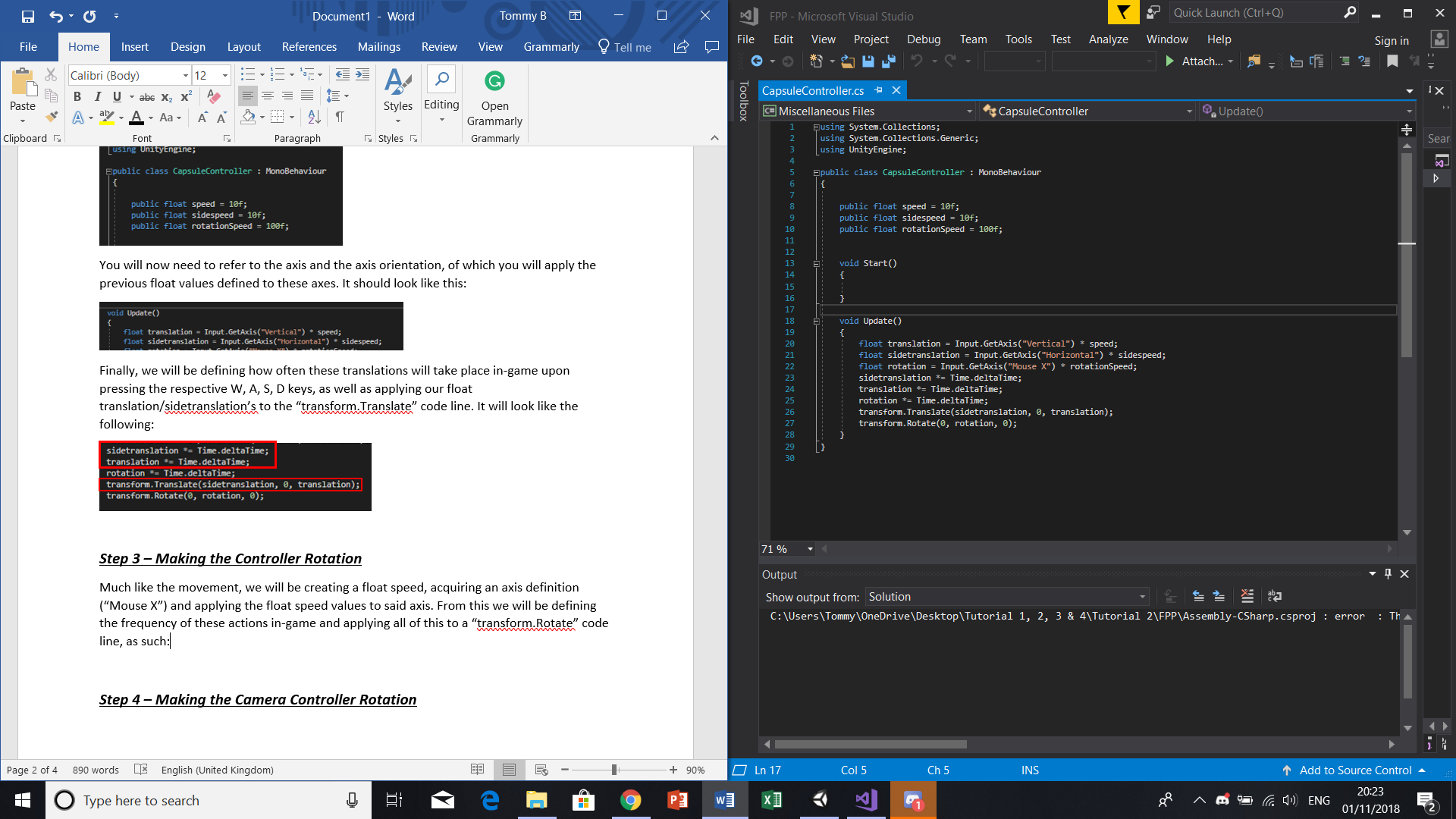


Finally, we will be defining how often these translations will take place in-game upon pressing the respective W, A, S, D keys, as well as applying our float translation/sidetranslation’s to the “transform.Translate” code line. It will look like the following:



**Step 3 – Making the Controller Rotation**

Much like the movement, we will be creating a float speed, acquiring an axis definition (“Mouse X”) and applying the float speed values to said axis. From this we will be defining the frequency of these actions in-game and applying all of this to a “transform.Rotate” code line, as such:



Once both the movement and rotation portions of the script are complete, drag-and-drop the script onto your **capsule** to apply the code. Once you press play you will notice that your character can move in any direction they choose, though you will **only** be able to look left and right. This will be amended in the next step of the tutorial.

**Step 4 – Making the Camera Controller Rotation**

This second script will simply set the Y rotation to the mouse, in which you will apply this script to the camera to allow for both an X and Y rotation in correspondence to your mouse movements. The only difference in the following script is the “\* -1” after the camera rotation float is assigned. This is to prevent the Y axis movement from being inverted (up = down and down = up). By putting it at a negative value, you flip the inversion to amend this issue. The script will look like this:

