Coding Journal

# 25th April

I am starting this module late because I have been stuck knee-deep in a lot of work from other modules. However, I am choosing to base my project on a first person platformer game. I have gotten started by creating a prototype level in Maya to base my game on.

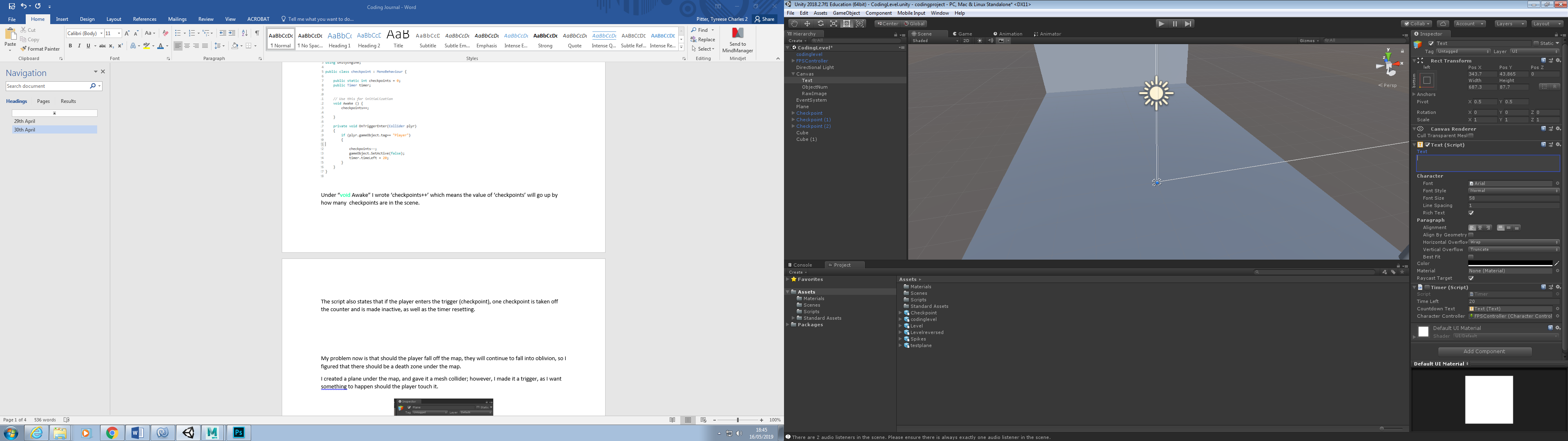
When I first imported it into unity, I ran into my first problem. My level was pitch-black and I could not texture it, however, I fixed it by simply remaking the map in Maya and re-exporting it to unity.

I then and added a mesh collider to the level added the FirstPersonCharacter asset **from a Unity made package** so that the player could walk and run around the level.

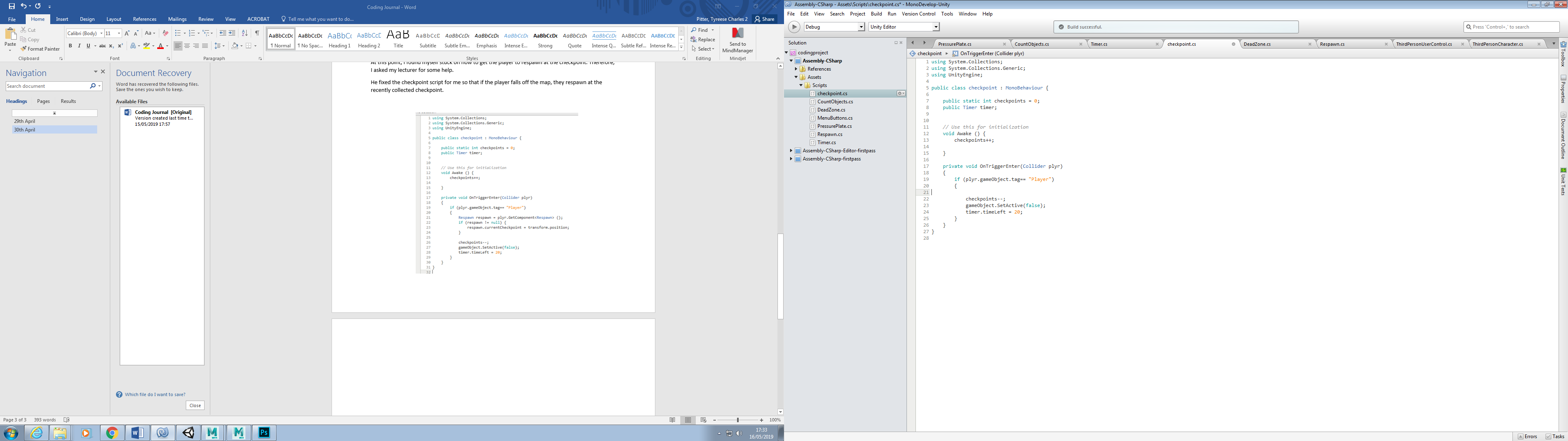
For my small game, the player will have to get past obstacles and reach a number of checkpoints before the time runs out. Therefore, I will need scripts for a working timer, collectable checkpoints that the player will go through, and a respawn should the player fall off the map.

# 26th April

I found a Timer Script from one of my old projects that I could use. So I created a UI Text and added the Timer script to it.



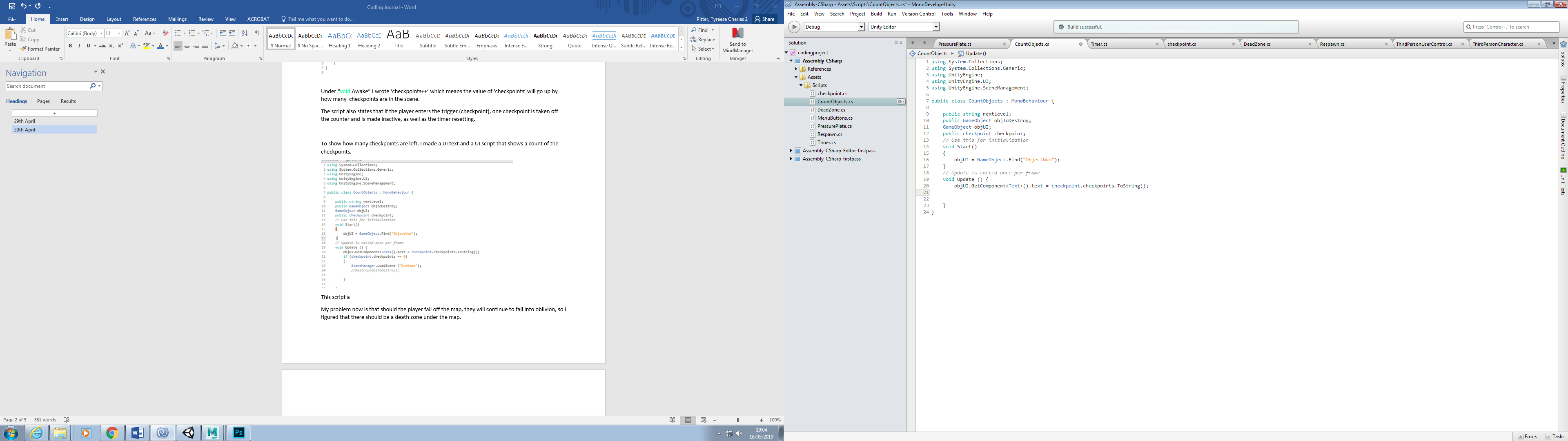
It works well, however it needs editing, as it needs to reset every time the player runs through a checkpoint, so I created a script so that checkpoint is collected (and shown on the screen) and the timer resets to the original time.



Under “void Awake” I wrote ‘checkpoints++’ which means the value of ‘checkpoints’ will go up by how many checkpoints are in the scene.

The script also states that if the player enters the trigger (checkpoint), one checkpoint is taken off the counter and is made inactive, as well as the timer resetting.

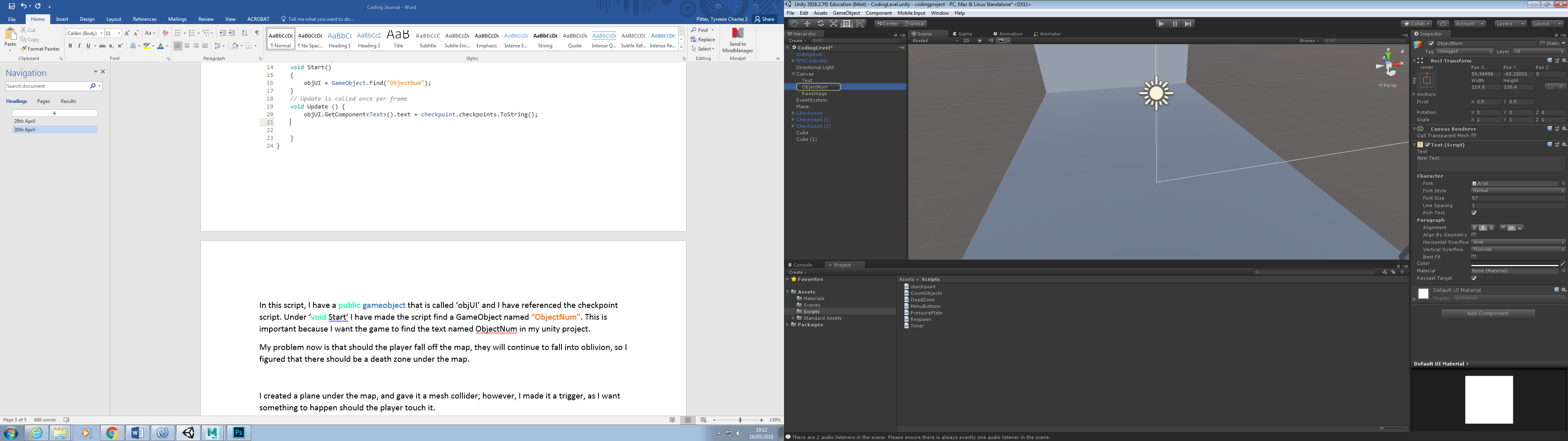
To show how many checkpoints are left, I made a UI “Event System” and a UI script that shows a count of the checkpoints, after I added this script to the Event S ystem.



**IMPORTANT!**

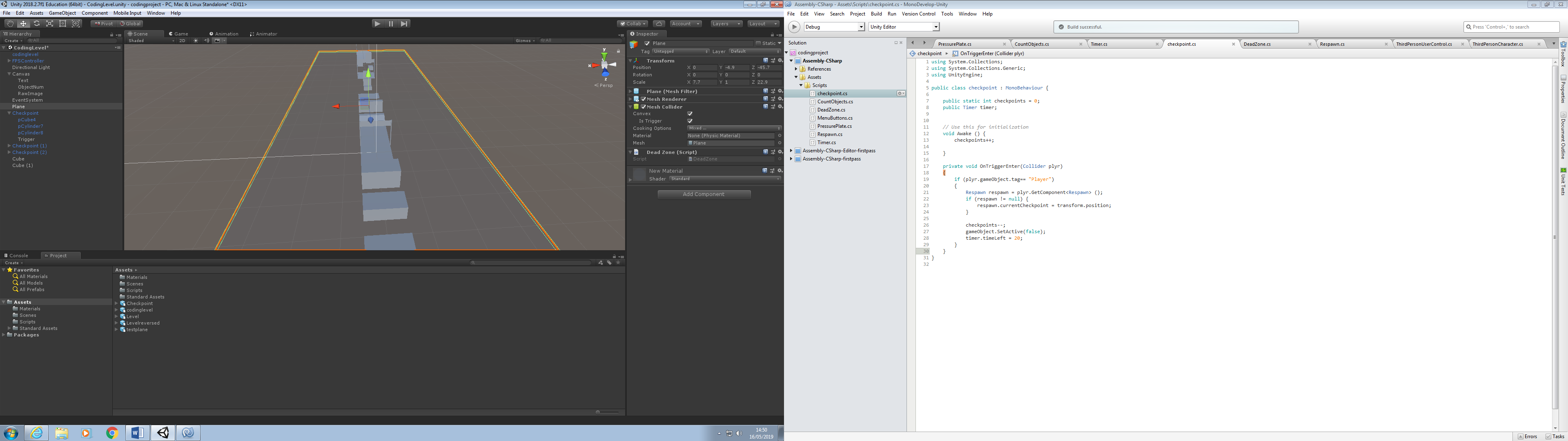
In this script, I have a public gameobject that is called ‘objUI’ and I have referenced the checkpoint script. Under ‘void Start ()’ I have made the script find a GameObject named “ObjectNum”. This is important because I want the game to find the something named ObjectNum in my Scene.

Under ‘void Update ()’ the objUI text will equal to how may checkpoints there are in the scene. It does this by going into the ‘checkpoint’ Script, finding the value of ‘checkpoints’, and displaying the value in the ObjectNum text.

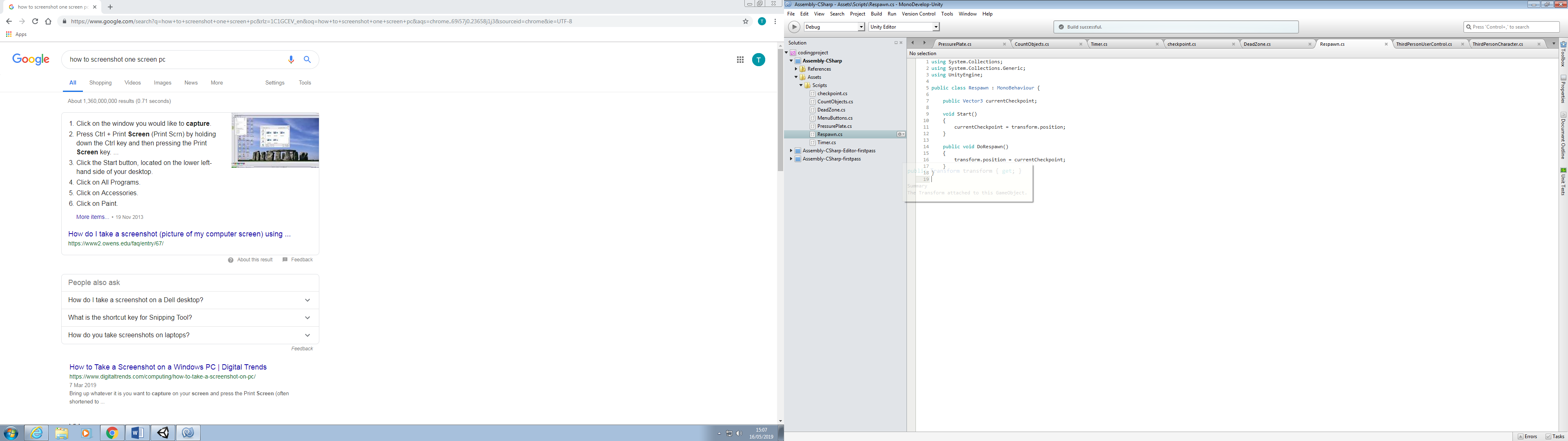


# 29th April

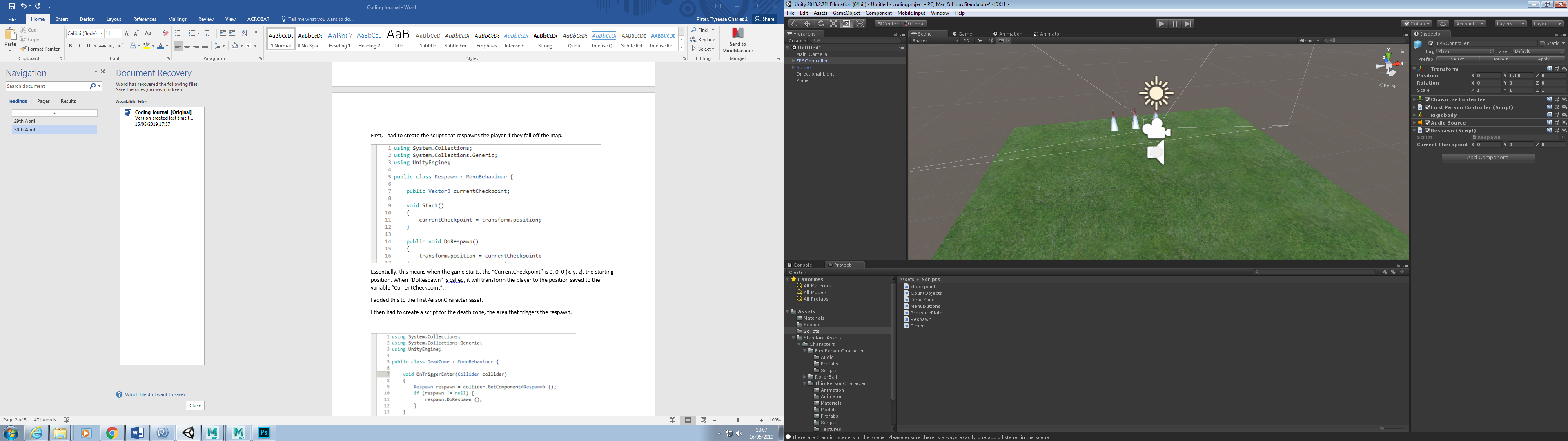
My problem now is that should the player fall off the map, they will continue to fall into oblivion, so I figured that there should be a death zone under the map.

I created a plane under the map, and gave it a mesh collider; however, I made it a trigger, as I want something to happen should the player touch it.

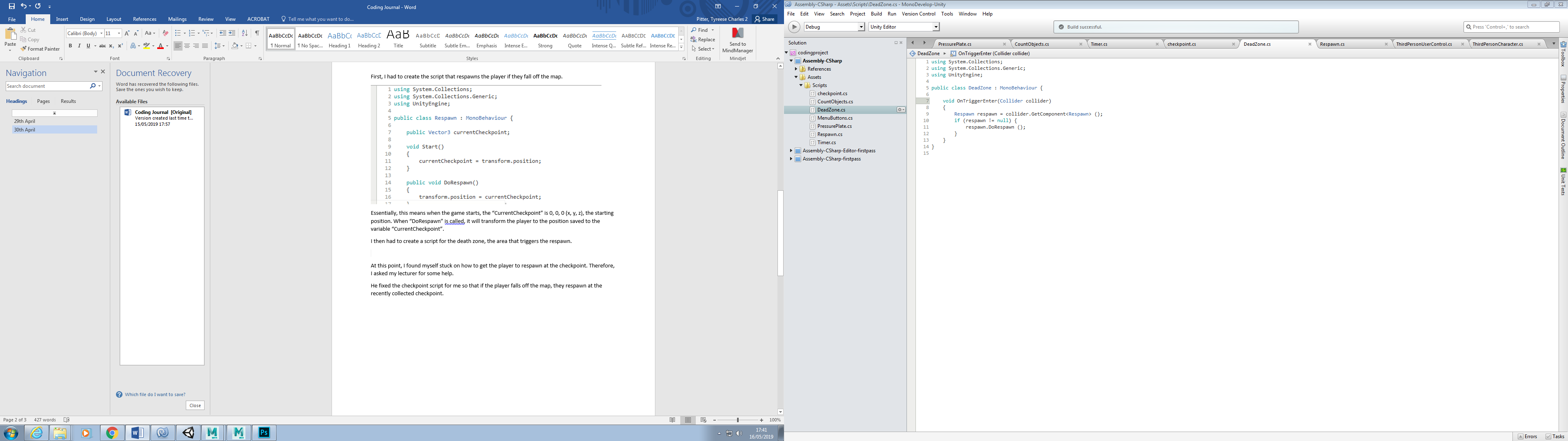
First, I had to create the script that respawns the player if they fall off the map.



Essentially, this means when the game starts, the “CurrentCheckpoint” is 0, 0, 0 (x, y, z), the starting position. When “DoRespawn” is called, it will transform the player to the position saved to the variable “CurrentCheckpoint”.

I added this to the FirstPersonCharacter asset.

I then had to create a script for the death zone, the area that triggers the respawn.

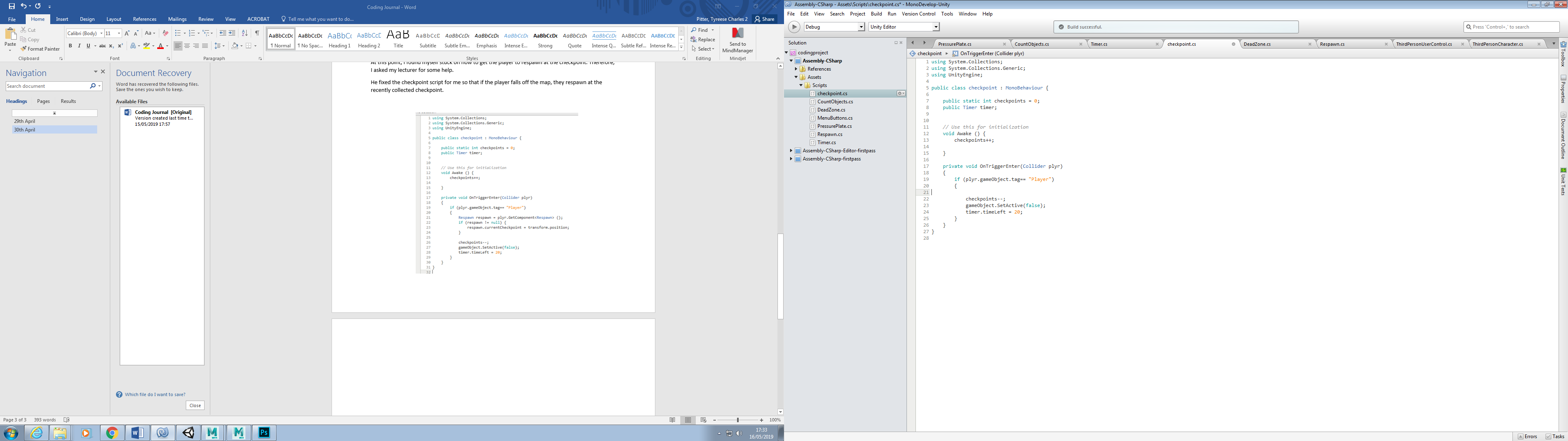
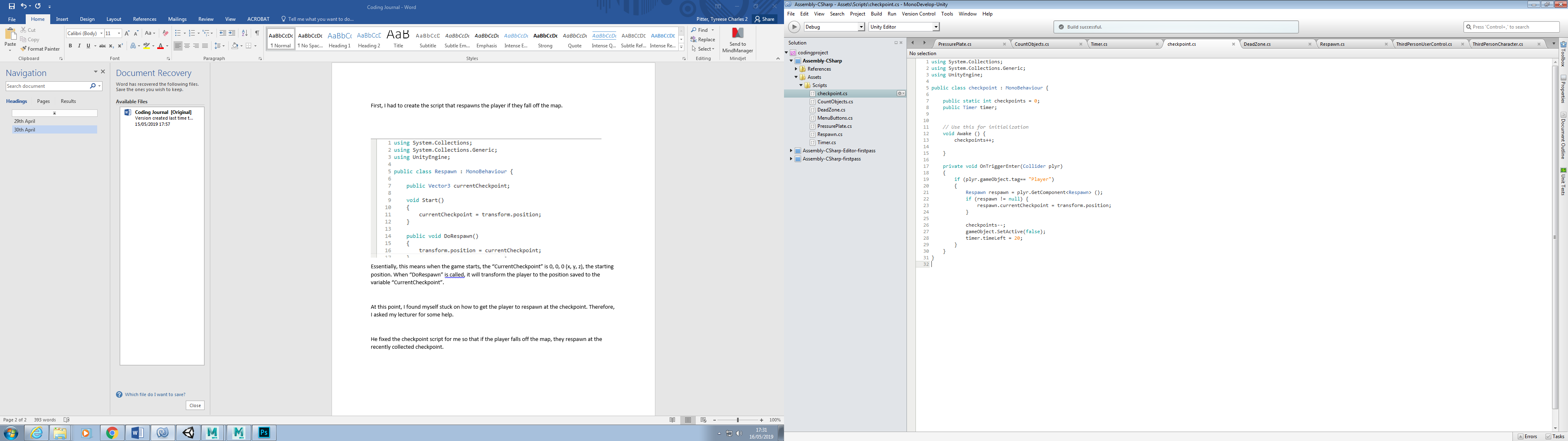


This script states that if something enters the trigger, the “Respawn” component is called, and if the variable “respawn” has no value, it will call the “DoRespawn” function.

# 30th April

At this point, I found myself stuck on how to get the player to respawn at the checkpoint. Therefore, I asked my lecturer for some help.

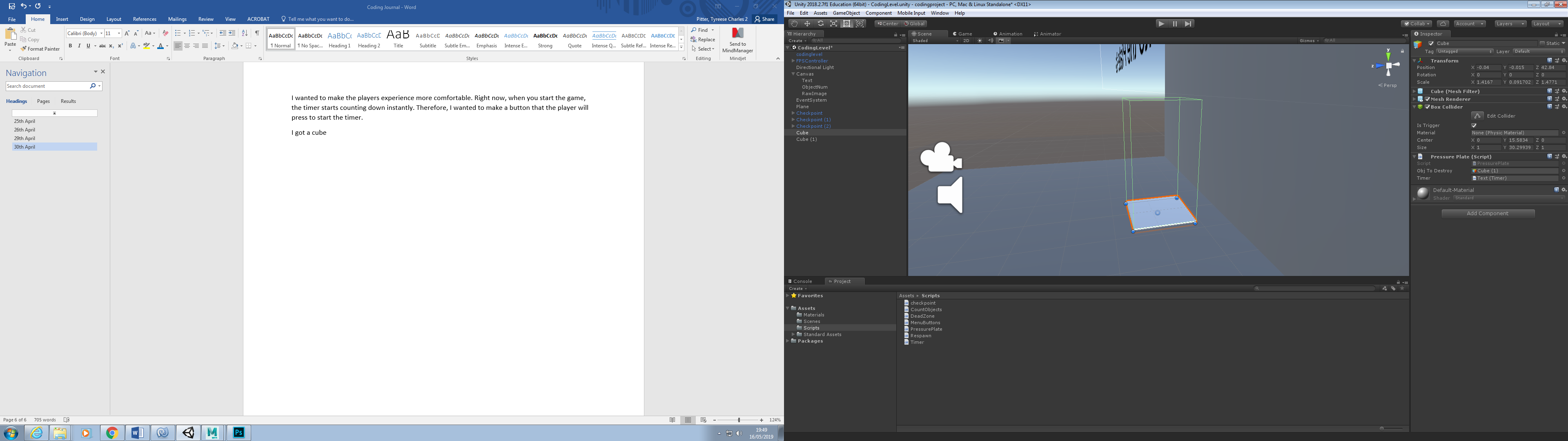
He fixed the checkpoint script for me so that if the player falls off the map, they respawn at the recently collected checkpoint.

Heres the before and after.

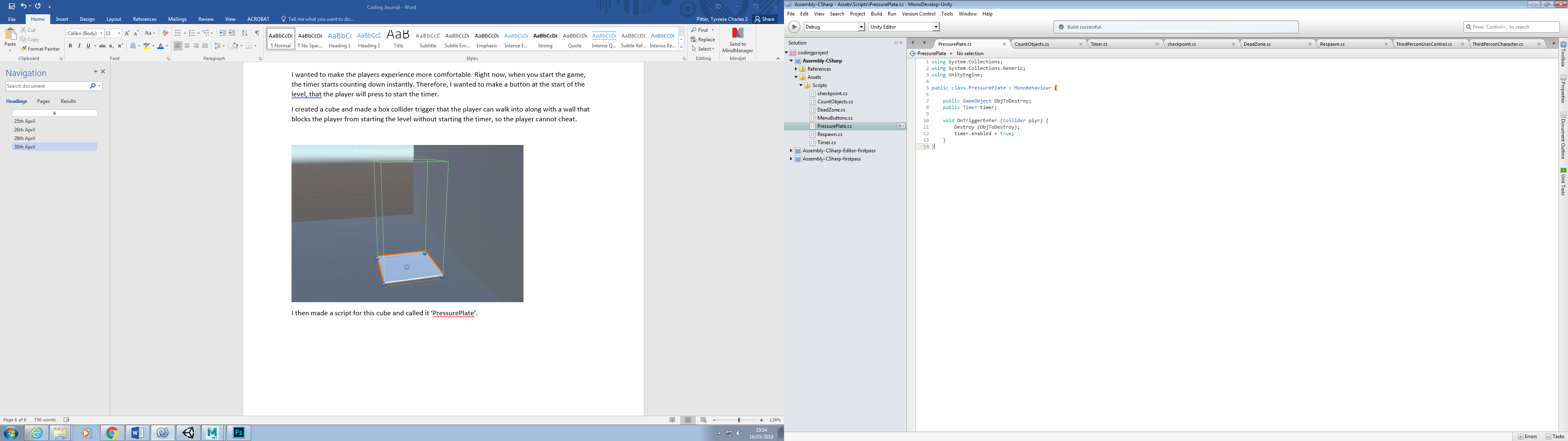
What he added (highlighted blue) states that, should the player touch the checkpoint, the “CurrentCheckpoint” is set to the checkpoints position.

I wanted to make the players experience more comfortable. Right now, when you start the game, the timer starts counting down instantly. Therefore, I wanted to make a button at the start of the level, that the player will press to start the timer.

I created a cube and made a box collider trigger that the player can walk into along with a wall that blocks the player from starting the level without starting the timer, so the player cannot cheat.



I then made a script for this cube and called it ‘PressurePlate’.

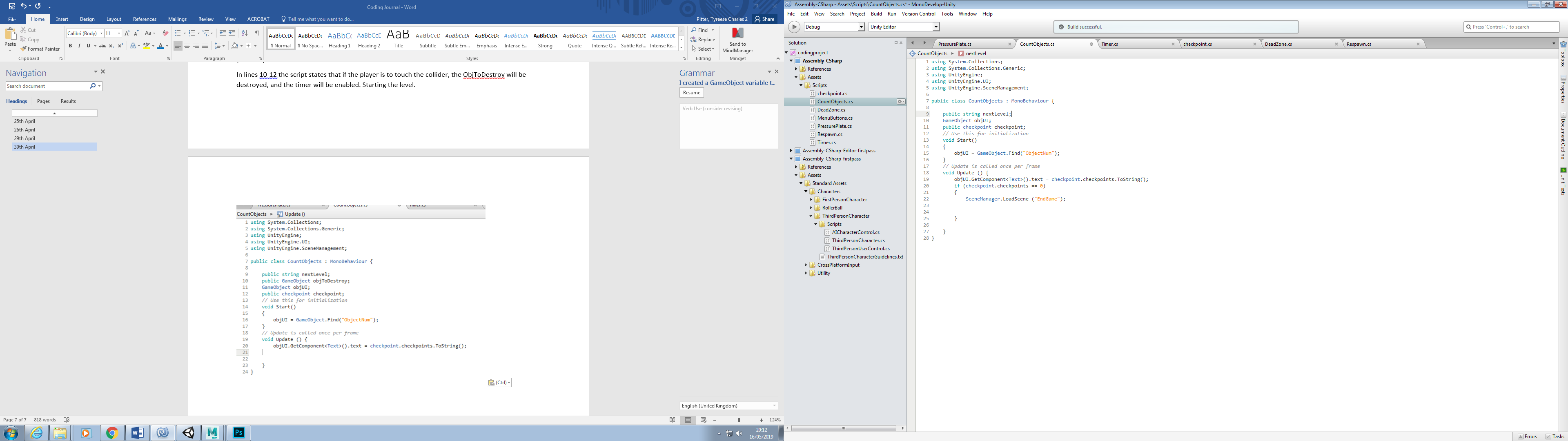


This script is simple; I create a GameObject variable that refers to an object that will be destroyed (the wall) and I create a variable that refers to the timer in the scene.

In lines 10-12 the script states that if the player is to touch the collider, the ObjToDestroy will be destroyed, and the timer will be enabled. Starting the level.

# 1st May

I wanted to make it so, should the player get to all checkpoints, the game ends.

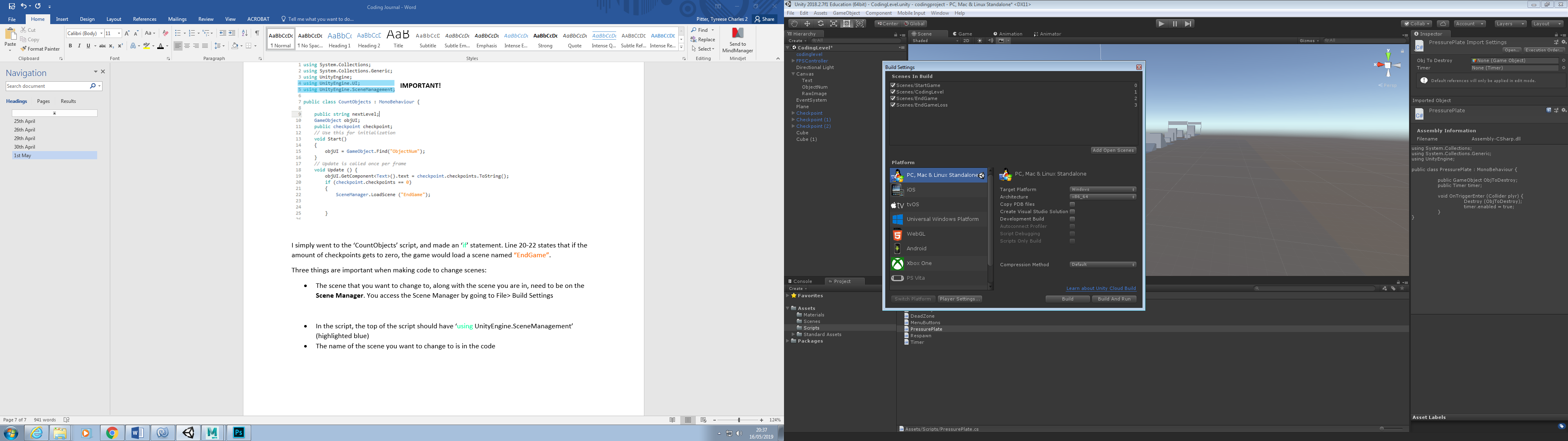


**IMPORTANT!**

I simply went to the ‘CountObjects’ script, and made an ‘if’ statement. Line 20-22 states that if the amount of checkpoints gets to zero, the game would load a scene named “EndGame”.

Three things are important when making code to change scenes:

* The scene that you want to change to, along with the scene you are in, need to be on the **Scene Manager**. You access the Scene Manager by going to File> Build Settings



* In the script, the top of the script should have ‘using UnityEngine.SceneManagement’ (highlighted blue)
* The EXACT name of the scene you want to change to is in the code. E.g. “EndGame” is an actual scene name.