Semra Usanmaz

Learning Journal

<u>Tuesday 8th October 2019 – Inventory – The Scrapped Tutorial</u>

The first tutorial I worked on was NOT a tutorial which I used for my final four – it was an inventory tutorial **(this tutorial is in the Google Drive link)**. The reason as to why I am talking about this, however, is because I learnt quite a lot about Unity from doing this tutorial that I did not know before, even though I did not manage to finish it. I used mainly Youtube to search for guides on how to create an inventory and came across a Youtube channel called "Sykoo" with a video titled "Inventory Tutorial in Unity 2019!" which helped me with a lot of the script. The basic scene and canvas adding and enabling was fine, but the code itself was a struggle. One of the main things I learnt whilst creating this tutorial is that you can change the "MonoBehaviour" part of a Unity script to "ScriptableObject", this is not something I knew you could do before and it is used so that you can derive the script from a scriptable object rather than have to place it on a game object. I also learnt about 'Grid Layout Group' and how useful and quick it is, the 'Grid Layout Group' component places objects equally spaced from each other – very helpful when creating an inventory. I also learnt that by writing [CreateAssetMenu] below the "Using UnityEngine;" section, you can right click in the project area and create the item in the script, which is again very useful.

The part of the tutorial where I got stuck on was the part with adding items into the inventory. I was able to create 'if' functions and bools related to the addition/removal of items, but I was not able to get it to work properly.

In conclusion, although the inventory tutorial did not work out for me, I learnt a lot about Unity from attempting to create it.

<u>Thursday 17th October 2019 – Countdown Timer – Tutorial 1</u>

I started to create my countdown timer tutorial, I knew the basic canvas and text creation from previous knowledge of Unity and so I was able to do that myself to prepare the canvas. The first main thing I learnt whilst creating this tutorial was that in Unity, 'Time.deltaTime' is the time passed since the last frame so it will be the same time for everyone regardless of computer speed. Therefore, whenever you do something related to time in Unity, you should multiply it by 'Time.deltaTime'. To find help on how to make this tutorial, I mainly searched Youtube and came across a video by a channel called "Single Sapling Games", called "Countdown Timer in Unity – Easy Beginners Tutorial/Guide" which provided me with a clear explanation of how to create a countdown timer. Rather than watching the entire video, I attempted to do some of the code myself and view parts of the video where I was completely stuck. For example, one part I was stuck on was trying to figure out how to stop the timer from going into negative numbers and just staying at 0, the video gave me the hint that to stop this, I will need to put an 'if' function around the original code stating that if the timer is above 0, it counts down, therefore if the timer reaches 0, or is below 0, it stops counting down and stays at 0.

Semra Usanmaz

<u>Thursday 24th October 2019 – Dialogue System – Tutorial 2</u>

Today I worked on a dialogue system tutorial. This tutorial was much harder to do than the countdown timer one, but easier than the (attempted) inventory one. Just as with the timer tutorial, I already knew the basic canvas and text creation from previous knowledge of Unity and so I was able to do that myself to prepare the canvas. I also knew about arrays within the script. I looked up a guide on how to do the harder parts of this tutorial via Youtube and came across a channel called "BlackThornProd" with a video called "Cool Dialog System – Easy Unity and C# tutorial". The main thing which I learnt whilst creating this tutorial was how to create a 'typing' effect with dialogue, which I did not know how to do before. To do it, you must create an IEnumerator/Coroutine function with a foreach loop within it which references the letters within an index in the array of sentences. Then within the foreach loop, make it so that one letter is added every time the function runs. After that add a "WaitForSeconds" function which will be equal to the typing speed you want for your game. When you start the coroutine within the Start() function, the text should be 'typed' out once you hit play.

<u>Thursday 31st October 2019 – 2D Movement and Jumping – Tutorial 3</u>

The third tutorial I worked on was a 2D movement and jumping tutorial. I was able to do most of the easier stuff such as setting up the canvas, rigidbody components, and colliders by myself, but when it came to the actual movement code, I did not know exactly what to do. I looked up videos on Youtube again for assistance and came across a channel called "Xlaugts" which had a simple 2D movement tutorial video called "Unity Movement Tutorial". The first code I learnt about was the "Input.GetAxis()" function which is a much easier alternative to typing out each key on the keyboard within a normal input function as if you put "Horizontal" within the brackets, the system automatically knows that you mean the left/right arrows or a/d. I also found out that a rigidbody has a velocity already attached to it and you can alter that by typing "rb.velocity = …" and that is how you make it so that when you press a specific key, the velocity of the object changes and it moves. The final thing that I learnt about which I wasn't too knowledgeable on before was that an "OnCollisionExit" function exists alongside "OnCollisionEnter" which means that the system can check whether something leaves collider boundaries as well as entering them, which is very useful.

<u>Thursday 14th November 2019 – Picking Up Objects and Gaining Points from them – Tutorial 4</u>

The final tutorial I worked on was a tutorial on picking up objects and gaining points to the overall score from them. I found the first part of the tutorial not too hard as I already knew about canvases and objects within the scene, as I have mentioned before. However, I did end up changing the code in the second part of the tutorial a few times as it used to be just for picking up one game object so when I tried to add a second object into the scene later on in the tutorial and played the scene, the first got removed but the second did not, I realised this was because I had to make the "GameObject pickUp" line into an array (GameObject[] pickUp) and change "FindGameObjectWithTag" to "FindGameObject**s**WithTag". One other thing I learnt whilst working on this tutorial was that when an object has "is trigger" enabled on it, "OnTriggerEnter" functions work with it. I tried to do this entire tutorial on my own, however there were specific parts I had to look up online to see how to do them correctly, for example, I looked up what exactly you can do with "is trigger" enabled on the actual Unity website as well as on "answers.unity.com".

Semra Usanmaz

<u>Monday 25th November 2019 – The Component Project</u>

For the component project, I mixed together all four of my tutorial components and added a few extra tweaks to make the component an actual, mini, playable game. In the game, you play as an astronaut in training who must collect space rocks within a time limit. Most of my tweaks were in the "Dialogue" script as I had to make sure the correct dialogue comes up at the correct times, for example, I wanted the dialogue to stop after the first three sentences and continue once you got every pickup on the screen. I did struggle a little with this as a blank dialogue panel was showing up instead of what I actually wanted to happen but I managed to figure it out in the end by making it so that if both the index is more than a certain number and the score is less than a certain number, the dialogue box is disabled and if the score hits a certain number, it is enabled.

I also added a reset function to the game which allows the player to restart the level if they were unable to get every space rock in time. This wasn't too hard to add as all I had to do was ensure that I added "UnityEngine.SceneManagement;" into the script, (this enables it to work with scenes), and then make it so when you press a button, the scene is reloaded.

Furthermore, I came across another issue within the component that was making my player character get stuck on the edges of platforms, I figured out that to fix this issue, I needed to add a physics material to it with 0 friction.

The final extra thing I added to make the component work well enough, was a fail state in which when the timer reaches 0 and you have not collected all the space rocks, a text box comes up saying "You have failed, press 'R' to restart" and the player movement controls are disabled.

I did not have to look up how to do these online as I was able to fix most of it via trial and error.