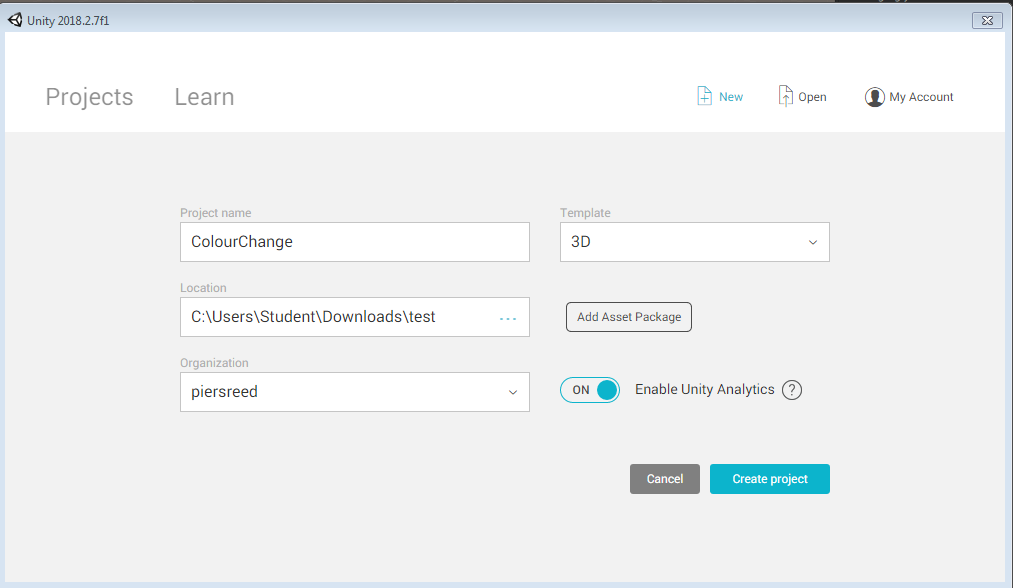
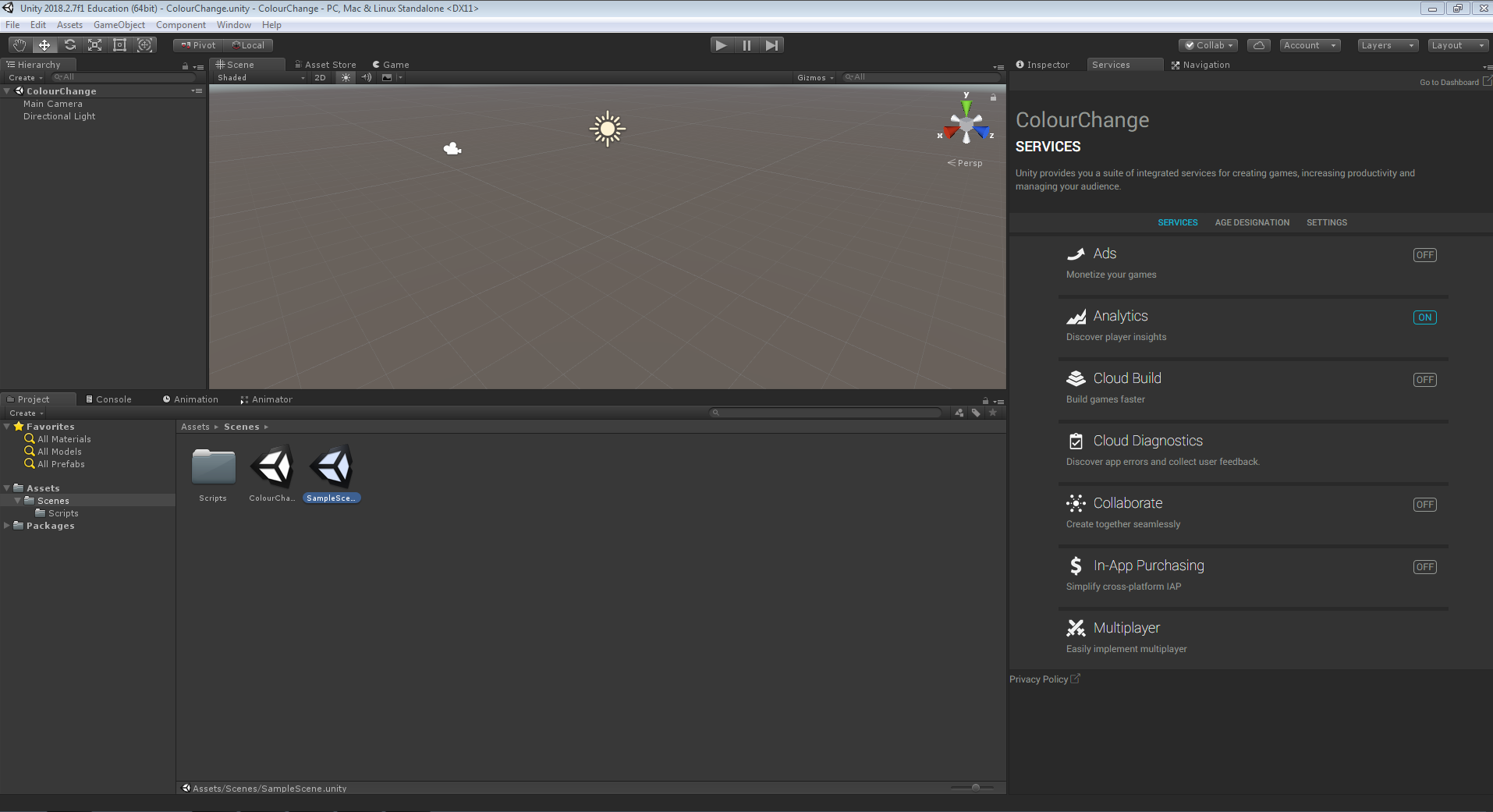
This is a tutorial that I have made in order to teach someone how you can change colour upon script. This is a tutorial meant for beginners as I am not the most adept when it comes to using Unity.

So today we will be discussing on how you would change the colour of a game object within Unity 5 with a script. Now I know that there is a simpler way into making an object by making a different material and applying it but I thought that it could be made a bit more interesting.

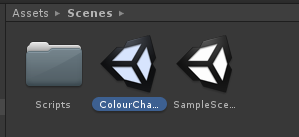
So let’s get started:

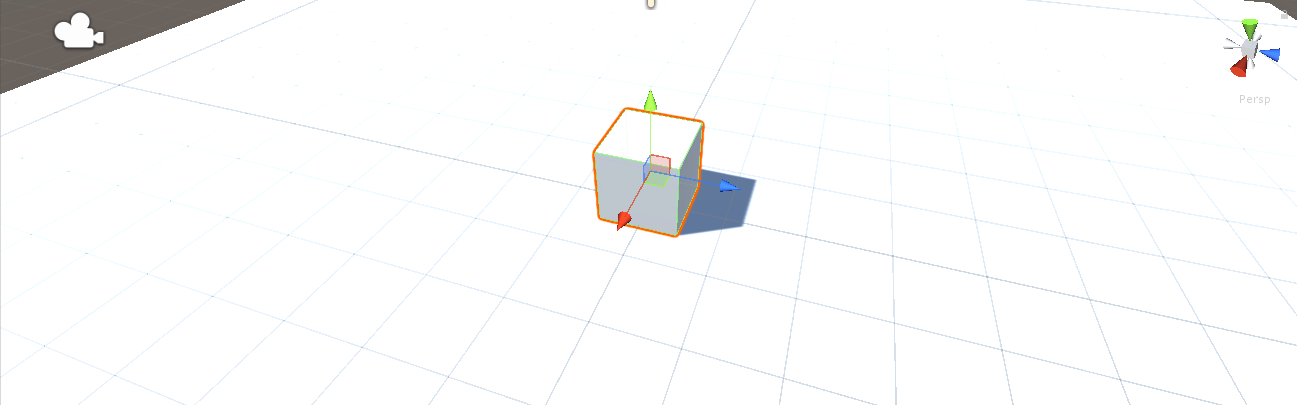
Firstly, you want to make a new Unity project.

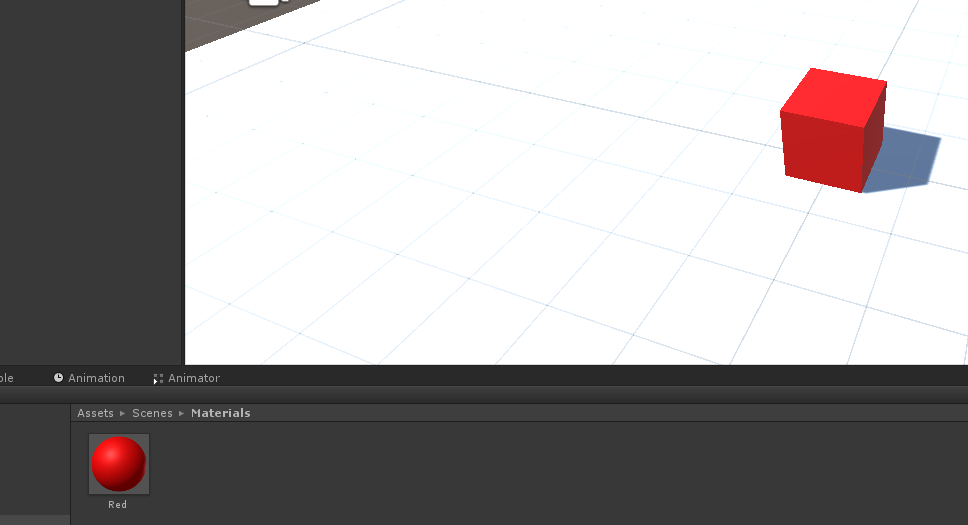
You can name it what you want as it the name is really never important; the only importance is that the name is relevant to you. As such I am going to name it **“ColourChange”**. Make sure it is in the 3-D template and you are saving it in a place you know, as for the organization, all that is about is the name you assigned your company when getting Unity, it asks you to make an organization name.

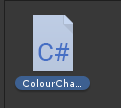
After that you will open to a blank screen called **“SampleScene”.**

I usually don’t edit this scene and create another as I will always use the **“SampleScene”** to leave my project before closing it as it will reduce loading time if you load in a blank scene first rather than a fully loaded scene as you can then load any scene you want rather than the last one you were on before closing, which you may not even want to use/edit on that day.

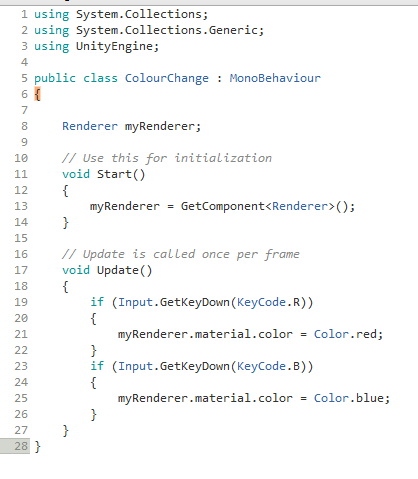
For this example, I will create a separate scene called “**ColourChange”** I will then make a new folder called “**Scripts**”. Again you don’t have to do any of this part as it is just me being a bit organised you can literally do what you want for organising your work. To make a new folder you right click > Create > Folder.

Now let’s create a basic layout for our scene so we can test our script and materials. Just create a simple 3-D Plane and a Cube, of which we will be changing the colour of. To create these just go into “**GameObject**” > 3D Object > Cube/Plane.

Now as you can see with a ground and an object to test for our script, but first for the people who just want to know on how to simply add new colours without any codes. To do this let’s create another folder called “Materials”. In this file we are then going to Right Click > Create > Material, then all you have to do is rename it (by double clicking on game) and then apply it to your object. You can either do this by click and dragging the material onto your target or by clicking on the object and adding it through the inspector.

Now that we have that sorted let’s create the script that will enable us to change the colour at will. First let’s get back to the “**Scripts**” folder we created and make a “**C# Script**” To do this just Right Click > Create > C# Script. Now what you name it is important as once you have done and wish to change it later you will have change the name both in and out of the script itself. I have called mine “**ColourChange**”

Now let’s double click on it so a software called “**Assembly-CSharp**” with you will be presented with a blank script containing two voids, it also contains using multiple systems and if I explain them right now we will be here till I am 50 so we can leave that to a different day. The void called “void Start” is used for initialization, in short it is used once and for statements like y = 5f for an example. The “void Update” is used for once per frame and this means that it is constantly going.

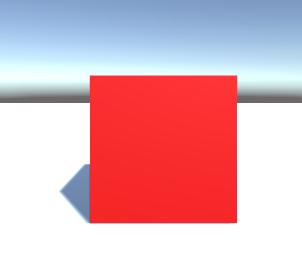
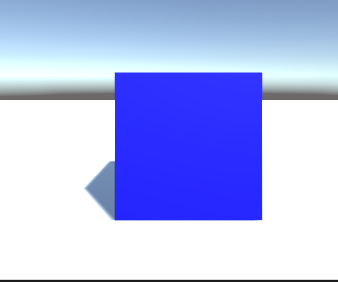
So add in the following context and I will then explain my script.

So this is the script that you will be needing now I used a video to find this out, which I can’t seem to find again, I then had someone’s help to make it work for Unity 5. So the “Renderer myRenderer;” stands for us references the Renderer later in the script as an object. You then have hat we have put in the void Start function. “myRenderer = GetComponent<Renderer>();” is us asking the script to get the renderer for our game object which we have already called myRenderer at the top of the script. Then for what is going on in the void Update is that we have put in some if statements. An if statement is use creating a what if statement in a way. We have in short by using “(Input.GetKeyDown(KeyCode.R)

{

myRenderer.Material.color.red;

}”

By doing this within an if Statement we have asked it to use a key code that we have chosen to activate a certain action in our case we have used “r”. We have also referenced our object under the if statement. We have essentially asked our script to when the button R is pressed on the keyboard to make our object Red. We can do other colours by making slight adjustments such as Blue.