Unity Tutorial 1: Designing a main menu for your game

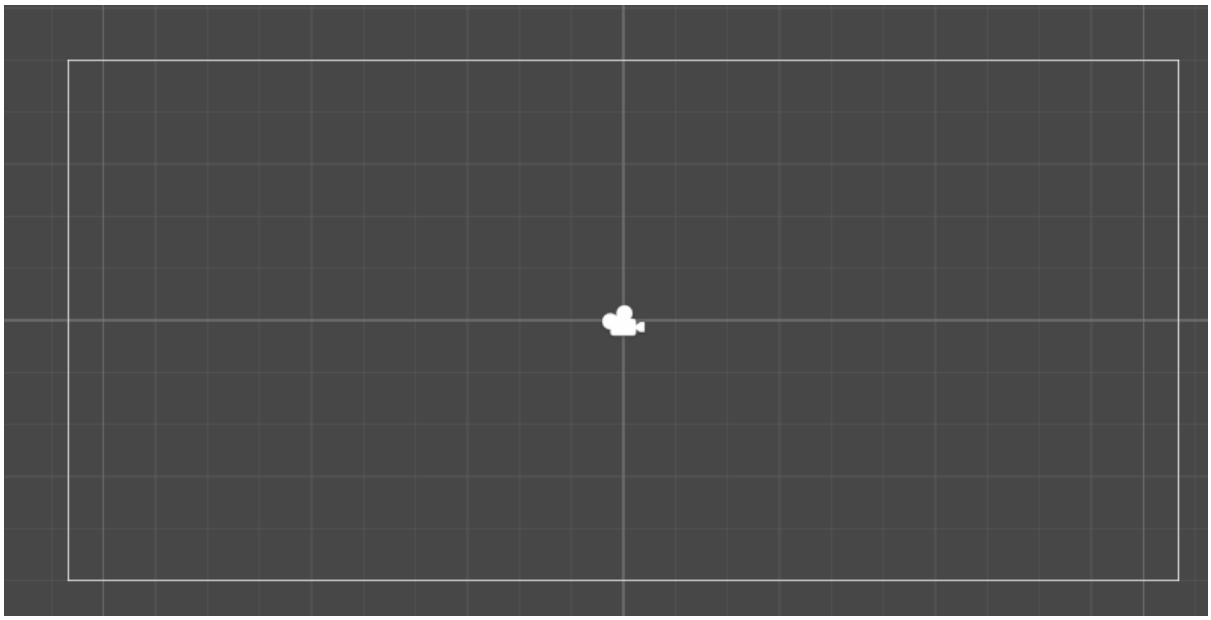
What is a Main menu??

So far in almost every game you play has one element in common. A main menu. A main menu is one of the most important parts of every game as it allows you, the player an option to play the game, whether it is starting a new save or playing on your recent save. A main menu also allows you to access the options for your game where you can modify your preferred choice of graphics and display according to the performance of your device.

Let's get started with your first main menu for your game.

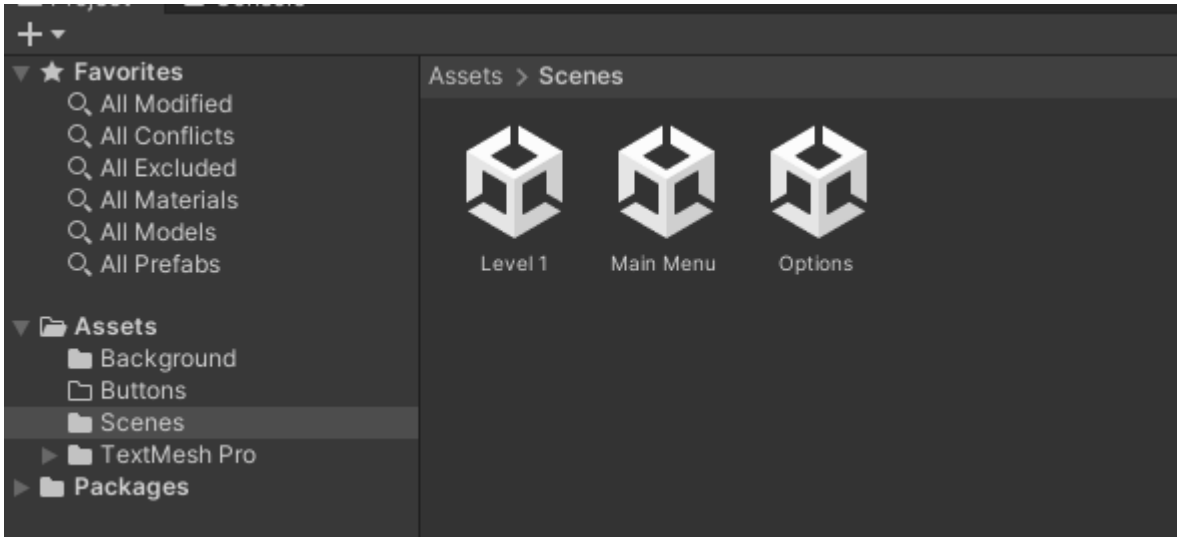
1. Create your project!!!

When you first create your project you will be met with a camera. This is what you will see when you first play your game.



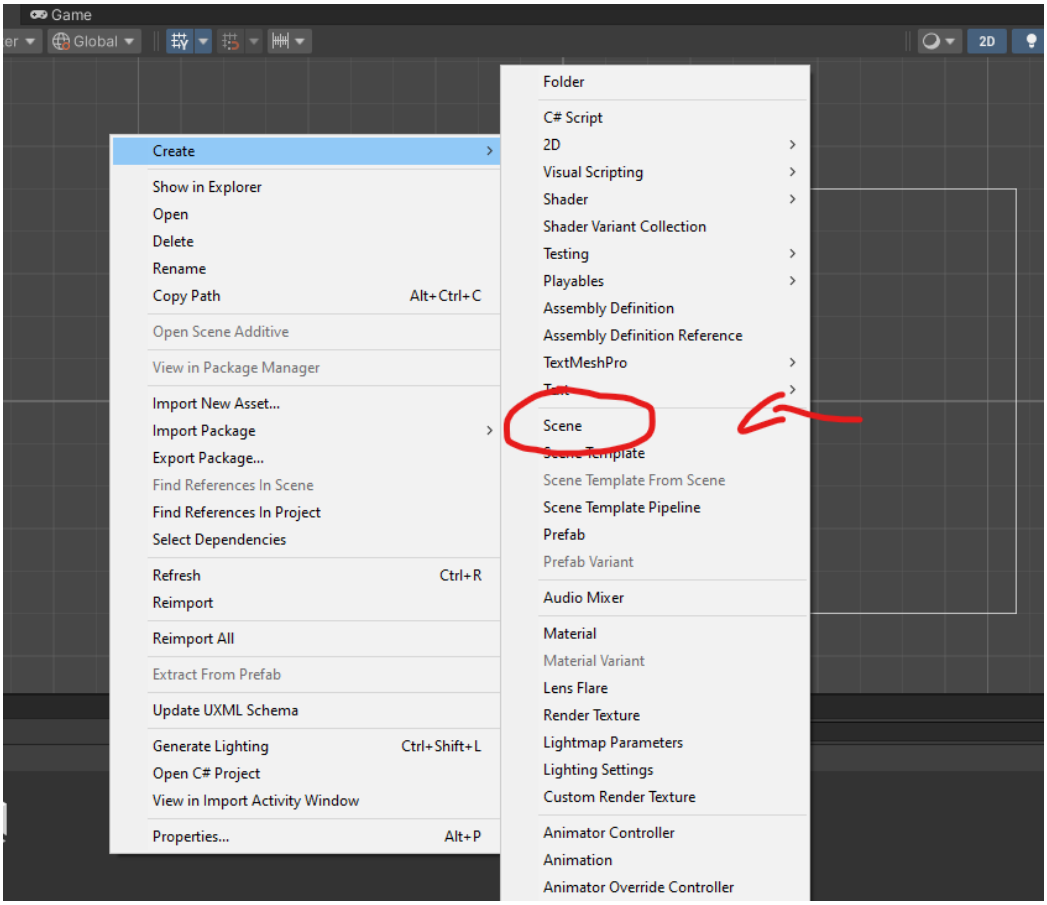
This is the default scene you get when you first create a unity project.

2. Add scenes



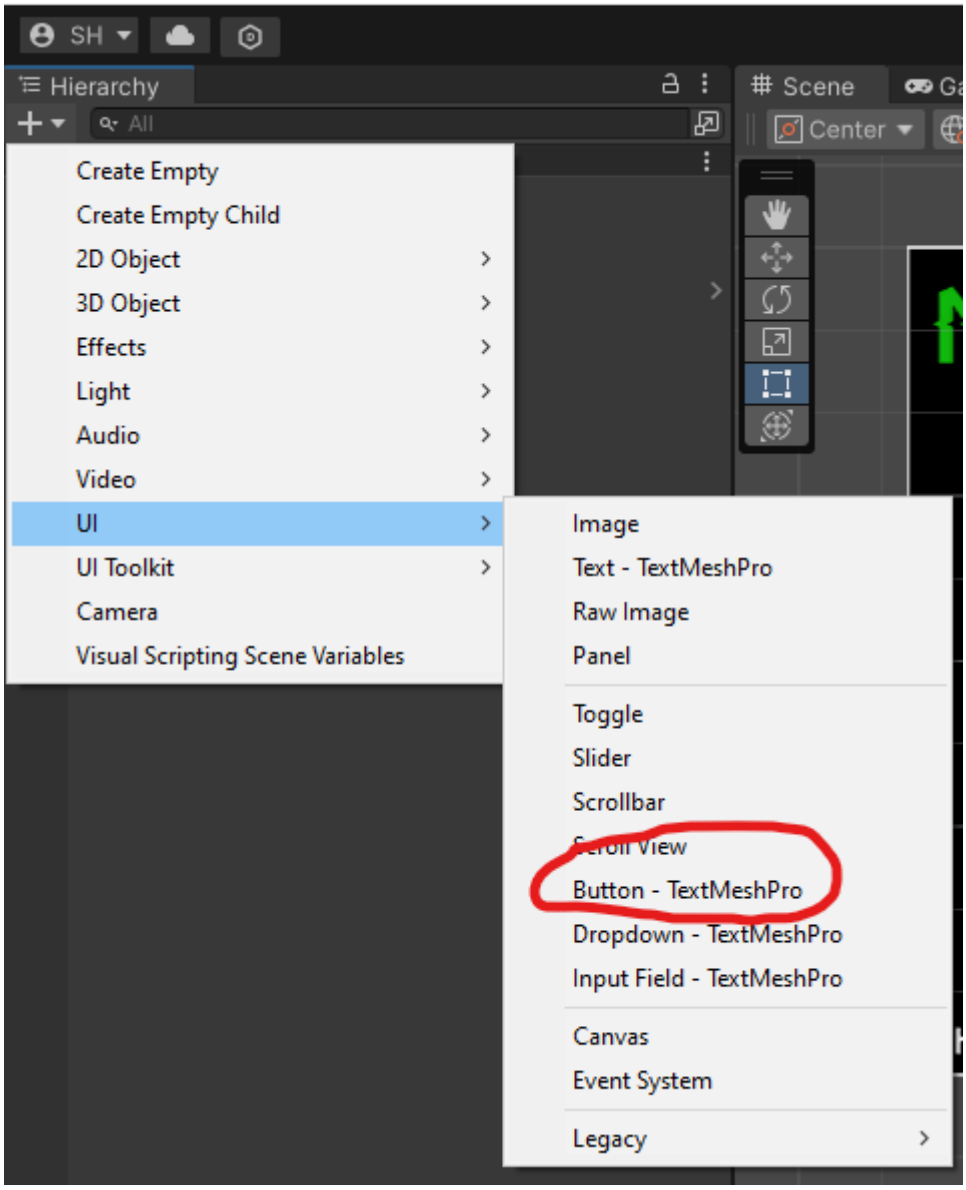
Now let's add scenes to our project. We use scenes to allow the players to navigate their way around the game from level to level (well some games do).

To create a scene you right click on your asset manager and create a scene. Using the image above as a reference, remember to name your scenes. Nobody likes their projects unorganised!

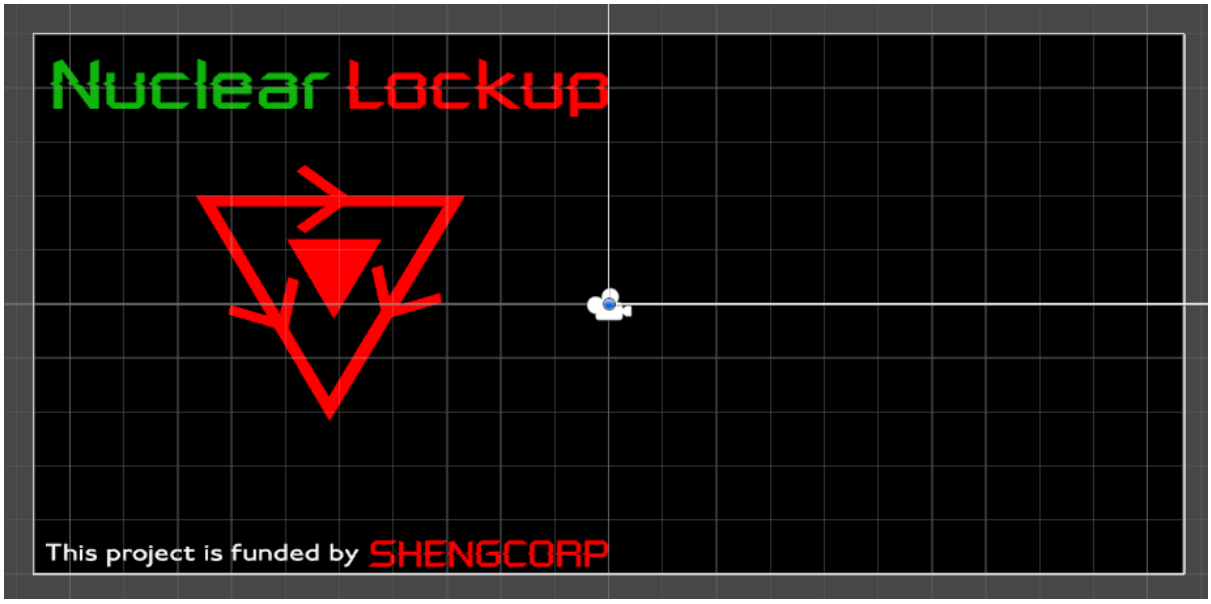


3. Adding buttons

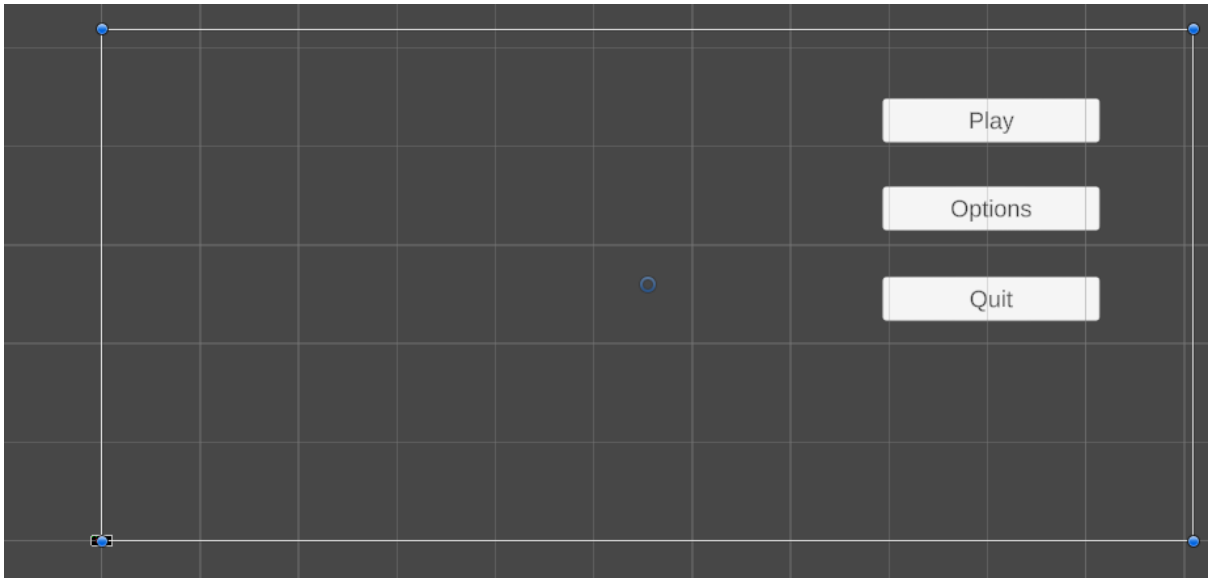
To add a button to your scene you click on the plus button under the Hierarchy, then go to the UI and click on the “Button – TextMeshPro” to add a button. This will automatically create a canvas where you can organise the layout of your main menu, specifically moving the buttons around.



This example image below is the camera view.

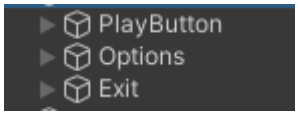


This is the canvas view (for organising the button and UI layout)



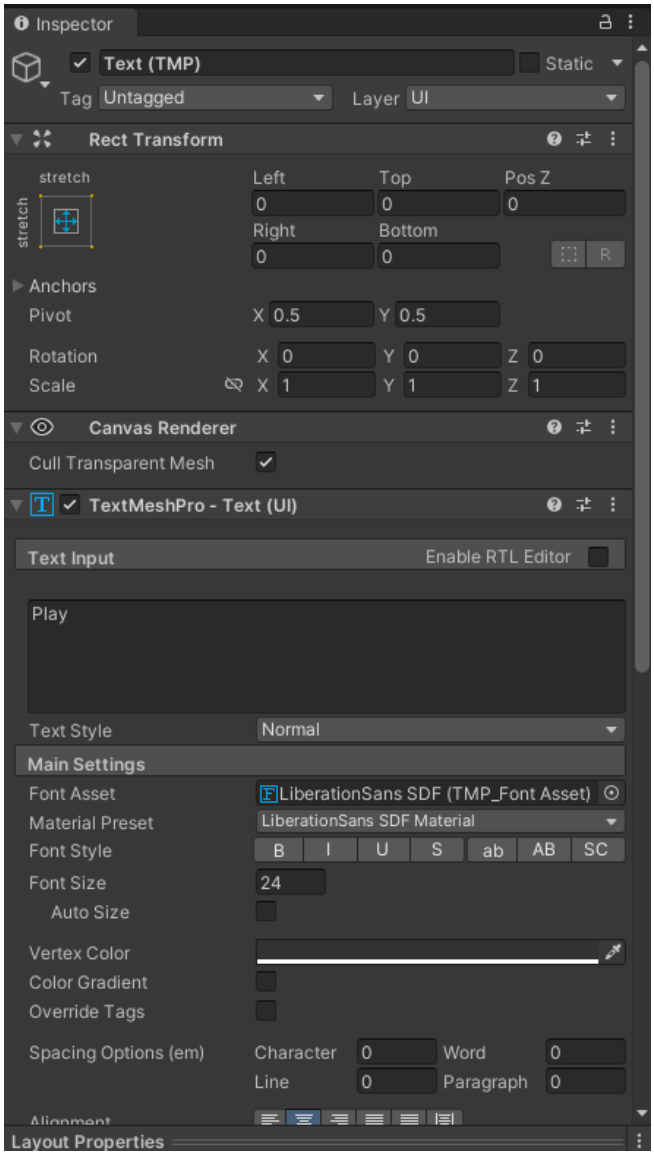
You can add as many buttons you want as long as it is reasonable. For now I will stick with the 3 buttons for this tutorial: Play, Options, and the quit button.

As usual. Don't forget to name the buttons.



To add text to the button you click on the text that corresponds to the button.

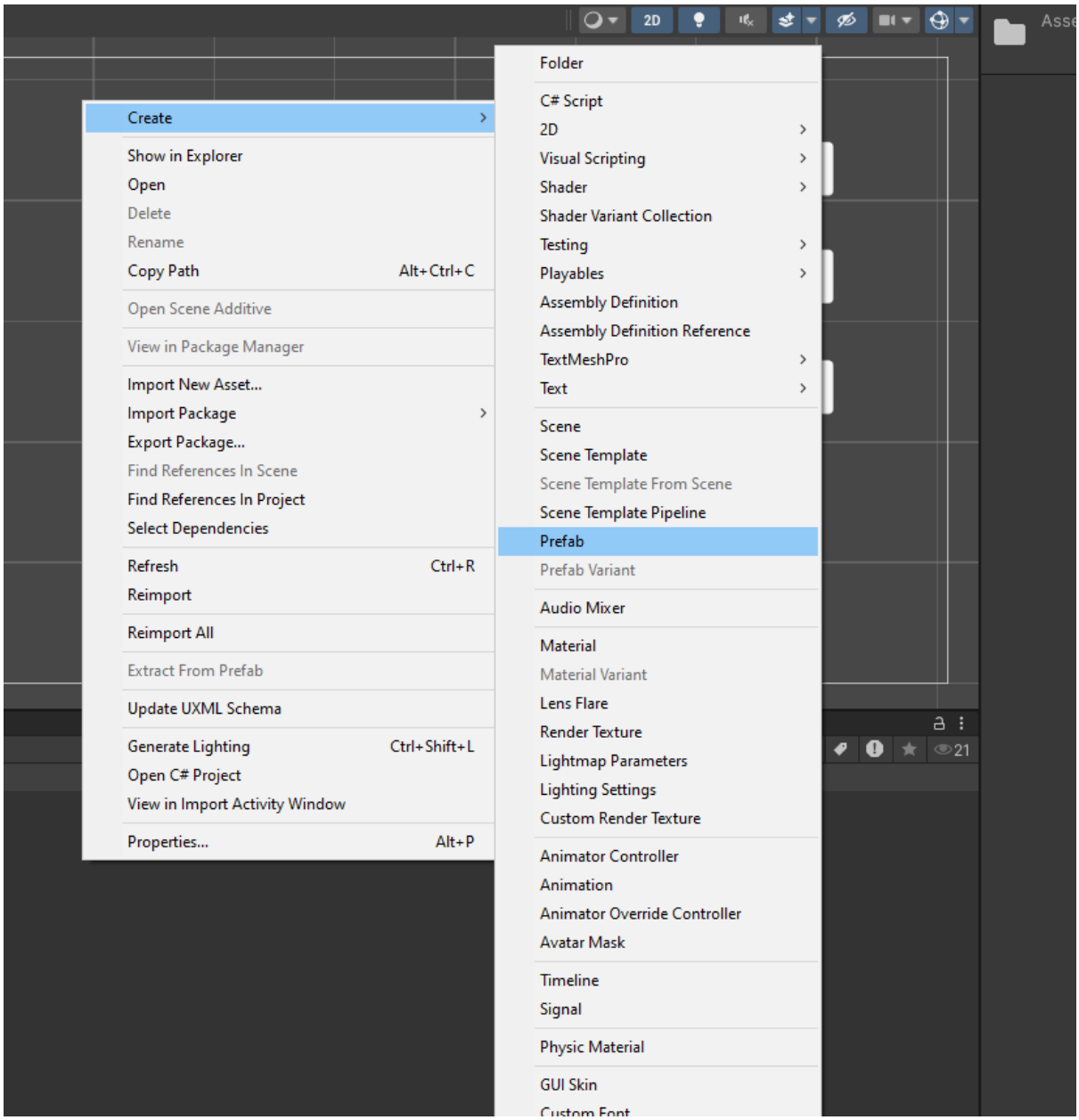
This will take you to the asset inspector where you can modify the text of your buttons.



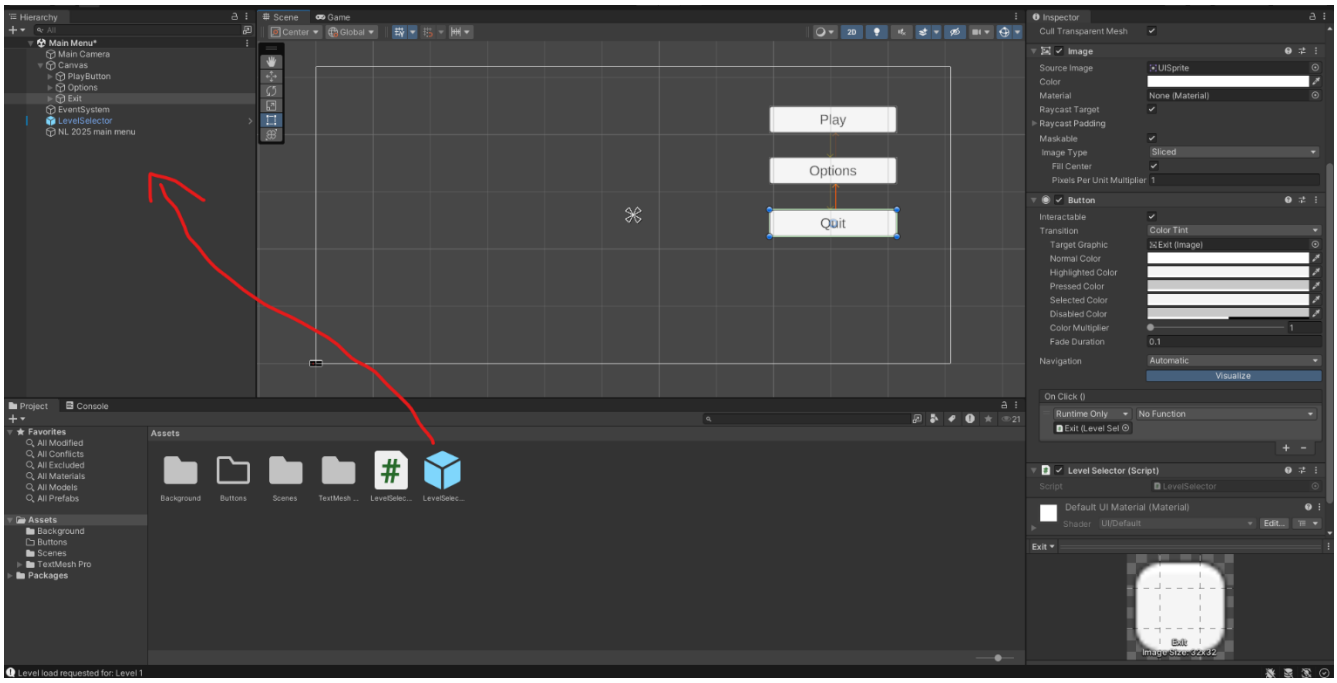
4. Button functions

Before you work on your button functions it is important to create a scene manager C# script. This will allow you to redirect players to the next scene when they start their game.

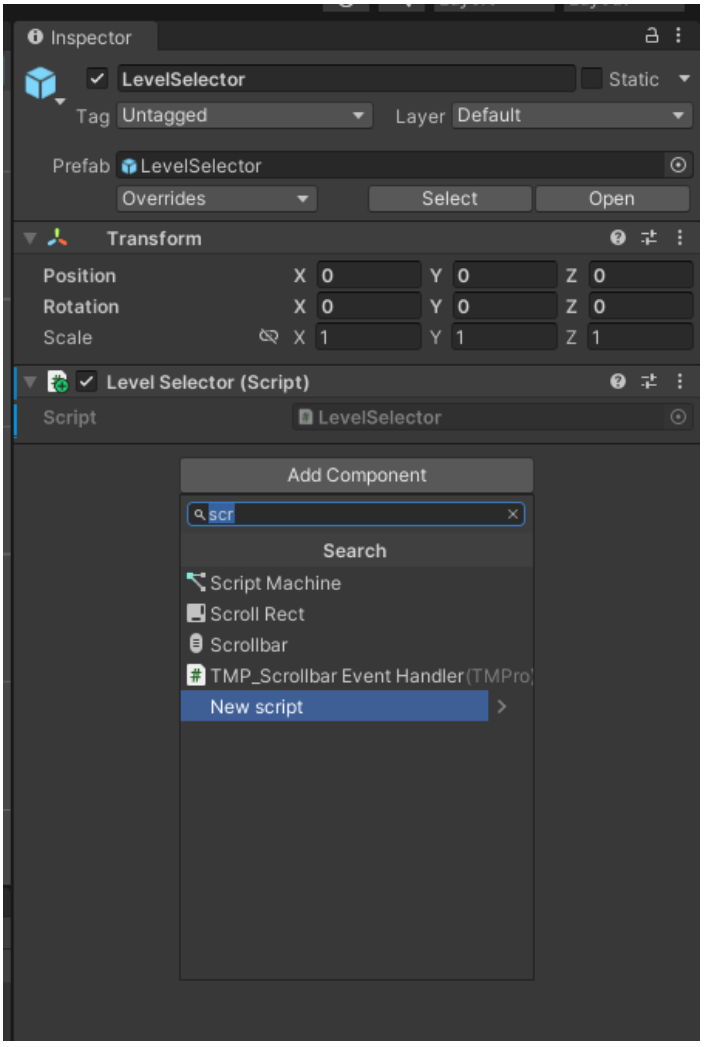
First, you create a prefab by right clicking onto the asset manager and select the prefab option. Remember to name the prefab otherwise your project will be unorganised! Personally I named my prefab “LevelSelector”.



After you created the level selection prefab, drag it to the hierarchy.



Select the prefab and add a new script and name it “Level Selector” and then open it on visual studio.



Add a namespace called “SceneManagement”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
```

Next add a public void function for open scene which will execute when the player presses the button that corresponds to the name of the scene the function is assigned to.

```
0 references
public void OpenScene(string name)
{
    Debug.Log("Level load requested for: " + name)

    SceneManager.LoadScene(name);
}
```

The debug log in the script will notify you when your game loads the scene.