

# Unity UI - Making simple menu screens

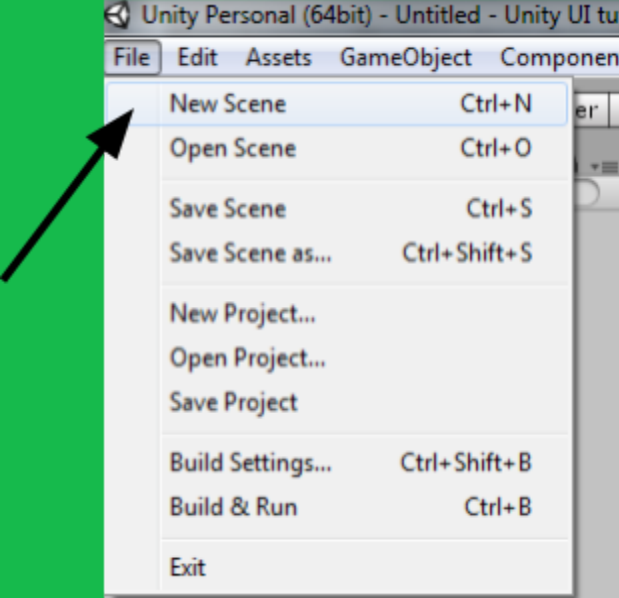
## Task 1. Setup the main menu scene and UI Canvas

Explanation

- We will create our Main Menu scene
- Then, we will create a Canvas to hold our UI elements

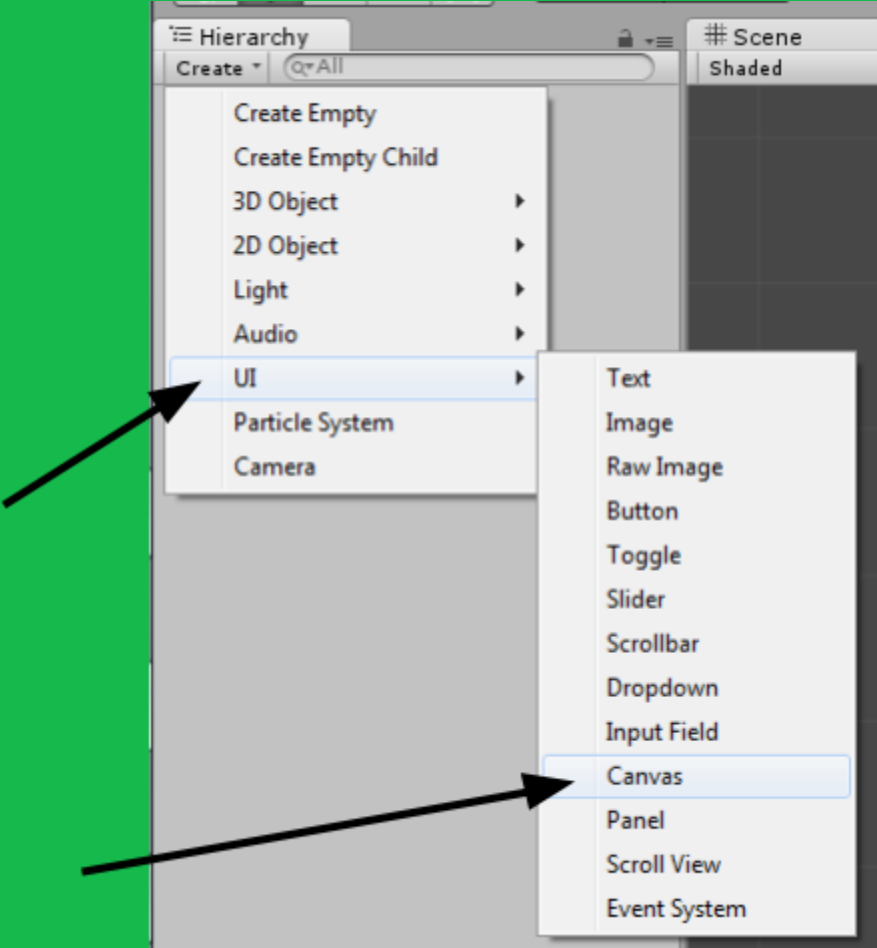
Do this

- In the top menu click **File**
- Select **New Scene**



Do this

- In the **Hierarchy**, click the **Create** button
- Select **UI**
- Select **Canvas**



Useful links

- Learn more about **Canvas**

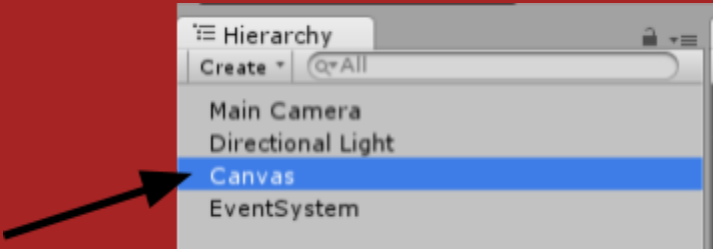
[Canvas - Manual](#)

[Canvas - Scripting](#)

[Canvas - Video](#)

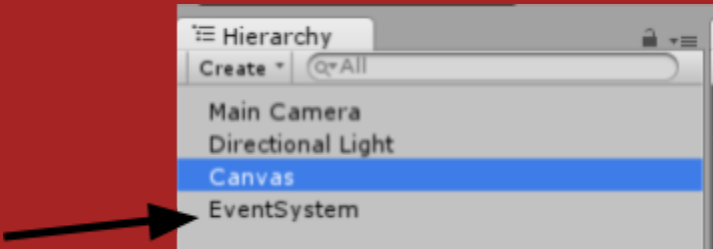
Check this

- Check in the **Hierarchy** that you have a **Canvas**



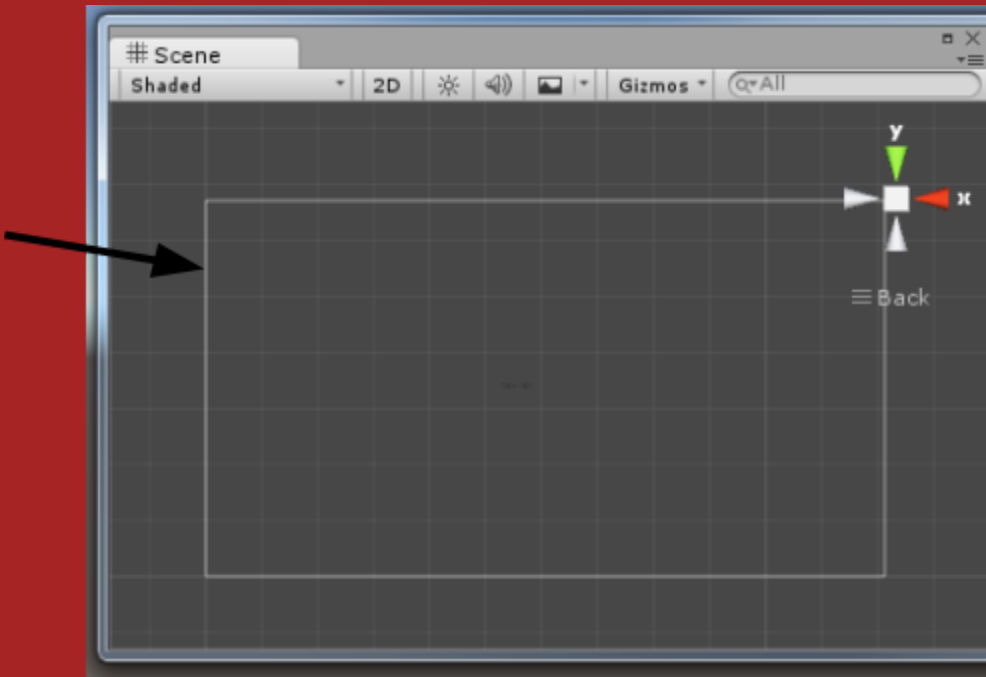
Check this

- When you create a **Canvas**, Unity will also create a **EventSystem**
- Check in the **Hierarchy** that you have a **EventSystem**



Check this

- Check in the **Scene** view for the **white outline** of the **Canvas**



Useful links

- Learn more about **Event System**
- Learn more about **Standalone Input Module**

[EventSystem Manager - Manual](#)  
[Standalone Input Module - Manual](#)

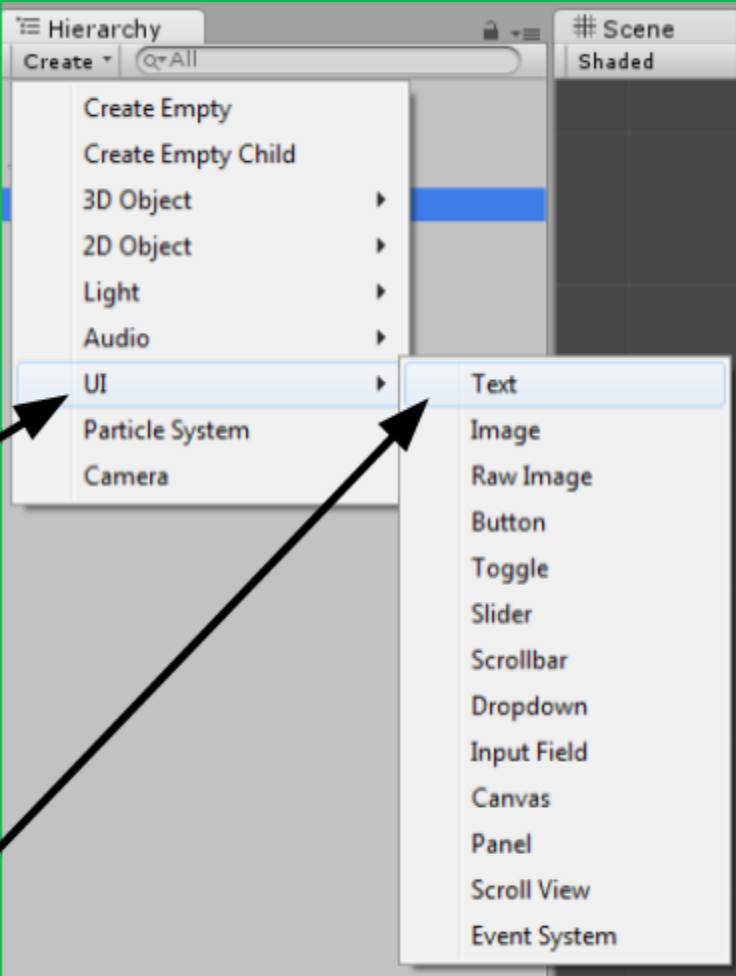
## Task 2. Create a title for the menu

### Explanation

- Our main menu needs a title
- We will create a **Text** field in the **Canvas** for this

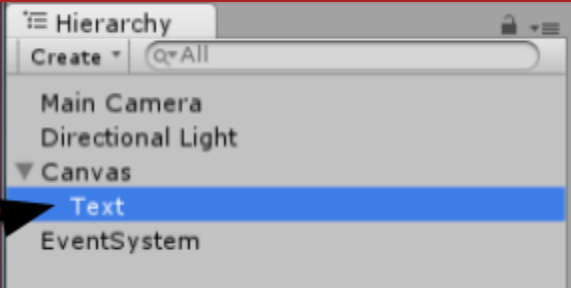
### Do this

- Select the **Canvas** in the **Hierarchy**
- In the **Hierarchy**, click the **Create** button
- Select **UI**
- Select **Text**



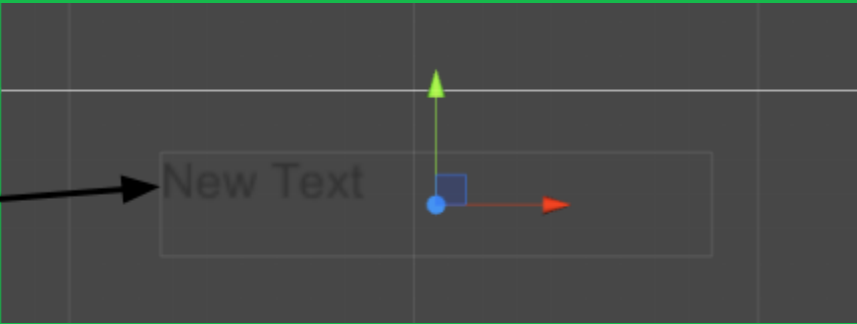
### Check this

- Check in the **Hierarchy** that you have a **Text** field



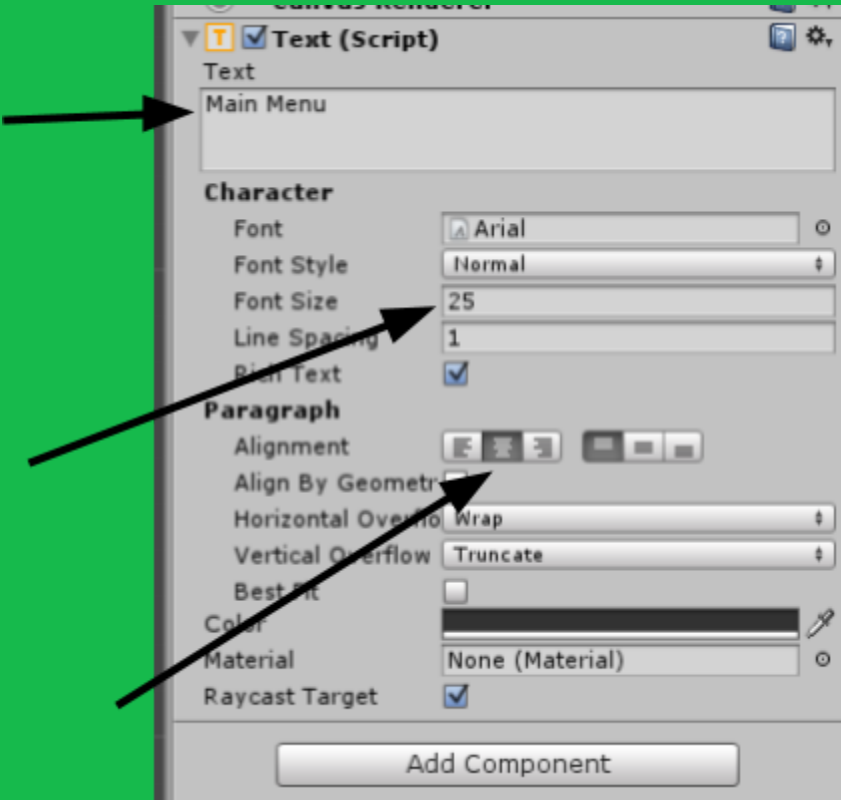
### Do this

- Select the **Text** in the **Hierarchy**
- In the **Scene view**, move the **Text** to the top of the **Canvas** outline



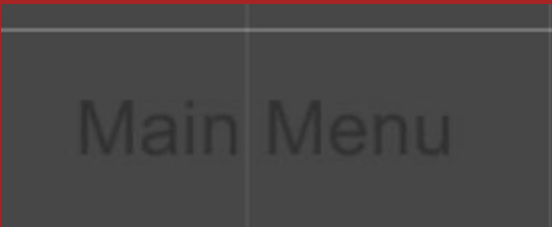
Do this

- Select the **Text** in the **Hierarchy**
- In the **Inspector**, type **Main Menu** in the **Text** field
- Set the **Font Size** to **50**
- Click the **Centre** button in the **Alignment** section



Check this

- Check in the **Scene view** that you have something like this
- The text should be a bit larger and centered



Useful links

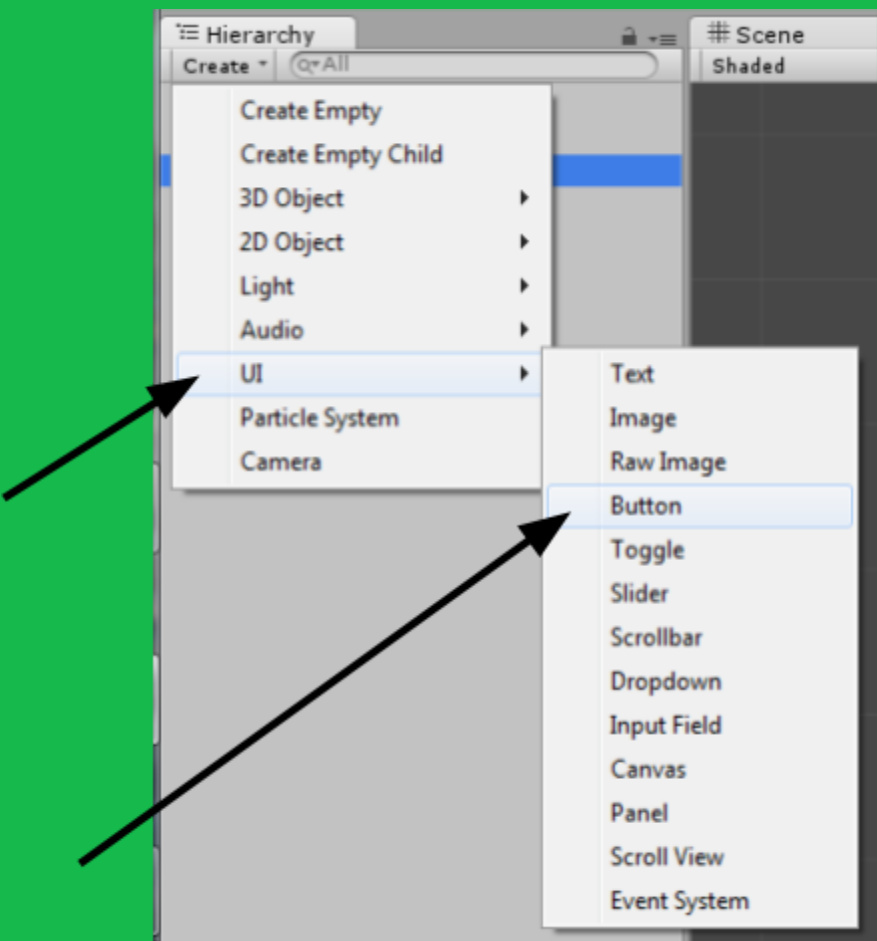
- Learn more about **UI Text**

[Text - Manual](#)  
[Text - Scripting](#)  
[Text - Video](#)

### Task 3. Create a button for the menu

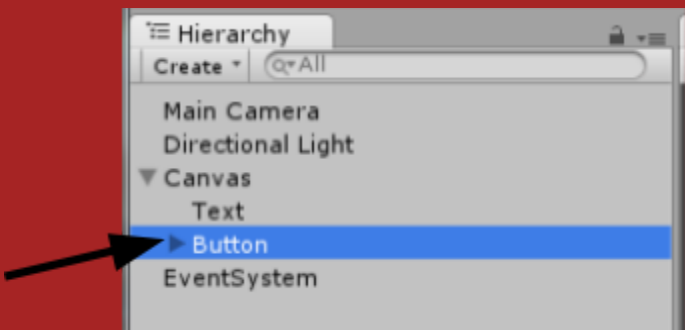
#### Do this

- Select the **Canvas** in the **Hierarchy**
- In the **Hierarchy**, click the **Create** button
- Select **UI**
- Select **Button**



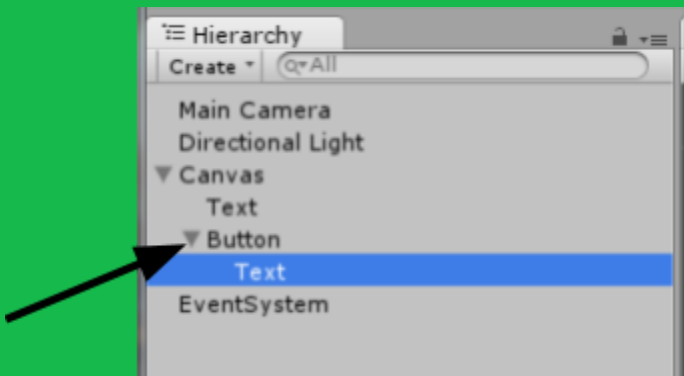
#### Check this

- Check in the **Hierarchy** that you have a **Button**



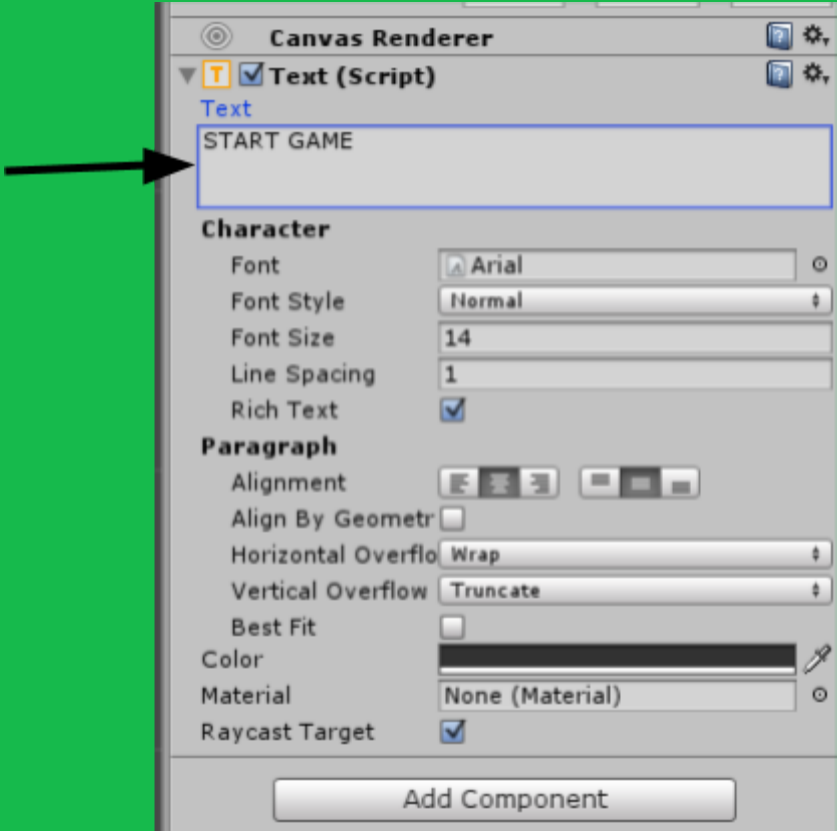
#### Do this

- Click the triangle next to the **Button** in the **Hierarchy** to open it
- You should see a **Text** field as a child of the **Button**



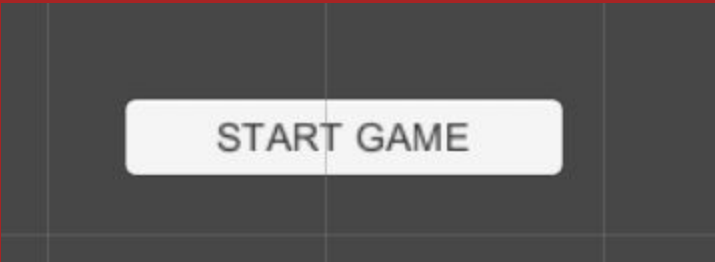
Do this

- Select the **Button's Text** field in the **Hierarchy**
- In the Inspector, type **START GAME** in the Text field



Check this

- Check in the **Scene** view that your **Button** looks like this



Useful links

- Learn more about **UI Button**
  - [Button - Manual](#)
  - [Button - Scripting](#)
  - [Button - Video](#)

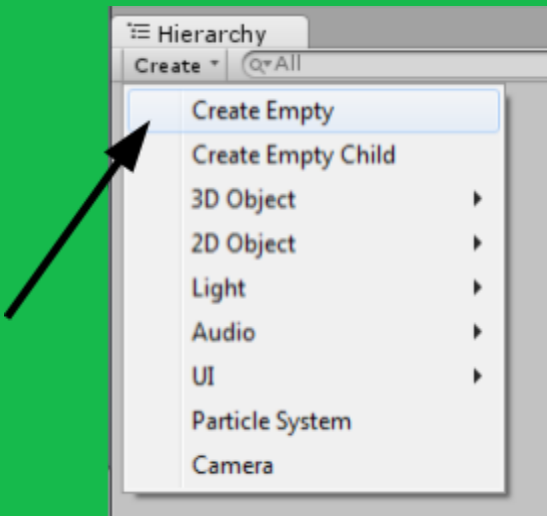
## Task 4. Create a Game Manager script

### Explanation

- We need a script to tell our buttons what to do!
- The START GAME button will load the Game scene, which we will create later

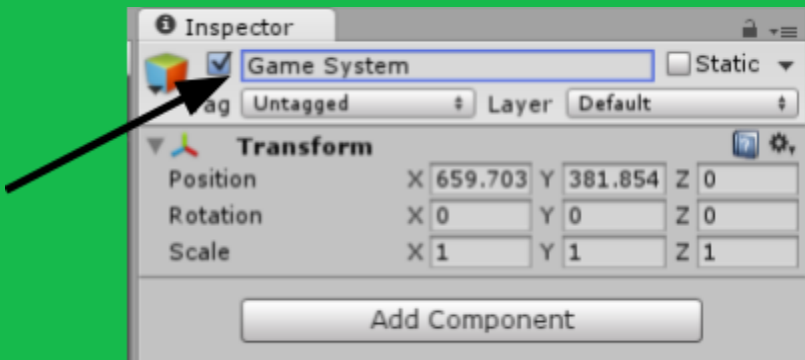
### Do this

- In the **Hierarchy**, click the **Create** button
- Click **Create Empty**



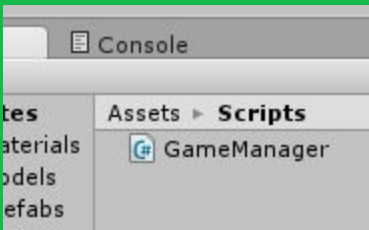
### Do this

- In the **Hierarchy**, select the new **GameObject**
- In the **Inspector**, change the name to **Game System**



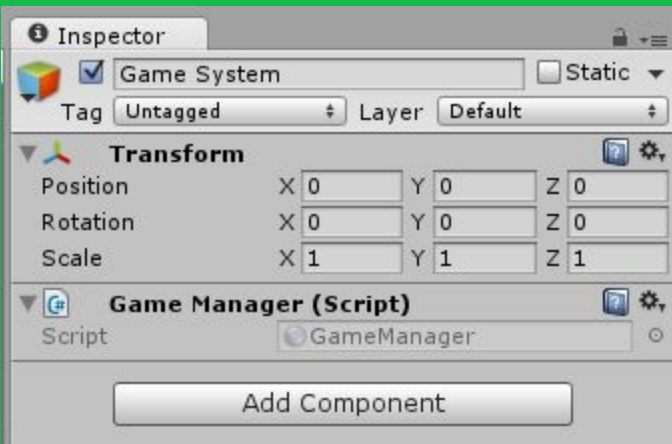
### Do this

- In the **Project view**, press the **Create** button
- Click **C# Script**
- Name the new script **GameManager**



### Do this

- Drag the **GameManager** script from the **Project view** onto the **Game System**



## Task 5. Code our button methods

### Explanation

- We will create 3 methods:
  - **StartGame** - to load the **Game** Scene
  - **EndGame** - to load the **Game Over** scene
  - **BackToMainMenu** - to load the **Main Menu** scene

### Do this

- Type out this code into your script file
- Make sure your code is **EXACTLY** the same!

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

public class GameManager : MonoBehaviour {

    public void StartGame()
    {
        SceneManager.LoadScene ("Game");
    }

    public void EndGame()
    {
        SceneManager.LoadScene ("Game Over");
    }

    public void BackToMainMenu()
    {
        SceneManager.LoadScene ("Main Menu");
    }
}
```

### Check this

- Make sure you have imported the **SceneManager** library
- Check you have the line below at the top of your script

```
using UnityEngine.SceneManagement;
```

### Explanation - StartGame method

- The method is public so other component (Like our buttons) can use it

```
public void StartGame(){
    SceneManager.LoadScene ("Game");
}
```

### Explanation - Line 1

- The **SceneManager** will handle loading a scene using its **LoadScene** method
- The name of the scene to load is passed as an argument (of type **String**), for this method it is the **Game** scene

```
public void StartGame(){
    SceneManager.LoadScene ("Game");
}
```

### Explanation - the other methods

- Both **EndGame** and **BackToMainMenu** have the same code as **StartGame**, apart from the scenes they load
- **EndGame** will load the **Game Over** scene
- **BackToMainMenu** will load the **Main Menu** scene

```
public void EndGame()
{
    SceneManager.LoadScene ("Game Over");
}

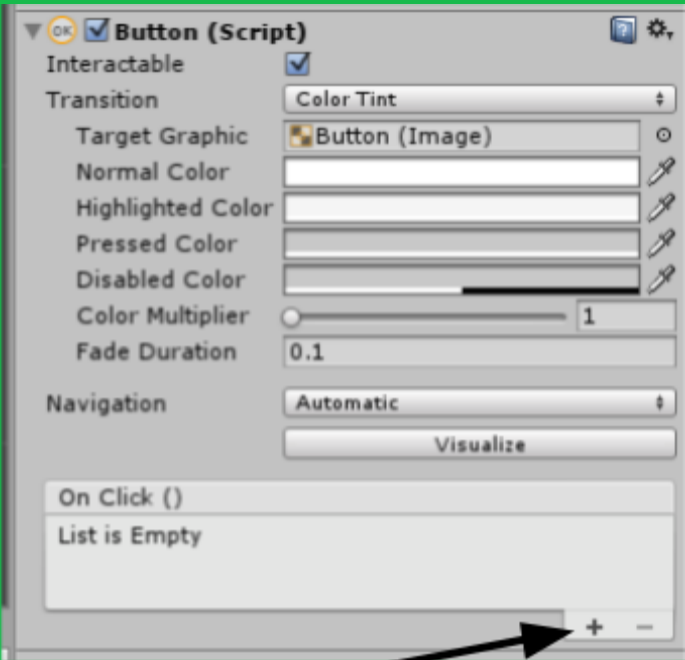
public void BackToMainMenu()
{
    SceneManager.LoadScene ("Main Menu");
}
```



# Task 5. Connect the START GAME Button to the StartGame method

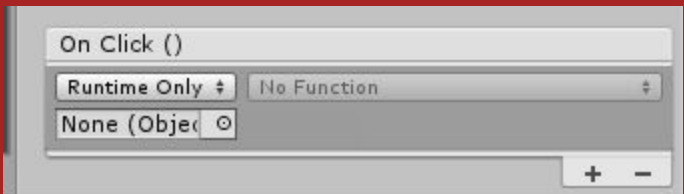
## Do this

- In the **Hierarchy**, select the **START GAME** button
- In the Inspector, Click the Plus (+) button on the **On Click()** event



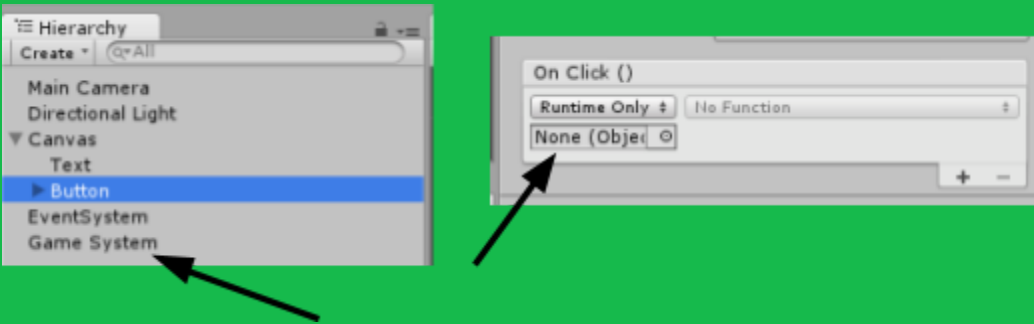
## Check this

- Check the **On Click()** event has a new event
- It should look like this



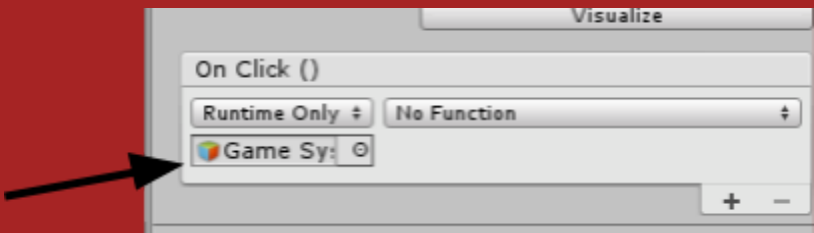
## Do this

- In the **Hierarchy**, drag the **Game System** GameObject onto the box of the **On Click()** event



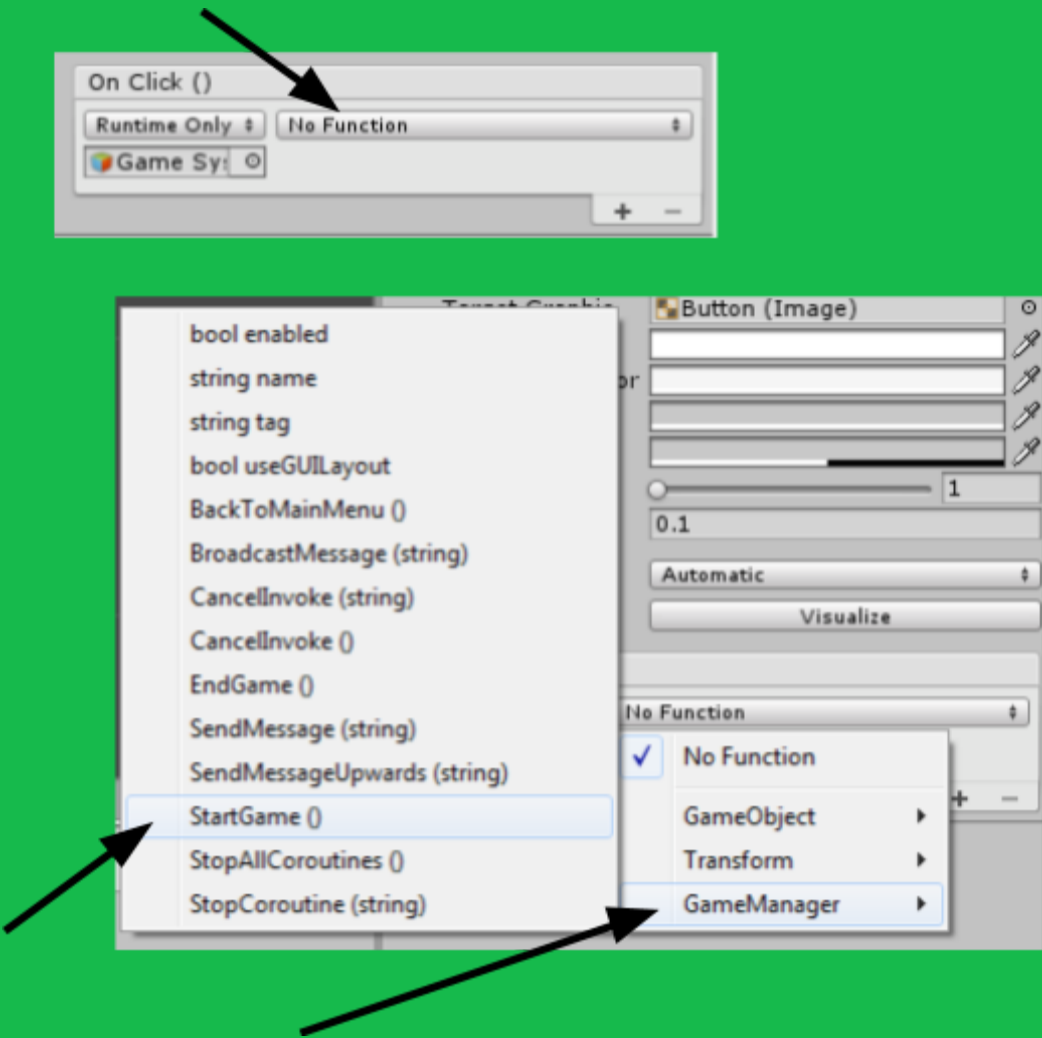
## Check this

- Check the **On Click()** event has a reference to the **Game System** GameObject
- It should look like this



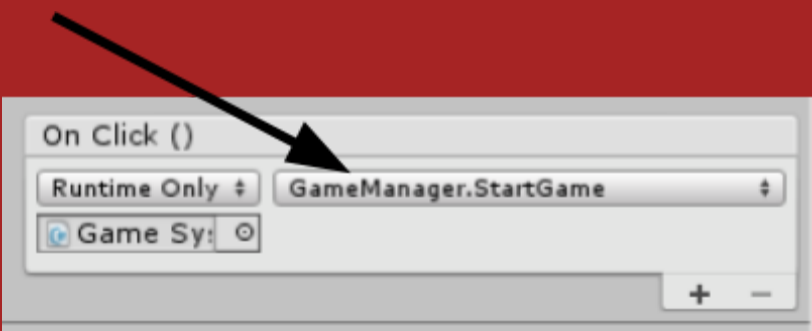
Do this

- Click the dropdown menu of the **On Click()** event
- Select **GameManager**
- Select **StartGame ()**



Check this

- Check the **On Click()** event has the **StartGame** method selected
- It should say **GameManager.StartGame**
- The **Game System** icon will change to a script icon



## Task 6. Create our other scenes

### Explanation

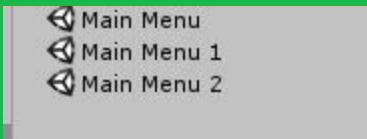
- We will create 2 more scenes
- One for the **Game**
- One for **Game Over**
- We can copy and paste the **Main Menu** scene, then edit

### Do this

- Save the scene you have been working in
- **Ctrl + s**
- Name the scene **Main Menu**

### Do this

- In the **Project view**,select the **Main Menu** scene
- In the top menu click **Edit > Duplicate** twice



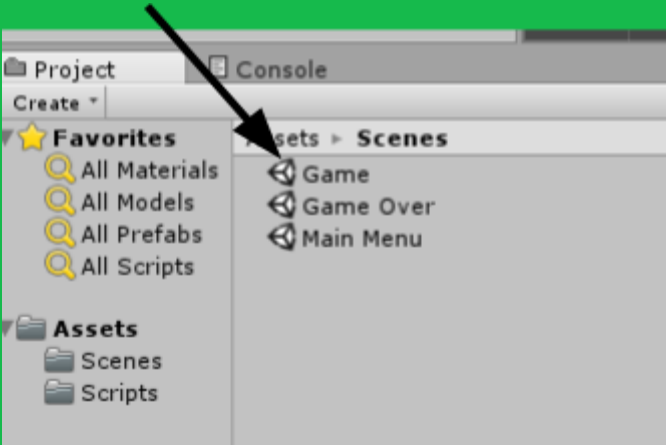
### Do this

- In the **Project view**, select **Main Menu 1**
- Press the **F2** key and rename the scene to **Game**
- Select **Main Menu 2**
- Press the **F2** key and rename the scene to **Game Over**



### Do this

- In the **Project view**, double click the **Game** scene to open it



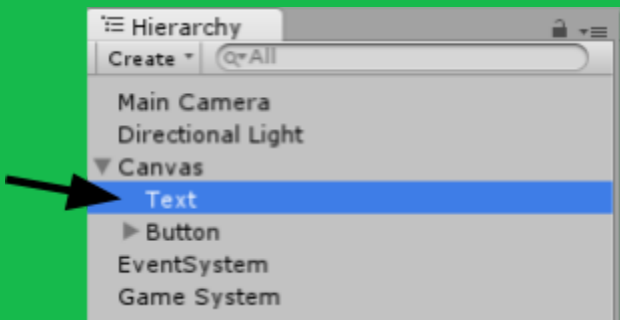
## Task 6. Setup our Game scene button and text

### Explanation

- We will setup the text on the button and title
- Then we will set the button event to load the Game Over scene

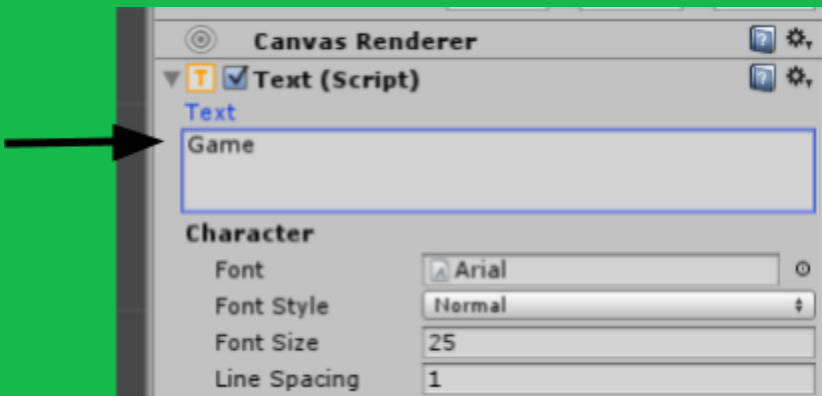
### Do this

- In the **Hierarchy**, select the title **Text** field in the **Canvas**



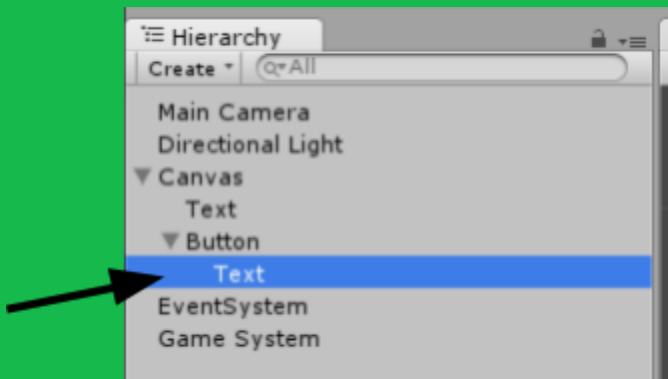
### Do this

- In the **Inspector**, change the **Text** field to **Game**



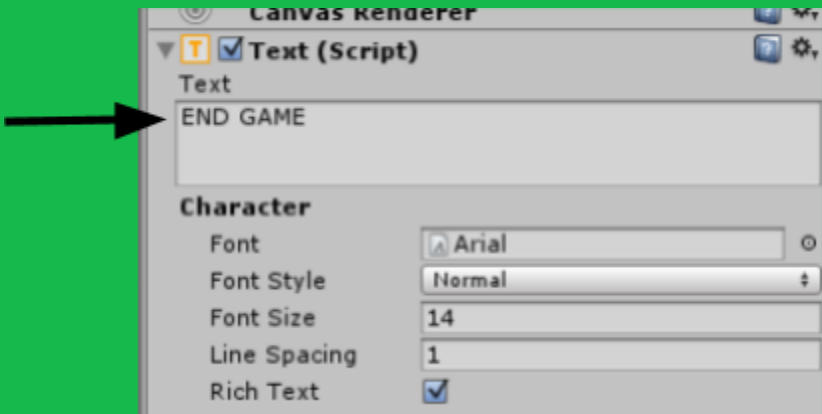
### Do this

- In the **Hierarchy**, select the **Text** field inside the **Button**



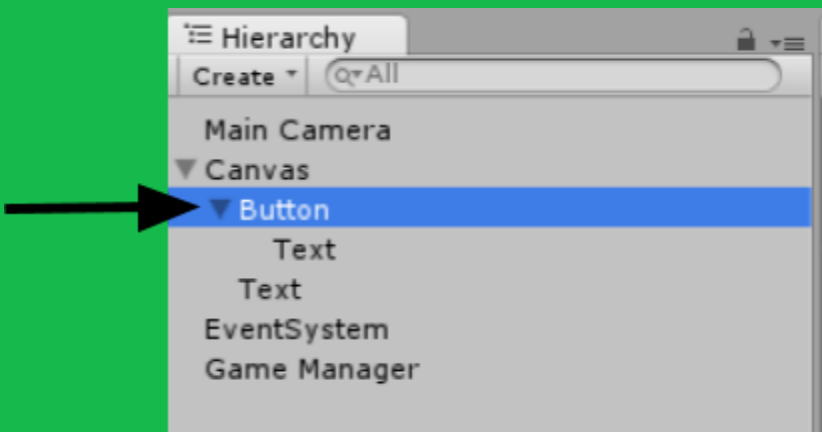
### Do this

- In the **Inspector**, change the **Text** field to **END GAME**



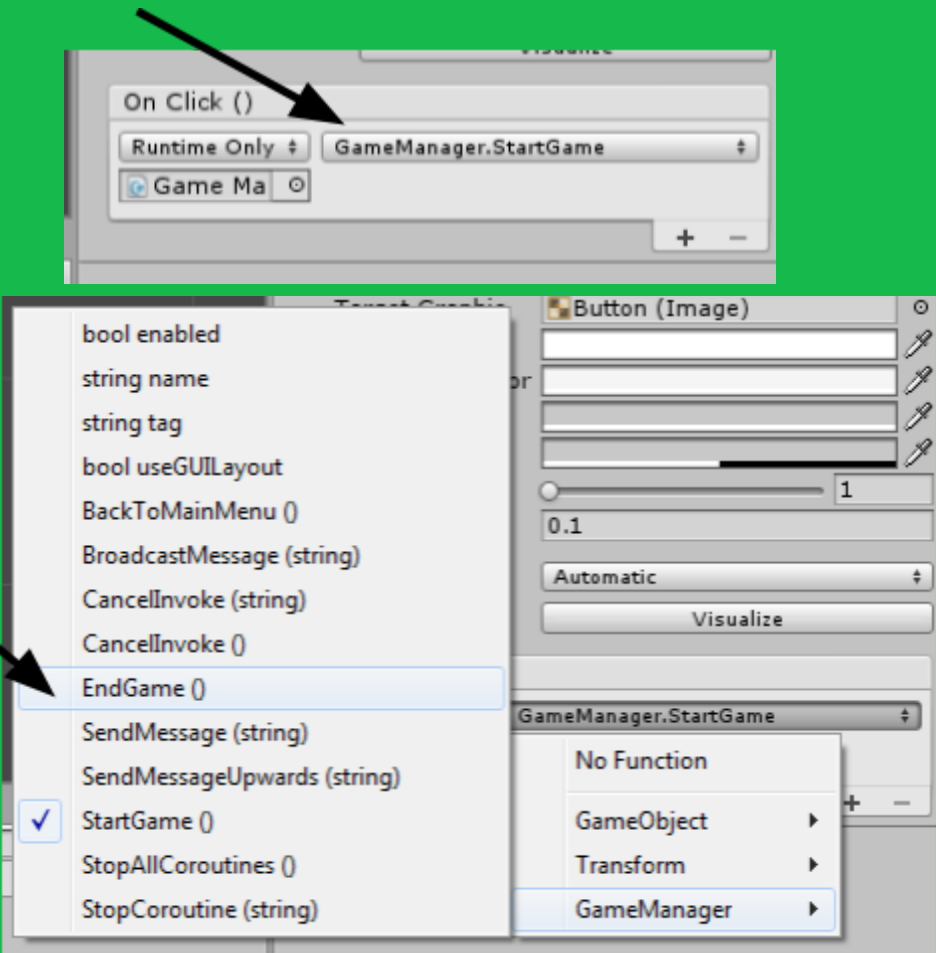
### Do this

- In the **Hierarchy**, select the **Button** in the **Canvas**



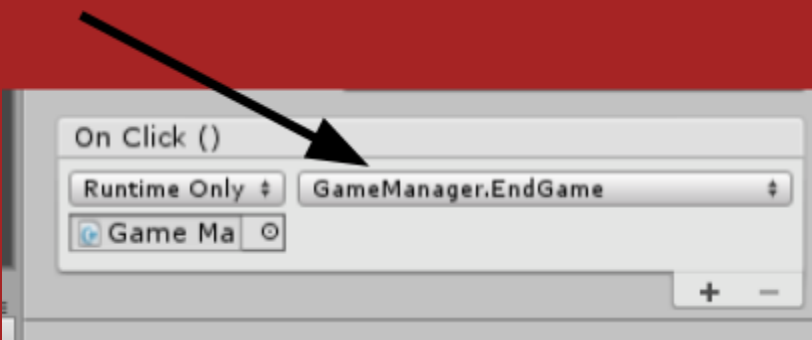
Do this

- In the **On Click()** event in the **Inspector**, click the method dropdown
- Select **Game Manager**
- Select **EndGame()**



Check this

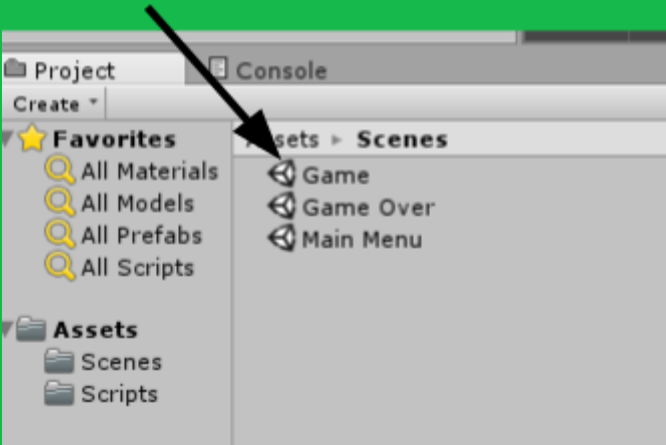
- Check the **On Click()** event has the **EndGame** method selected
- It should say **GameManager.EndGame**



# Task 7. Setup our Game Over scene button and text

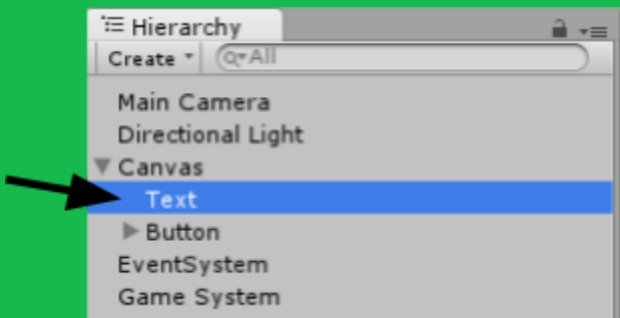
## Do this

- In the **Project view**, double click the **Game** scene to open it



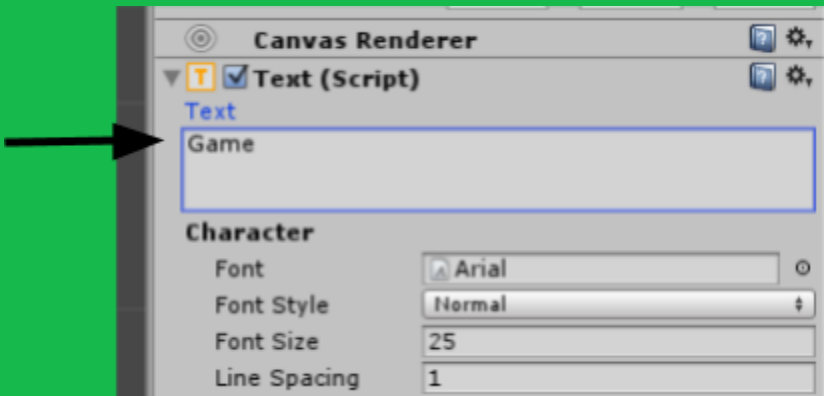
## Do this

- In the **Hierarchy**, select the title **Text** field in the **Canvas**



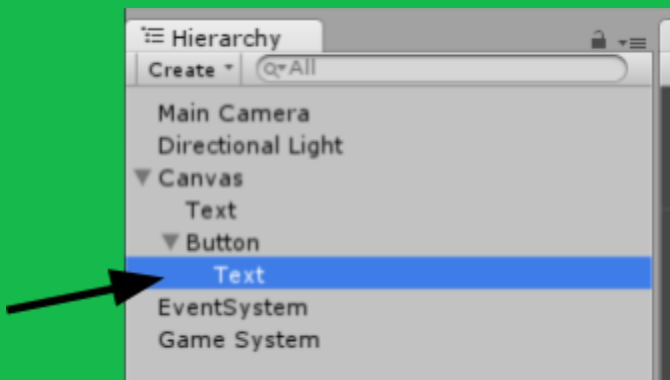
## Do this

- In the **Inspector**, change the **Text** field to **Game Over**



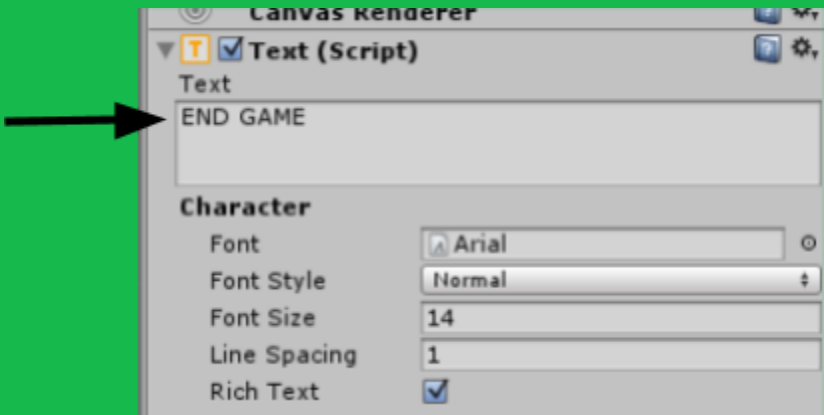
## Do this

- In the **Hierarchy**, select the **Text** field inside the **Button**



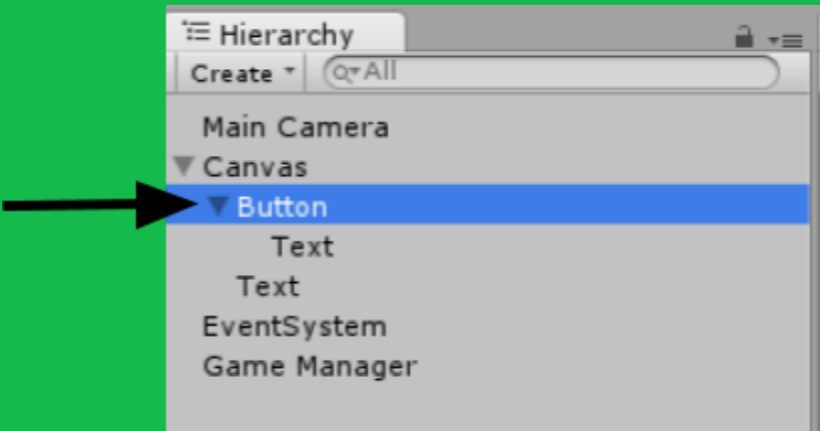
## Do this

- In the **Inspector**, change the **Text** field to **MAIN MENU**



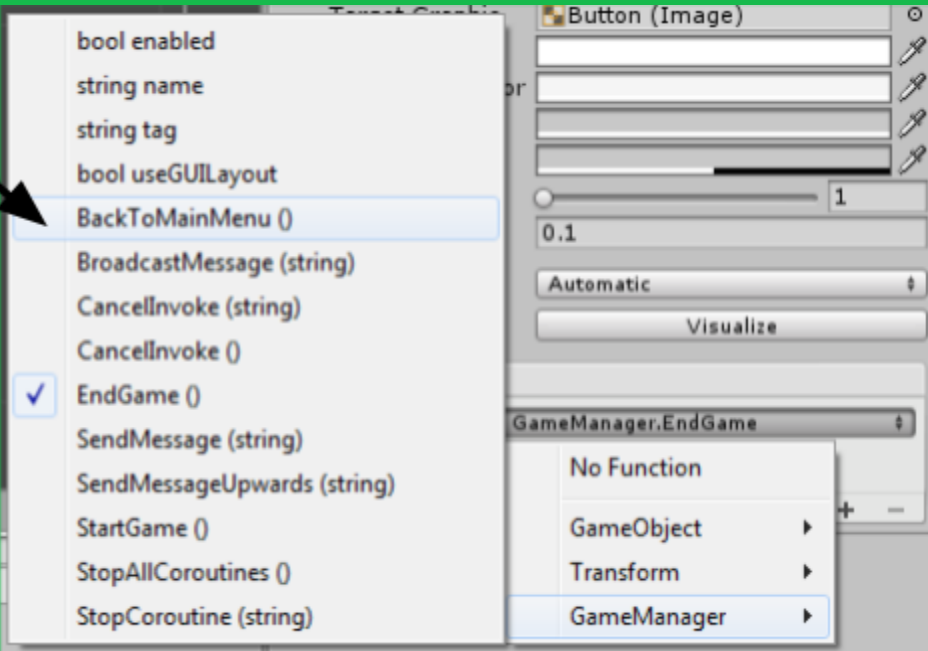
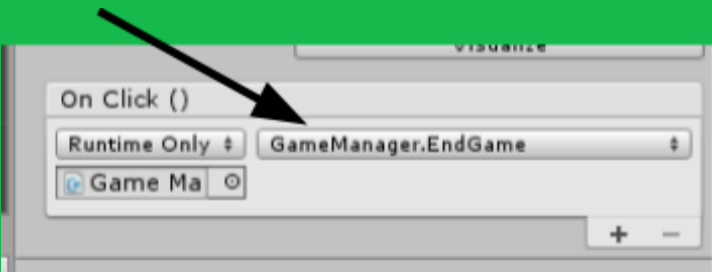
Do this

- In the **Hierarchy**, select the **Button** in the **Canvas**



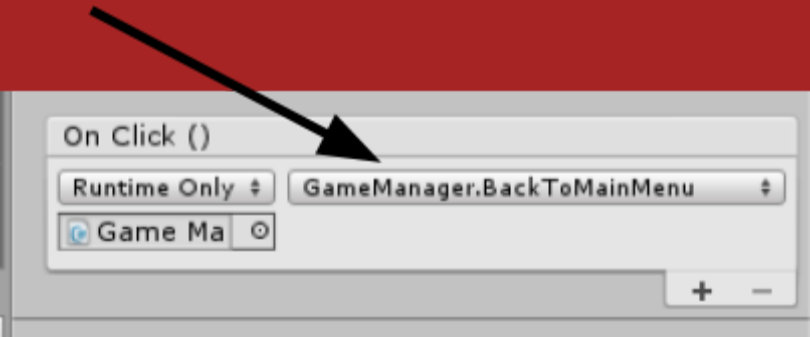
Do this

- In the **On Click()** event in the **Inspector**, click the method dropdown
- Select **Game Manager**
- Select **BackToMainMenu()**



Check this

- Check the **On Click()** event has the **EndGame** method selected
- It should say **GameManager.EndGame**



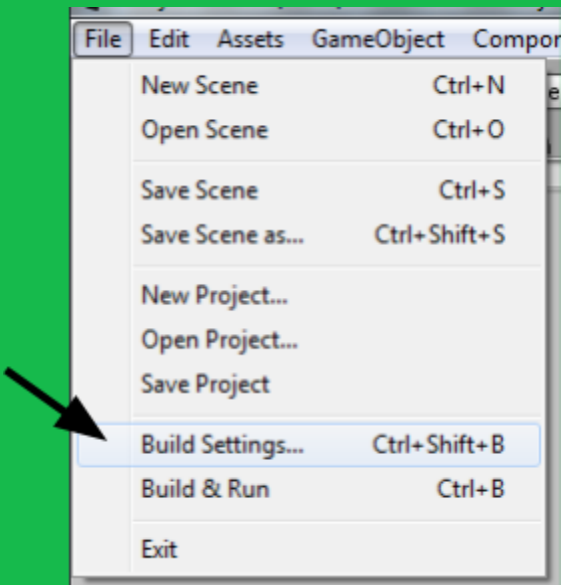
# Task 8. Setup the scenes in the build panel

## Explanation

- In order to load different scenes when playing the game, we need to list the scenes in the **Build settings** panel

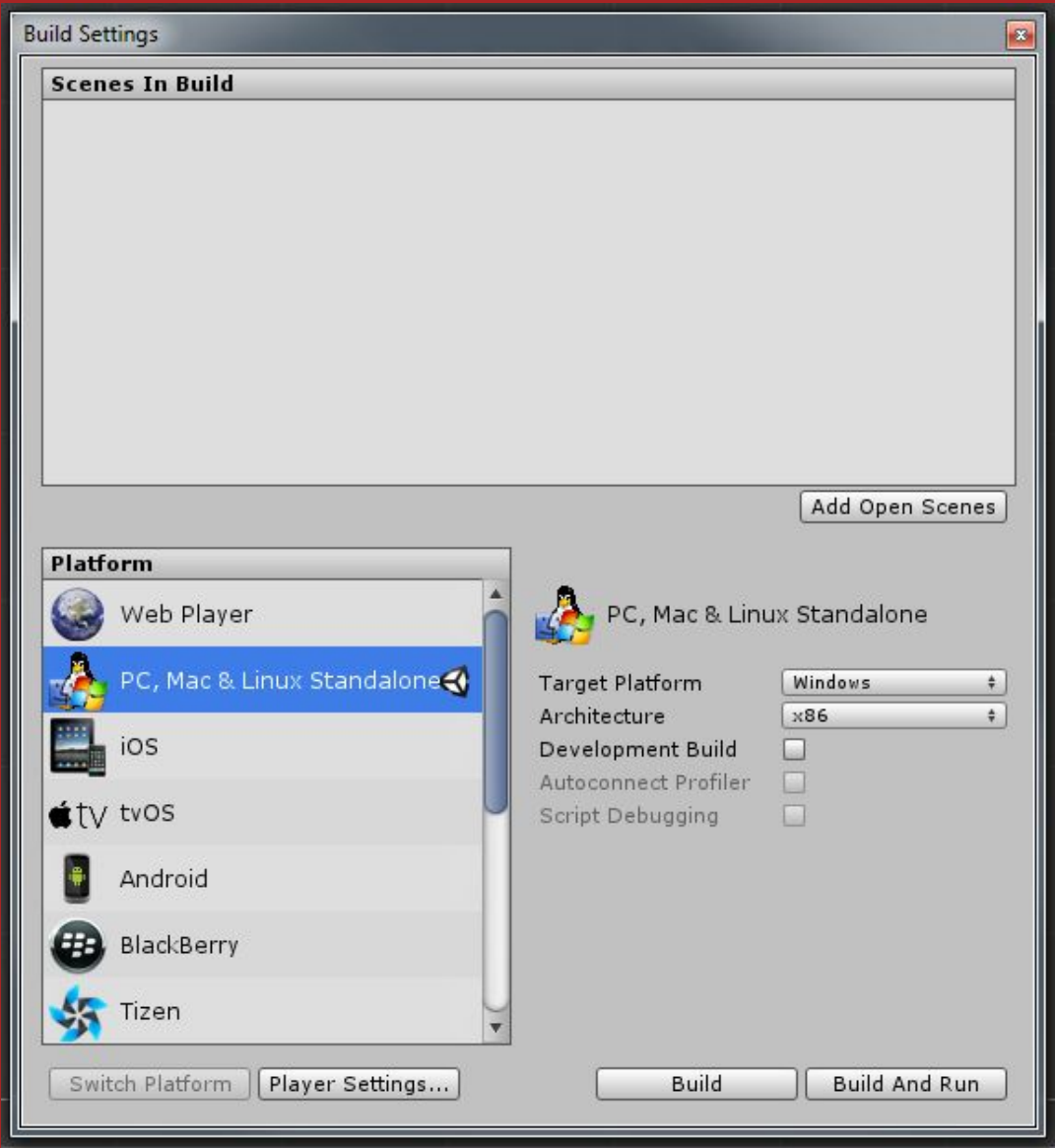
## Do this

- Open the **Build settings** panel
- On the Top menu, click **File**
- Select **Build Settings**



## Check this

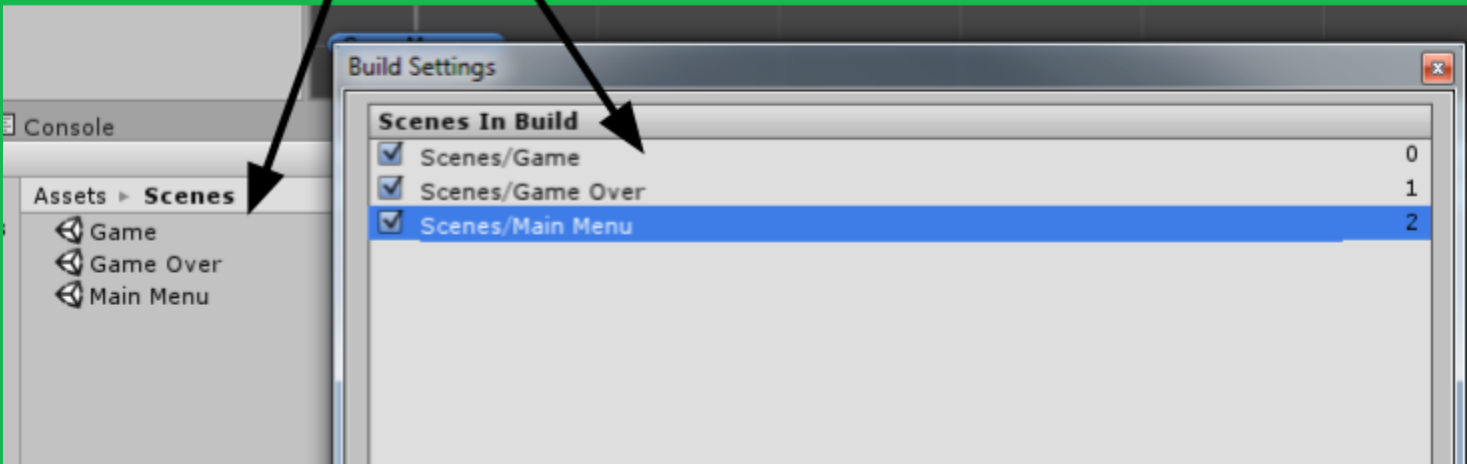
- Check the Build Settings panel is displaying





Do this

- Drag each scene from the **Project** panel onto the **Scenes In Build** section of the **Build Settings** panel



Explanation

- You are now ready to test the scenes!

Useful links

- Learn more about **Build Settings** [Build Settings - Manual](#)

