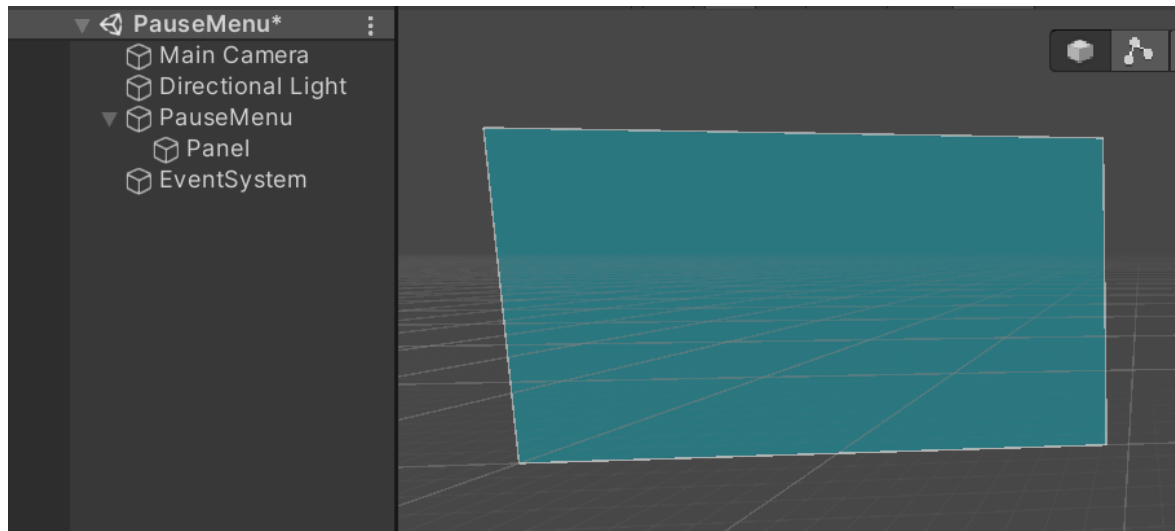# Unity Pause menu tutorial

Create your pause menu

1. Add a canvas called PauseMenu
2. Make a panel that is a child object, set the colour to what you want and have the opacity low at around 100

3. Create a button as a child of the canvas and change it to look how you want
4. Write RESUME in the button text
5. Duplicate the button twice and rename these two other buttons Menu and Quit
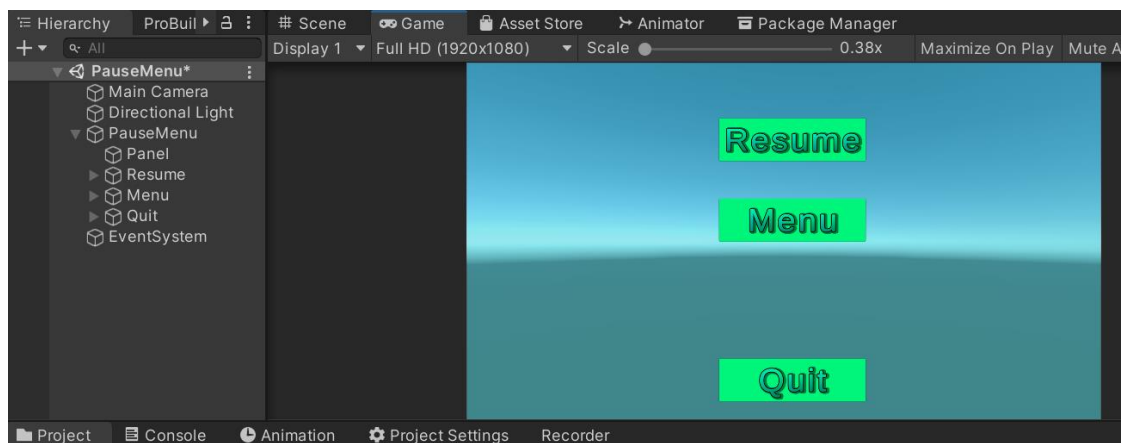
Program the quit and menu buttons

1. Add an Empty Gameobject parent to the PauseMenu by right clicking
2. On this GameObject add a new script called PauseMenu
3. Remove the Void Start
4. Copy this code

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

// Unity Script | 0 references
public class PM : MonoBehaviour
{
    // Update is called once per frame
    // Unity Message | 0 references
    void Update()
    {

    }

    // 0 references
    public void QuitGame()
    {
        Debug.Log("QUIT");
        Application.Quit();
    }
}
```

This code is used whenever you want to quit the game and we will be attaching it to our quit button soon.

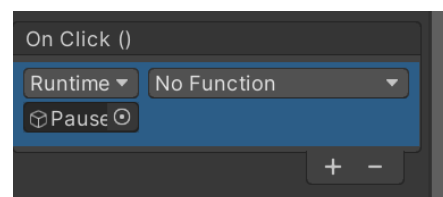5. Now copy this code which will be used for our menu button

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
   Unity Script | 0 references
6  public class PM : MonoBehaviour
7  {
8      // Update is called once per frame
       Unity Message | 0 references
9      void Update()
10     {
11
12     }
13
       0 references
14     public void Menu()
15     {
16         Time.timeScale = 1f;
17         SceneManager.LoadScene(0);
18     }
       0 references
19     public void QuitGame()
20     {
21         Debug.Log("QUIT");
22         Application.Quit();
23     }
24 }
```

Presumably you already have your main menu and it is set to the first scene to load, if this is not the case you can remove this button or change the number in LoadScene to match that of the scene you really want to load (this can be checked by going File > Build settings and checking what number is next to your menu.
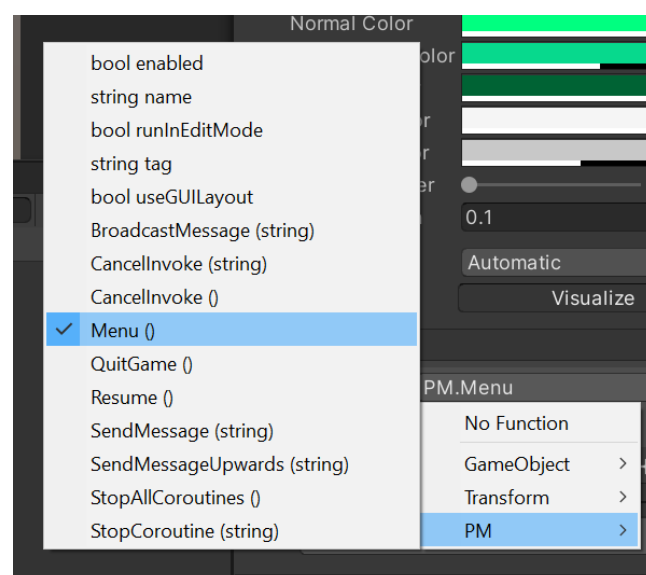
6. Save your script
7. Return to Unity and click the menu button
8. In the on click section click the + and drag the pause menu into it

On Click ()
Runtime ▾   No Function   ▾
⊡ Pause ⊙

+   −

9.Click No function > PauseMenu > Menu()

Normal Color

bool enabled
string name
bool runInEditMode
string tag
bool useGUILayout
BroadcastMessage (string)
CancelInvoke (string)
CancelInvoke ()
✓ Menu ()
QuitGame ()
Resume ()
SendMessage (string)
SendMessageUpwards (string)
StopAllCoroutines ()
StopCoroutine (string)

0.1
Automatic
Visualize

PM.Menu
No Function
GameObject   >
Transform   >
PM   >

9. Repeat steps 8 and 9 for the quit button but click QuitGame() instead

Getting the menu to appear and disappear

1. Go back into your script and add this code

```
1    using System.Collections;
2    using System.Collections.Generic;
3    using UnityEngine;
4    using UnityEngine.SceneManagement;
5
     Unity Script | 0 references
6    public class PM : MonoBehaviour
7    {
8        public GameObject PauseMenuUI;
9
10       public static bool GameIsPaused = false;
11
```

We are setting the PauseMenu as a public GameObject so that we can easily assign it and therefore have it be influenced by our script.

We are making a public bool for when the game is paused so that we can switch between the two states of paused and unpaused. This means we can check whether the game should do the code to pause the game or resume it otherwise the game would only be able to pause and not resume which wouldn't be helpful.

```
Unity Message | 0 references
void Update()
{
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        if (GameIsPaused)
        {

        }
        else
        {

        }
    }
}
```

For this script since we have multiple ways to unpause the game (hitting the button or pressing escape again) it will be easier if we can just resue the same code for both of them so while all the pausing code could be written in the if statement we are instead going to put them in a function and then put that in the if statement so the resume button can use the pause function. We will come back to this if statement later

2. Write this under the Update() but before the button functions

```csharp
1 reference
void Pause ()
{
    Cursor.lockState = CursorLockMode.Confined;
    Debug.Log("Pausing game");
    PauseMenuUI.SetActive(true);
    Time.timeScale = 0f;
    GameIsPaused = true;
}

1 reference
public void Resume ()
{
    Cursor.lockState = CursorLockMode.Locked;
    Debug.Log("Resuming game");
    PauseMenuUI.SetActive(false);
    Time.timeScale = 1f;
    GameIsPaused = false;
}
```

Cursor.lockState = CursorLockMode.confined/Locked

This may not be needed however this ensures that when your pause menu comes up you will be able to use the cursor to click the buttons. While the version used in the resume function ends with locked which will restrict the use of the cursor.

Debug.Log("Resuming/Pausing game")

This doesn't do anything it will just send a message to the console to make sure it is working.

PauseMenuUI.SetActive(true/false)

From earlier we made a public gameobject which will let us attach the pause menu UI, this bit of code means that that the gameobject attached will appear or disappear depending of if it is set to true or false so for pausing we want it to be true and for resuming we want it to be false.

Time.timeScale = 1/0f

This is not used in every pause menu but here it is used to freeze time in the game so that the player doesn't get punished while paused. By setting the time to 0f time will stop unless the object is otherwise specified or is some UI, setting it back to 1 will make everything return to normal speed.

GameIsPause = true/false

As said earlier we don't want the game to never be able to resume so we set out pause bool to be able to tell which state we are in.

3. Finally fill this bit in from earlier with the names of your functions

```csharp
// Update is called once per frame
⦿ Unity Message | 0 references
void Update()
{
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        if (GameIsPaused)
        {
            Resume();



        }
        else
        {
            Pause();
        }

    }
}
```

4. Save your script
5. Go back to Unity and go to the On click section for your resume button
6. Drag and drop the menu and apply the resume function to the resume button

```
On Click ()

Runtime Oı▼   PM.Resume                    ▼
⊞ PauseCı ⊙

                                    +  −
```
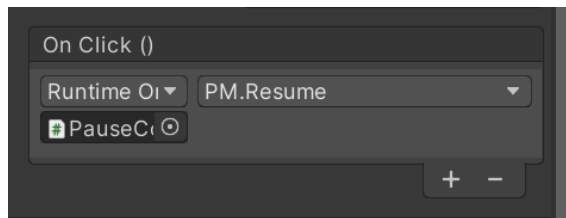
7. Disable the Pause menu in the inspector so it is only on after you press escape

```
ⓘ Inspector                              ⊟  ⋮

⬡     ☐  PauseMenu              ☐ Static ▼
         Tag Untagged    ▼   Layer UI       ▼
```
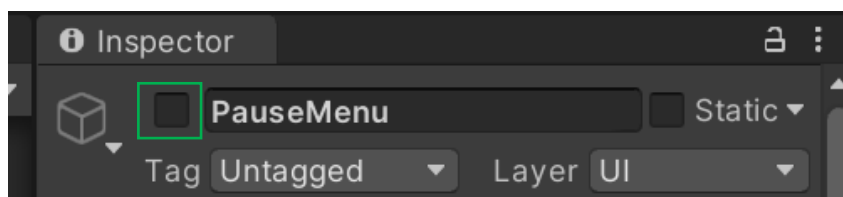
8. Finally go to the Empty Gameobject and drag the pause menu into the script

```
▼ ⊞ ✓ PM (Script)                    ❓ ᠊ᵗ ⋮

Script              ⊞ PM                        ⊙
Pause Menu UI       ⬡ PauseMenu                 ⊙
```