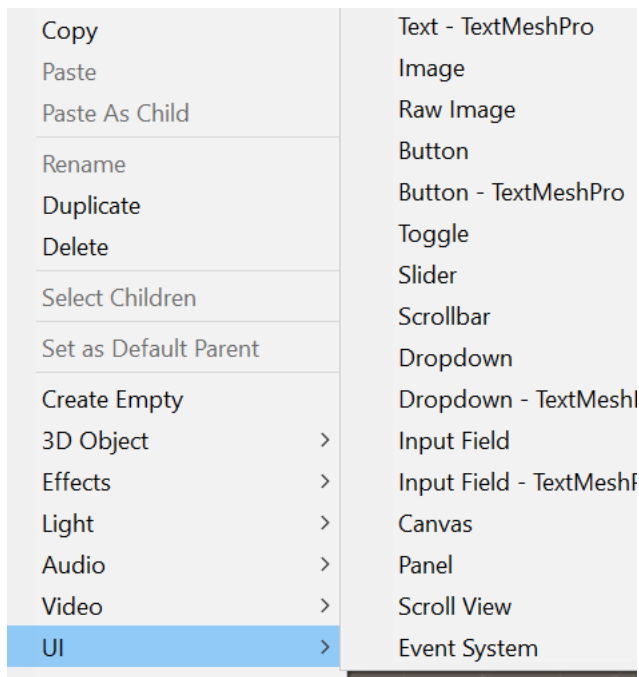
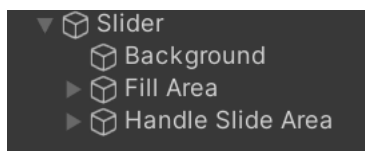


In this tutorial, im going to show you how to create a slider that will allow you to ajust the volume in your game.

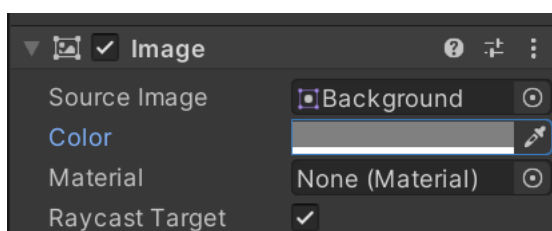
First, go into your hierachy and right click. Go to UI, and click on Slider.



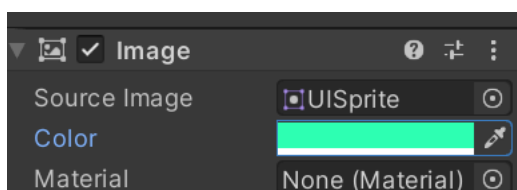
On your slider object, we are going to want to change the background and fill colour.



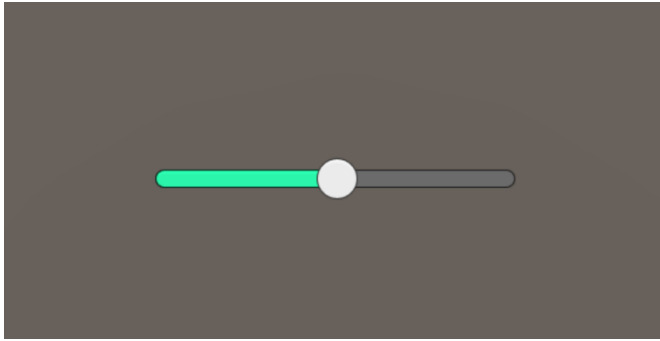
For the background, change the colour to something darker then your original colour.



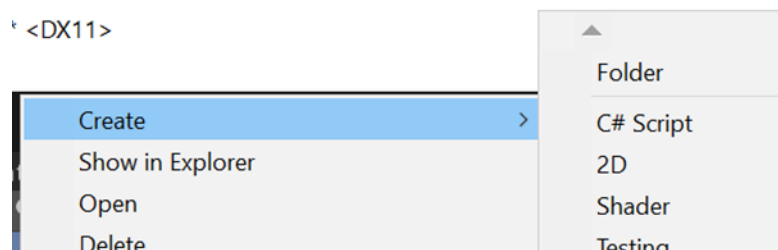
For your fill colour, change it to something bright and colourful.



When you use the slider in-game, it should look something like this.



Next, we are going to start working on our script. Right-click in our assets folder, go to Create, and select C# Script.



Rename this script 'SoundMangaer' and double click to open it.

Start of by creating a SerializeField attribute. We use this when we want our variable to be private but also show up in our editor.

```
Unity Script | 0 references
public class SoundManager : MonoBehaviour
{
    [SerializeField] Slider volumeSlider;
}
```

To reference our UI and access the UI properties, we are going to want to add a using statement like the one below.

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

Delete the Update function as we don't need it for this script. Below the start method, create a new method called 'ChangeVolume'. Here, we are going to make sure that the volume of the game matches up to the volume of the slider.

```
0 references
public void ChangeVolume()
{
    AudioListener.volume = volumeSlider.value;
}
```

Now we are going to make the volume settings save so that when you next open the game, the volume slider stays the same. Create two more private functions called 'Load' and 'Save'. In the save function, we are going to use the 'GetFloat' function to save the data. In the load function, this code sets the value of the volume slider to be the same as the value stored.

```
private void Load()
{
    volumeSlider.value = PlayerPrefs.GetFloat("musicVolume");
}

1 reference
private void Save()
{
    PlayerPrefs.SetFloat("musicVolume, volumeSlider.value");
}
```

Go back to the ChangeVolume method. Here, we are going to reference that Save method for whenever the player changes the volume of the slider.

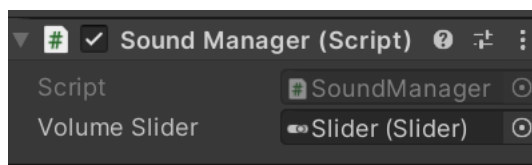
```
0 references
public void ChangeVolume()
{
    AudioListener.volume = volumeSlider.value;
    Save();
}
```

In the Start method, if there isn't any saved data, we want the volume to be set to '1' which is equal to 100%. Else, if there is saved data, then reference the Load method.

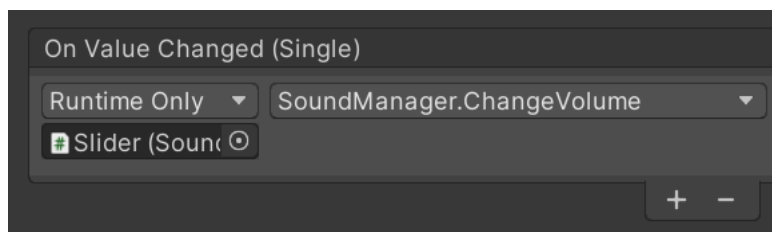
```
// Start is called before the first frame update
Unity Message | 0 references
void Start()
{
    if (!PlayerPrefs.HasKey("musicVolume"))
    {
        PlayerPrefs.SetFloat("musicVolume", 1);
        Load();
    }

    else
    {
        Load();
    }
}
```

Attach the Sound Manager Script to your slider, and drag your slider component to where it says 'Volume Slider'.



Finally, we want to call the ChangeVolume method every time the player changes the value of the slider.



You should now have a functioning volume slider which increases and decreases the volume of your game and saves your final input.