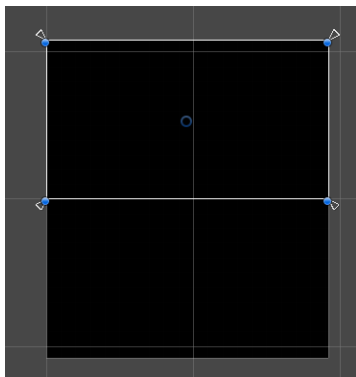


# Tutorial 3 – Scene Transitions

This tutorial will go through creating a script that adds screen transitions when moving between scenes.

The first step is to create a canvas. Add two black panels to the canvas, both the same size as the screen. One panel should be covering the screen and the other directly below it. *The panel covering the screen transitions in and the one below transitions out.*



Add a script to the canvas called transitionScript.cs.

At the top of the script, import UI and SceneManagement.

```
using UnityEngine.UI;
using UnityEngine.SceneManagement;
```

In the script, create five variables. Two RectTransforms, one for transitionIn and one for transitionOut. Also create a float for transitionSpeed, a Boolean for startTransitionOut and a string for nextSceneName.

```
public RectTransform transIn, transOut;
public float transSpeed;
```

In update, add the following line of code. This code moves the transitionIn panel up when it is blocking the screen.

```
if(transIn.anchoredPosition.y <= 1080) { transIn.anchoredPosition += new Vector2(0, transSpeed * Time.deltaTime); }
```

Create a new public method called TransitionOut. Add a string parameter called newScene. Inside the method, set the startTransitionOut boolean to true and the nextSceneName to the newScene string parameter.

```
public void TransitionOut(string newScene)
{
    startTransOut = true;
    nextSceneName = newScene;
}
```

Back in update, add an if statement that checks if startTransOut is set to true. If it is, move the transitionOut panel up.

```
transOut.anchoredPosition += new Vector2(0, transSpeed * Time.deltaTime);
```

Below this, add another if statement that checks when the transitionOut panel is covering the screen.

```
if(transOut.anchoredPosition.y >= 0)
```

Inside this if statement, do two things. First of all, set the position to be exactly covering the screen (to ensure there is no small gap at the bottom when changing scene). Secondly, change the scene to the scene mentioned in nextSceneName.

```
transOut.anchoredPosition = Vector2.zero;  
SceneManager.LoadScene(nextSceneName);
```

Now, whenever the TransitionOut method is run, it will start a transition to the scene passed into it. This entire canvas object should be on each scene that is being transitioned to and from, creating a seamless wipe scene to scene (as seen in the unity project). *I recommend creating a canvas prefab and putting that in each scene.*