

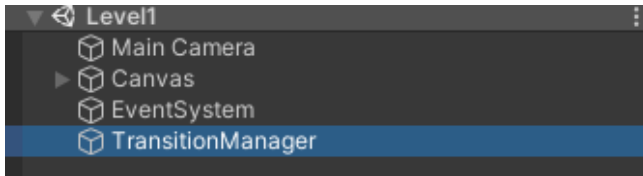
Easy Transitions

Documentation

Overview

Setting up the Transition Manager

First create a new GameObject. To make things more clear lets call Game Object „TransitionManager“.



The second step is to add the **TransitionManager** component to your Game Object. Make sure to do this in every scene you want to load a transition in.

You won't need to assing the **TransitionManager** component to any script. Instead use **TransitionManager.Instance()** to get the instance.

Changing the Scene

To change the scene with a premade or custom transition you can use the **TransitionManager.Instance().Transition()** (*string SceneName, string TransitionID, float LoadDelay*);

function inside the TransitionManager class.

Just call the function from a button or other script and it will change the scene with the given transition.

```
Unity-Skript (13 Objektverweise) | 0 Verweise
public class DemoLoadScene : MonoBehaviour
{
    public TransitionSettings transition;
    public float loadDelay;

    0 Verweise
    public void LoadScene(string _sceneName)
    {
        TransitionManager.Instance().Transition(_sceneName, transition, loadDelay);
    }
}
```

Loading just the transition

To display a transition without switching the scene you can use the

TransitionManager.Transition(*string* TransitionID, *float* LoadDelay);

function inside the TransitionManager class. It's the same as before, just without the sceneName / sceneIndex parameter.

This will display the given transition. To make this useful in a case e.g. displaying a transition whenever the player opens the settings menu you can use the **UnityEvents** inside of the **TransitionManager** class.

```
© Unity-Skript (13 Objektverweise) | 0 Verweise
public class DemoLoadScene : MonoBehaviour
{
    public TransitionSettings transition;
    public float startDelay;

    public GameObject settingsMenu;

    TransitionManager manager;

    0 Verweise
    public void OpenSettingsMenu()
    {
        manager = TransitionManager.Instance();

        manager.onTransitionCutPointReached += ActivateSettingsMenuPanel;

        manager.Transition(transition, startDelay);
    }

    2 Verweise
    public void ActivateSettingsMenuPanel()
    {
        settingsMenu.SetActive(true);

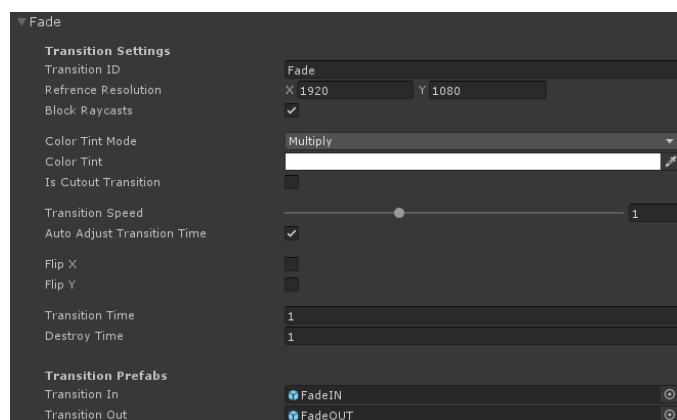
        manager.onTransitionCutPointReached -= ActivateSettingsMenuPanel;
    }
}
```

Add your enable gameobject function to the event and play the transition afterwards.

Make sure to remove the event again to avoid unintended behaviour while another transition plays.

Changing the Transition Settings

To change the premade Transition Manager Settings locate the „Transitions“ sub folder in the Easy Transition folder. All transitions and its transition settings will be located there. You can change them at any time to fit your needs.



Reference Resolution: Is the original resolution of the canvas you made it in. For most transitions this would be **1080p** or **(1920, 1080)** but if you have made your transition with another screen resolution you can change that here.

Block Raycasts: Is for blocking the interaction with the UI on screen while the transition is running. If you set it to **false** you can still interact with the UI on screen, even without having to wait for the transition to end.

Color Tint Mode: Changes the way of the tint mode. **Multiply:** The tint color gets multiplied with the color of the transition. **Add:** The tint color gets added to the color of the transition.

Color Tint: This changes the color of the transition based on the **Color Tint Mode**. If you want to change the color of a black transitions use **Add** mode. If the transition is already white use **Multiply** to change the color of the transition.

Is Cutout Transition: This must be set to true if the transition uses the **CutoutMaskUI** component. Changing the color is disabled when this option is set to true. If changing the color of the transition is needed, directly change it in the transition prefabs.

Transition Speed: Changes the speed of the animator components on the transition. To be able to change the transition speed with this value your transition must contain a **Animator** component.

Auto Adjust Transition Time: Automatically changes the **Transition Time** and **Destroy Time** value based on the set **Transition Speed** value.

Flip X: Flips the transition on the x axis.

Flip Y: Flips the transition on the y axis.

Transition Time: Is the time between the transition start and the scene switch in seconds. Depending on the type of transition you'll have to change this value.

Destroy Time: Is the time between the scene switch and the transition instance getting destroyed in seconds. Depending on the type of transition you'll have to change this value.

Transition In: Is the transition prefab for before the scene switch.

Transition Out: Is the transition prefab for after the scene switch.

Events, Functions and Properties

Transition Manager

Function	Parameters	Description
Transition	<code>string</code> SceneName, <code>string</code> TransitionID, <code>float</code> LoadDelay	This function switches the scene with the given parameters. (Scene Name)
Transition	<code>Int</code> SceneIndex, <code>string</code> TransitionID, <code>float</code> LoadDelay	This function switches the scene with the given parameters. (Scene Index)
Transition	<code>string</code> TransitionID, <code>float</code> LoadDelay	This function only displays the transition.
GetSceneIndex	<code>string</code> SceneName	Returns the index of the given scene name

Property	Type	Description
transitionTemplate	GameObject	This function switches the scene with the given parameters. (Scene Index)
multiplyColorMaterial	Material	(Hidden) The material for the multiply color mode.
addColorMaterial	Material	(Hidden) The material for the add color mode.

Event	Description
onTransitionStart	Gets called when a transition gets triggered.
onTransitionCutPointReached	Gets called when the transition cuts from IN to OUT (cut point).
onTransitionEnd	Gets called when a transition is over.

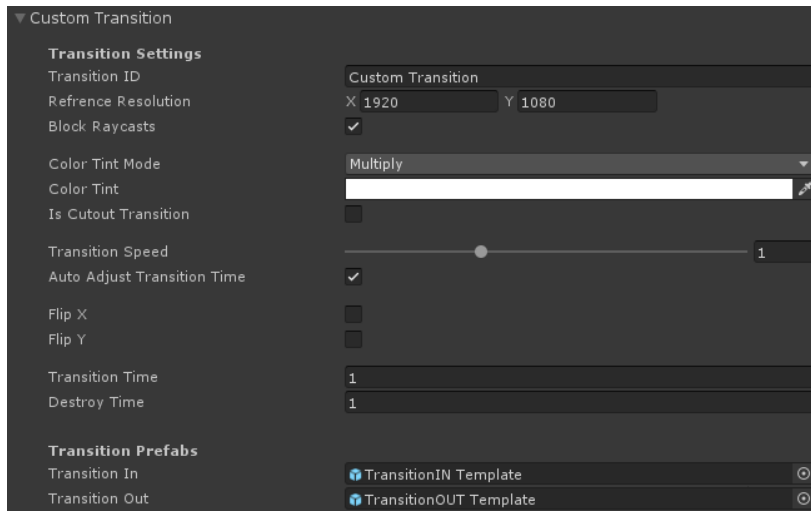
Transition Settings

Property	Type	Description
referenceResolution	Vector2	This defines the resolution of the canvas the transition was made in. Normally this is (1920, 1080) but this could change for some transitions.
blockRaycasts	bool	If this bool is set to true, any interactions with the UI on screen will be blocked until the transition is over.
colorTintMode	ColorTintMode	Sets the color mode to either multiply or add.
colorTint	Color	Changes the color of the transition based on ColorTintMode .
isCutoutTransition	bool	Set to true if there is a CutoutMaskUI component on the transition.
transitionSpeed	float	Changes the speed of the Animator components on the transition.
autoAdjustTransitionTime	bool	Automatically adjusts the Transition Time and Destroy Time values based on the Transition Speed value.
flipX	bool	Flips the transition instance on the x axis.
flipY	bool	Flips the transition instance on the y axis.
transitionTime	float	The time in seconds between the transition start and the scene switch.
destroyTime	float	The time when the transition object gets destroyed after the scene load.
transitionIn	GameObject	This variable stores a prefab of the In animation of the transition.
transitionOut	GameObject	This variable stores a prefab of the Out animation of the transition.

Included Transitions

Transition ID	Description
Fade	A simple black screen fade transition.
CircleWipe	A simple circle wipe transition.
LiniarWipe	A simple liniar wipe transition.
RectangleGrid	A more complex transition using many rectangles on a grid.
DoubleWipe	Similar to the liniar wipe transition but using two rectangles.
DiagonalRectangleGrid	Similar to the rectangle wipe transition but diagonal.
RectangleWipe	Similar to the circle wipe transition but using a rectangle instead of a circle.
VerticalCurtain	A simple vertical curtain transition.
HorizontalCurtain	A simple horizontal curtain transition.
Brush	A more complex transition where a brush paints across the scene.
PaintSplash	A more complex transition with many paint splashes on the screen.
Noise	A simple noise dissolve/resolve transition.

Custom Transitions



To add a custom transition locate the „**New Transition Settings**“ option under Create -> Florian Butz -> **New Transition Settings**. You can access this menu by right clicking in your Project Browser. This will create a new file with all the settings for your transition.

Change the **Reference Resolution** to the resolution the canvas you made your transition in. If you made the transition with the HD (1920, 1080) resolution you can leave this setting at its default values.

Change the **Transition In** and **Transition Out** with a prefab of an animated image. Make sure to not have a canvas in your prefab but only the image.

Change the **Transition Time** to the time in seconds where the cut (scene change) of your transition should happen. And the **Destroy Time** to when the transition instance should get destroyed after switching the scene in seconds.

The other settings are not that important for the beginning. You can play around with them to perfectionate your transition.

To use your custom transition just call the **Transition()**; function from the **TransitionManager** instance. Make sure to set the **Custom Transition** scriptable object to your newly made one.

```
0 Verweise
public void LoadScene(string _sceneName)
{
    ...
    TransitionManager.Instance().Transition(_sceneName, customTransition, startDelay);
}
```