

Mask Transitions

Simple Shape Mask Transitions for your games.

DOCUMENTATION

Version: 1.0

Author: Nearmint Studios

Last Updated: 10 September 2024

TABLE OF CONTENTS

INTRODUCTION	2
HOW TO USE	2
Drag And Drop	2
Transition With Scene Load.	3
Transition Without Scene Load.	3
Individual Transitions	4
CUSTOMISATION	4
TROUBLESHOOTING	5
SUPPORT & FEEDBACK	5
CHANGELOGS	5

INTRODUCTION

Mask Transitions is a sleek and simple solution for implementing cutout mask transitions in your game. It allows for fully *customizable transitions*, where you can easily change the color and sprite to suit your game's aesthetic.

Additionally, the *Transition Manager* is designed with flexibility in mind—feel free to modify or extend it to fit your specific use case.

Key Features

- **Sprite and Color Flexibility:** Freely change the transition sprite and color to match your game's visual style.
 - **Scene Transition Support:** Seamlessly load new scenes while playing transition animations.
 - **Partial Transitions:** Option to play only the first or second half of a transition, offering more control over the flow of your animations.
 - **Easy Integration:** Simply drop the asset into your project and start using it with minimal setup.
-

HOW TO USE

A. Drag And Drop

Locate the '**Transition Manager**' prefab in the imported asset. Here's the file location:

```
Assets > Nearmint Studios > Mask Transitions > TransitionManager.prefab
```

Drag & Drop this prefab in your scene.

NOTE: The *Transition Manager* has already been added to the *DontDestroyOnLoad* list. Add the prefab to the first scene that loads in your game.

B. Include MaskTransitions namespace.

In the script you wish to call a Transition, declare the Mask Transitions namespace:

```
using MaskTransitions;
```

B. Transition With Scene Load.

The *Transition Manager* will handle both loading your scene and playing the transition. Here's how you can load your preferred scene with a transition from your desired script:

```
0 references  
public void SwitchScene()  
{  
    ...  
    TransitionManager.Instance.LoadScene(sceneToLoadName);  
}
```

In the example above, *sceneToLoadName* is a string that contains the name of the scene we desire to load.

Optionally, you can add a delay before the scene is loaded.

```
0 references  
public void SwitchScene()  
{  
    ...  
    TransitionManager.Instance.LoadScene(sceneToLoadName, 0.5f);  
}
```

C. Transition Without Scene Load.

The *Transition Manager* will handle both loading your scene and playing the transition. Here's how you can load your preferred scene with a transition from your desired script:

0 references

```
public void PlayTransition()  
{  
    TransitionManager.Instance.PlayTransition(totalTransitionTime);  
}
```

Here, *totalTransitionTime* is a variable that defines the total time the transition should take from start to finish.

Optionally, you can also pass a delay variable to add some delay before the transition starts.

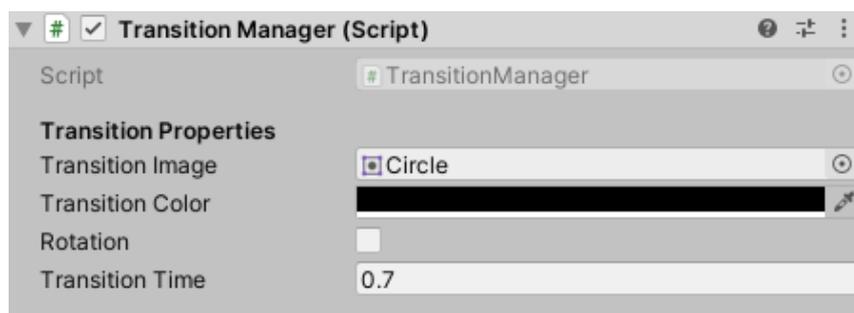
D. Individual Transitions

The asset provides you the flexibility to play the start and end of the animation individually.

```
1 reference
void PlayStartOfTransition()
{
    ...
    TransitionManager.Instance.PlayStartHalfTransition(totalTransitionTime);
}
1 reference
void PlayEndOfTransition()
{
    ...
    TransitionManager.Instance.PlayEndHalfTransition(totalTransitionTime);
}
```

CUSTOMISATION

The asset allows you to change certain aspects of the transition to customize it to your liking. Locate the Transition Manager script on the Transition Manager prefab.



- Transition Image:** By default, the asset plays the *circle transition*. But, you can choose to add any sprite of your choice to replace the circle transition.
- Transition Color:** Easily change the color of your transition by changing the color variable in the Transition Manager.

- c. **Rotation:** Determines whether your image component rotates during the transition. Since the default transition is a circle, it doesn't make sense to rotate it. But, if you decide to change the Transition Image, turning on the rotation variable can provide a more dynamic-looking transition.
 - d. **Transition Time:** Finally, the transition time variable determines how long your transition takes to complete. This does not include the time it takes to load your scene.
-

TROUBLESHOOTING

- a. If you get a **warning** saying, *"There are 2 event systems in the scene. Please ensure there is always exactly one event system in the scene"*. The TransitionManager prefab already contains an **EventSystem**. And since the game object will not be destroyed when other scenes are loaded, you can remove any existing event systems from your scenes.
 - b. If you get a **null reference exception** at the line of code where you call your transition from the TransitionManager, make sure you've added the TransitionManager to your scene correctly.
 - c. **ERROR:** *The type or namespace name 'DG' could not be found* If you encounter this error as soon as you import the asset. It means that you don't have the Dotween asset imported into your project. This asset animates sprites using dotween to play transitions. You can add DoTween to your library for FREE on the asset store:
<https://assetstore.unity.com/packages/tools/animation/dotween-hotween-v2-27676>
 - d. To test the **'Switch to Scene'** demo button in the demo scenes. You must add both the demo scenes to your build settings.
-

SUPPORT & FEEDBACK

If you encounter any issues or have suggestions for improvements, feel free to reach out:

Email: admin@nearmintstudios.com

Please do leave us a review on the Unity Asset store if we can help improve your game development experience. 😊

CHANGELOGS

- Version 1.0: Initial release of the Mask Transitions asset.
-

LICENSE

You are free to use this asset for both personal and commercial projects. You are allowed to:

- Modify the asset to suit your needs.
- Use it in any type of Unity project, including commercial products.
- Distribute and share the asset as part of a larger project.

You are not allowed to:

- Resell or redistribute the asset as-is or as part of a collection where the asset is the primary value.

Attribution is appreciated but not required.