

Transitions Manager by DFT Games

Features

- Splash screens display
- Story display
- Game pause
- Scene loading
- Works on Mobile, Standalone and Web apps
- Takes care of web streamed scenes loading delays

How to use it

This class is a Singleton one which will not be destroyed loading new scenes. Basically the best way to implement it is to create a start scene having just the main camera and an instance of the Transition Manager prefab. This scene should be your first one: in the build settings move it on the top as scene 0.

Splash screen and story textures

Read this part very carefully. To reduce the memory footprint we assume that your content textures are transparent and separate from their background. So, while we scale the backgrounds to fit the screen, we scale proportionally the content (logo or story), so avoiding to create a texture for each screen resolution. You can also decide not to scale the content texture un-checking the script parameter **Scale Content Textures**.

Important: Remember to set the Texture Type to **GUI** in the inspector! Another important thing to remember is to properly set the **TexturesNativeResolution** property: this tells the script the original intended resolution for your logos and story textures.

The Splash Screens can have a background texture: if they are not supplied then the Background Colour parameter is used; the Story Screens use only the Background Colour parameter. You can feed the Story textures at runtime as well.

Delegates

To simplify your task and make the app swift and light we expose to your scripts two delegates: **PauseMenu** and **LoadingLevel**. The best way to use these is to write your custom code to display a pause menu and/or the loading screen (for streamed web scenes) in a static class and assign the methods to the corresponding delegate. The script checks if the delegates are null or not: if they are assigned then will execute the delegate instead of the default code.

Methods and properties exposed to your scripts

From any script you can call:

TransitionManager.Instance.WaitingStreamedScene is a read only Boolean value. When using your own loading routine (assigning the delegate) you can poll this property to know if we are loading a streamed scene.

TransitionManager.Instance.LoadingProgress is a read integer value. When using your own loading routine (assigning the delegate) you can poll this property to know the loading progress expressed as percentage (only if we are loading a streamed scene).

TransitionManager.Instance.ShowStory(nextScene) to show the story screens: if nextScene is not empty it will be loaded after the last story screen had been faded out.

TransitionManager.Instance.LoadScene(SceneToLoad, fadeOutFirst) to load a new scene. If the second parameter is True then the current scene is faded out before to load the requested one.

TransitionManager.Instance.ClickThroughStory is a List of Textures. You can feed this list at runtime as well to show a click through story.

TransitionManager.Instance.TimedStory is a Boolean value (default is False). If you set this True the Story pages will go ahead using the splash screen timing rules.

TransitionManager.Instance.CanSkipSplashWaiting is a Boolean value (default is True). Setting this False will make the user unable to skip the splash screen/timed story waiting time. The default behaviour allows to skip the waiting time by touching any key, clicking or (on mobile) touching the screen.

TransitionManager.Instance.DetectPauseRequest is a Boolean value (default is False). Set this True to let the Script manage the Pause requests.

Support

In case you need any help implementing our Kit do not hesitate to write to pino@dftgames.com

Thank you for using our Transitions Manager! **Don't forget to add your review on the Store!**

COMMERCIAL GRADE FREEBIES
by DFT Games