**Impulse Framework**

This barebones framework is designed to help you expedite the creation of your next idea in Unity. It includes a number of highly customizable prebuilt systems such as a scene manager, mobile-optimized main menu, cameras, and more.

**Initial Setup**

If you choose not to use prebuilt managers that come with the framework, feel free to skip this section. However, if you do not already have a solution in mind for switching scenes, setting up the main menu, or handling music, it is highly recommended that you do this initial setup.

In the Build Settings, set the Splash scene to 0 and Menu to 1. Unity preloads everything in each scene, with the exception of the first scene (scene 0). For optimal performance, you should keep your splash scene as lightweight as possible and try not to add too many more objects.

**Customizing the Splash Image**

A video demonstration of setting the splash image, setting up scene loading, and customizing the menu can be found here: <https://www.youtube.com/watch?v=btNqHCoRwB8&list=PLLXw4Fw6qNw5WVLPn1hhJNEcwXjxt3b9j&index=1>

1. Locate the Resources/Prefabs/Scene/SplashFadeIn object in the project files (not in the hierarchy!). Select the ImageToFade child object.
2. Set the Source Image of the Image component to whatever splash image you want to display.

**Customizing scene loading (fade in/out, interpolation, duration)**

1. Locate the Resources/Prefabs/Scene/SceneManager object.
2. In the SceneManager component, you can specify the Duration of fade in/out as well as the Interpolation of the fade. If you do not want to fade in/out scenes, set the duration to 0.

**Customizing the main menu**

1. Open \_Scenes/Menu.unity
2. Open the MenuSystem object. You’ll notice a main menu and options menu are already set up for you, but are inactive.
3. Create a new child object under MenuSystem and attach the MenuScreen script to it.
4. Add your new menu elements to this new child object.
5. Set your new child object as inactive once you are finished with it.

To switch menus using UGUI OnClick(), call the MenuManager.ChangeMenuAndFade() or MenuManager.ChangeMenu() function.

**Using the music manager and music playlists**

For a video demonstration of the music manager and music playlists: <https://www.youtube.com/watch?v=jQGTqGalGVw&index=2&list=PLLXw4Fw6qNw5WVLPn1hhJNEcwXjxt3b9j>

1. Drag the MusicManager prefab from Assets/Prefabs/Music/MusicManager into your splash scene, or whichever scene is the first one in your build settings. The MusicManager is persistent from scene to scene, so you do not need to instantiate it in each scene.
2. In each scene where you want music to be played, create a new empty game object and attach the MusicPlaylist.cs script. This script can be found in Assets/Scripts/Music/MusicPlaylist.cs. I recommend naming the game object ‘MusicPlaylist’. Then, just populate the Music List array in the game object with song files. Leave ‘Activate On Awake’ to true if you want the playlist to begin playing as soon as the scene is loaded.

**Using the Top-Down Camera**

For a video demonstration of the top-down camera: <https://www.youtube.com/watch?v=DLTyrbMxytA&list=PLLXw4Fw6qNw5WVLPn1hhJNEcwXjxt3b9j&index=3>

1. Locate the script in Assets/Scripts/Camera/TopDownFollow\_Camera.cs.
2. Attach this script to a camera object in your scene.
3. Drag a Transform into the Follow Target parameter. This is the object the camera will try to follow.
4. Set the Target Offset and Move Speed parameters to your liking. Target Offset is x,y,z distance from the follow target (the camera position offset relative to the follow target object). Move speed is how fast the camera moves when the object moves.