Documentation - Quick Guide

Essentials

Summary

Guillem Poy

@guplem

TriunityStudios.com

This file is meant to describe the main features included in the asset.

All the code is commented with XML documentation and **full code documentation of the latest version of the asset exists here**, so this file does not go in deep in every class implementation and neither how to use them.

If you want to see examples, know what each method does, most of the class functionality and how to use most of the asset's features, I recommend you to look into the "Examples" folder (and read the comments inside the "Essentials/Scripts" folder).

Essentials' Settings and Adjustments

This window can be found under the "Window/Essentials/Settings" menu. In there you can apply adjustments to your project.

Additionally, you can add custom adjustments simply by programming them in a class that inherits from the abstract class "Adjustment".

Save Data

The class "SaveDataManager" allows you to save, load and delete data stored in JSON format consistently. It should be able to save data cross-platform without any tweaks.

Its use is very simple by just taking advantage of the Methods "Save", "Load", "Delete", "DeleteAll" and "Exists".

To open the folder containing the saved data, you can use the menu item: "File/Open saves data folder".

An example can be found in the scene "SaveDataExample" and a video hosted on YouTube exist as a demonstration of the feature.

Debug Essentials

Using "DebugEssentials" you can easily debug IEnumerables (lists, arrays, ...) using the method "DebugEssentials.LogEnumerable(...)". Example: DebugEssentials.LogEnumerable(Array).

In addition, it implements all the functionality from the Unity Engine's "Debug" class.

You can see an example of its use in the scene “DebugEssentialsExample” which uses the code “DebugEssentialsExample.cs”.

Pool System

The Pool class allow you to easily improve the performance of your game avoiding the constant instantiation and destruction of objects by automatically enabling and disabling them with a limited number of instances.

You can see an example of its use in the scenes and scripts inside the “Essentials/Examples/Pool” folder.

Simple Animations

The SimpleAnimation class in conjunction with the SimpleAnimationsManager allows you to easily animate GameObject’s components without having to deal with the extensive world of Unity’s animations.

You can see an example of how an animation of a GameObject’s transform could work in the scene “TransformAnimationExample” which uses the code “TransformAnimationExample.cs” and in the scene “RectTransformAnimationExample” which uses the code “RectTransformAnimationExample.cs”. These examples use the class “TransformAnimation” and “RectTransformAnimation”, which can be taken as an example to create animations of any component following the same patterns shown in it.

Other examples can be found in the same folder and a video hosted on YouTube exist as a demonstration of the feature.

Random Essentials

The RandomEssentials class allows you to easily work with randomness. It inherits the “System.Random” class but integrates part of the functionality of the “UnityEngine.Random” class.

So, it contains the same functionality plus some methods that are easier to understand than the original ones and some more that add new functionalities.

You can see an example of its use in the scene “RandomEssentialsExample” which uses the code inside the file “RandomEssentialsExample.cs”.

Extensions

Multiple class extensions have been added to the asset to add some commonly-needed functionality in video games into some already implemented classes.

Some of the extended classes are:

- Camera
- Component
- Debug
- Float
- GameObject
- ICollection
- IEnumerable
- Int
- LayerMask
- Mathf
- Rect
- RectTransform
- SerializedProperty
- String
- Transform
- Vector
- VectorInt
- Etc

Flow Control

C# classes have been added to control the flow during the execution of the project. This classes are:

- DoN (and DoOnce): Allows the call of a function N (or one) time even though they are called more times.
- Sequence (and FlipFlop): Executes one of the linked functions every time it is called rotating between them in sequence or randomly.

You can see examples about how to use them in the folder “Essentials/Examples/FlowControl”.

A video hosted on YouTube exist as a demonstration of the feature.

Preset tools

3 tools have been added to increase the functionality of the presets:

1. A tool to know if all Game Objects of the open scene match a preset (if they contain the component the preset is meant to set). It can be activated from the context menu of a preset.
2. A tool to make all Assets match the presets contained in the same folder. It can be activated using the Essentials Settings window.
3. A tool to check if all Game Objects in the open scene match the default presets. It can be activated through the GameObject menu in the Hierarchy or the menu items at the top of the engine.

A video hosted on YouTube exist as a demonstration of the feature.

Audio Source Manager

It is a component that allows the user to play as many audio clips as they want without worrying about the audio sources that they have set up.

It can play and stop clips in addition to allow access to the audio sources that host them so their configuration (volume, pitch, ... can be updated).

A video hosted on YouTube exist as a demonstration of the feature.

Editor Dividers

A tool that allows the creation of consistent dividers in the editor's hierarchy.

To use this feature, you have to simply right-click on the hierarchy, choose “Create Divider”, configure it in the appearing window and hit “ok”. It will create an “EditorOnly” GameObject with the desired name to divide the different types of GameObejcts.

Console - In builds (in-game)

A very easy-to-use and set up console that is handy to the developers and testers specially while running builds.

It is easily toggled on and off, and supports most UI systems.

It allows the automatic saving of the logs in a file in a dynamic folder inside the desktop named “Unity_Console_Logs”.

Scripting with it is easy working with the open class “Console” which is fully documented.

A demo for the feature exist as an example in the scene “ConsoleExample” and a video hosted on YouTube exist as a demonstration of the feature.

Other Features

Shortcuts

- Save Scene and Project (Ctrl + Shift + Alt + S)
- Save project (Ctrl + Alt + S)
- Clear Console (Ctrl + SPACE)

Menu Items

- File/Save Scene and Project