SPLITSCENES

User Documentation v2.4
1/22/2023

CONTENTS

WELCOME

Thanks for checking out **SplitScenes**! Although there are a few other additive scene management tools on the Asset Store, and there are a few dynamic loading systems, we do think SplitScenes offers more usability, simplicity, and features out of the box than the alternatives—especially at our price point.

Due to time constraints and the effort of maintaining the many incremental changes that Unity goes through, we no longer support non-LTS releases. There are just too many random bugs in all the little versions of Unity between LTS releases that it wastes too much time to keep them all running in the same codebase. I hope you understand and enjoy our products with Unity's most stable releases.

Although it has grown since its inception, the concept was initially to enable a better team workflow where a single root scene could automatically manage a set of other scenes **like a Photoshop file has Layers**. And this works pretty well.

A number of users have asked if there was a way to do dynamic scene transitions for big worlds, or set up asynchronous loading zones for hallway shooters. This most recent release can handle this and much more. A basic example of both of these workflows is included in this asset.

Finally, the Scene Locking feature has been extended to take advantage of the Unity 2019 option to disable object picking in scenes. What this does is prevent you accidentally selecting objects in scenes you don't intend to modify, but additionally Locking detects whenever a scene is modified unintentionally and prompts you before saving it or discarding it accidentally. On a team where artists and gameplay designers work together, it's easy to get things tangled up, but with SplitScenes it's really easy to keep straight. **No more merge conflicts!** We use SplitScenes internally for our own products, and am happy to fix bugs and add features as they become apparent. Please reach out if you have questions or run into problems.

If you like the asset, *please* do take a moment to write a positive review or give it a ★★★★★ rating. Be sure to recommend it to your friends. Building assets for the Unity developer community takes a lot of time and effort, so be sure to support the ones that make your life easier. Otherwise, we won't be able to keep doing it for you.

The URL to leave a review is right here:

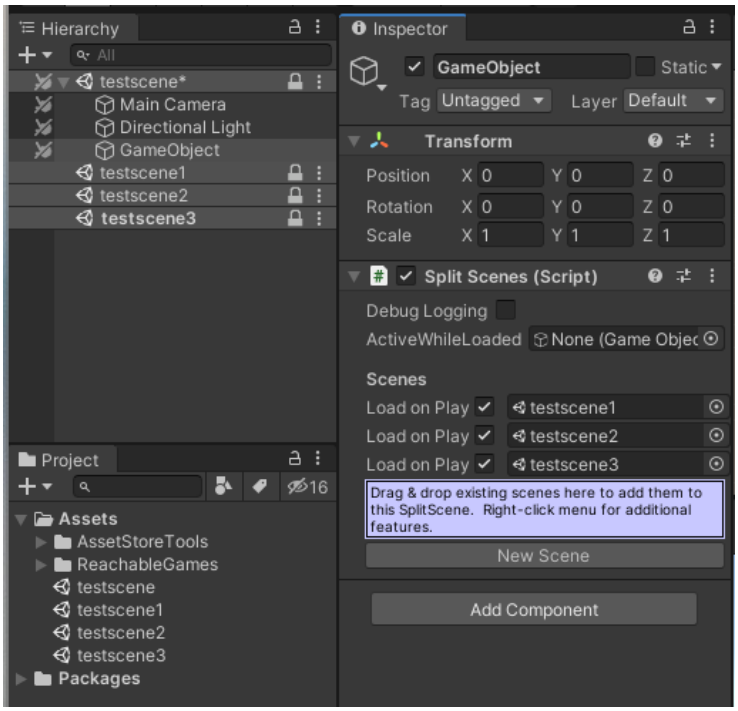
https://assetstore.unity.com/packages/tools/utilities/splitscenes-133941

INTRODUCTION

Although the scene system in Unity is powerful, Unity does not come with a good scene management interface. Especially when considering team member specializations, there are logical partitions you can make that allow artists, designers, and programmers to work independently in the same space. However, doing so in Unity's default interface is a very tedious process.

SplitScenes makes working in multiple scenes as easy as using Photoshop layers. The 30 second walkthrough:

1. New scene.
2. Create Empty.
3. Add a *SplitScenes* Component.
4. Save the scene as "testscene".
5. You can create more scenes rapidly by clicking the New Scene button on the SplitScenes custom inspector.
6. Drag and drop existing scenes onto the colored text box to add them quickly to the list.
7. Right-click scenes in the inspector to rearrange the order scenes will load at runtime or delete them from the list.



Notice the new Lock icon on the hierarchy window? This is the fancy new button that helps you control which scenes are modified and which are not. When locked, a scene is defaulted to *unpickable* so you won't accidentally grab objects and move them in the Scene View. You can toggle that back on if you want, or just pick objects directly in the Hierarchy Window, but the whole point is for you to not accidentally change things. Unlock the scene to do that. It's very common to go do work on a large scene and mess with something unintentionally, and just save everything because it's hard to know what changed. By making it a conscious decision, you are always aware if your work may impact someone else. Don't worry, you can always save changes you meant to make, even if the scene is locked, but SplitScenes does ask you first. I recommend you try it out for a bit, and get used to it, though. All scenes default to locked, even new ones, but it's only one click to unlock that scene and your settings are stored, so the effort is incredibly minimal. But if it's not for you, you can turn off the Lock functionality entirely by toggling Tools → ReachableGames → SplitScenes → Use Scene Locking. It does not affect the SplitScenes functionality in any way.

When you add a scene to a SplitScenes object, it defaults to "Load on Play". That means it will be loaded when the SplitScenes object is activated at runtime. In order for that to work, though, all these scenes must be added to the Build Settings scene list. This is done automatically for you, so you don't have to remember to manage that.

If you don't want some of the scenes to load when you hit Play, uncheck the "Load on Play" toggle. If you've divided up the scenes well, you can use this to quickly check out the lighting or environment without the enemies showing up, or add in a debug scene that gives extra information that you want hidden in the final build. This feature opens a lot of possibilities!

My personal suggestion for scene organization should look something like this:

1. Root Scene (with a single SplitScenes object in it, and a single light probes set)
    a. (1 or more) Static Level Geometry, Static Props, etc
    b. (1) Static Collision
    c. (1) Baked Static Lights
    d. (1 or more) Realtime Lights
    e. (1 or more) Dynamic Props
    f. (1 or more) Gameplay Scripts (enemy spawners, triggers, etc)

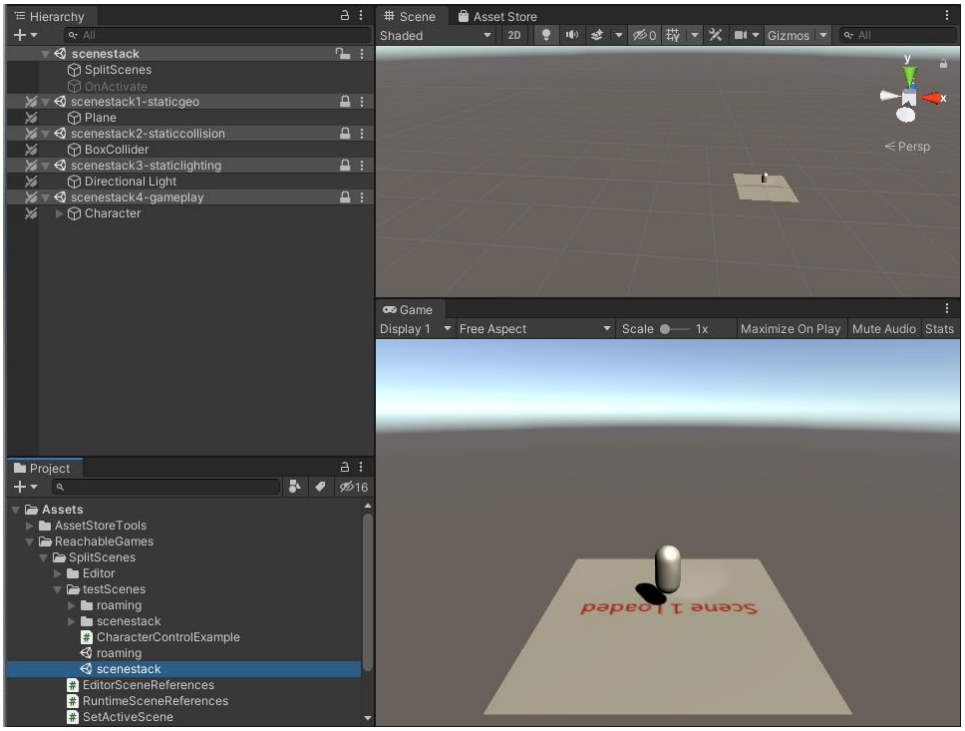
FIRST STEPS (BORING, BUT RECOMMENDED)

This document is the best place to learn about the features and uses about this asset. Please read it thoroughly, as we want to make sure you get the most out of your investment.

1. Create a ***brand new project***. This may seem unnecessary, and while we go through great lengths to make sure there are no collisions with other assets, it is impossible to provide a guarantee for every combination of settings. Overall, this will give you the best possible experience to first figure out how to use the asset without introducing other variables.

2. Install **SplitScenes** from the Asset Store.

3. Open the provided test scene called *splitscenestack*. All our assets come with example data specifically for trying them in safe isolation. (Later, when you install it into your project, everything under the *testScenes* folder may be removed.)



4. Notice how *scenestack1*, *scenestack2*, *scenestack3,* and *scenestack4* are automatically loaded? This is because they are already listed in the *SplitScenes* object in *scenestack*. The functionality of the component is to make sure the **managed scenes are loaded in the editor or at runtime automatically according their settings.** These are easily accessed on the custom inspector for the SplitScenes component.
5. If you toggle off the SplitScenes component or disable its GameObject, the script instantly removes all the managed scenes from the Editor for you. However, it's possible to have several SplitScenes objects in a scene at once, and they may have overlapping lists of scenes. **This is how you can easily build large worlds!** A simple example of this is shown in the *roaming* scene.

6. Turn the script back on, then right-click one of the scenes in the Hierarchy Window and select Unload. This built-in Unity mechanism takes the scene out of memory, but leaves it in the scene list.
7. Right-click another scene and Remove it from the Hierarchy Window.
8. Press the Play button. Notice that all of the scenes are loaded and activated in sequence. Also note that after the scenes are all loaded, the bottom-most scene is highlighted? That is because an object with the *SetActiveScene* script was connected to the **ActiveWhileLoaded** field. Whenever all the scenes are loaded, whatever GameObject you connect there will be enabled the next frame so you can initialize things at the right time. It is also disabled just before scenes get unloaded, in case you need to save some state.
9. Press the Stop button. Notice the scene you Unloaded goes back to that status, but any missing scene is loaded again. So, if some scene is really heavy, rather than just making it invisible with the eyeball button, you can just Unload the scene, which removes it from memory entirely.
10. Open the *roaming* scene now.
11. Press Play.
12. Walk around with WASD. Notice that there's nothing visible in the scene when you started except a character and some bounding box triggers? This demonstrates multiple overlapping scenes that allows your character to trigger the progressive loading of parts of a level with minimal effort. Pretty sweet!

IMPORTING INTO YOUR PROJECT

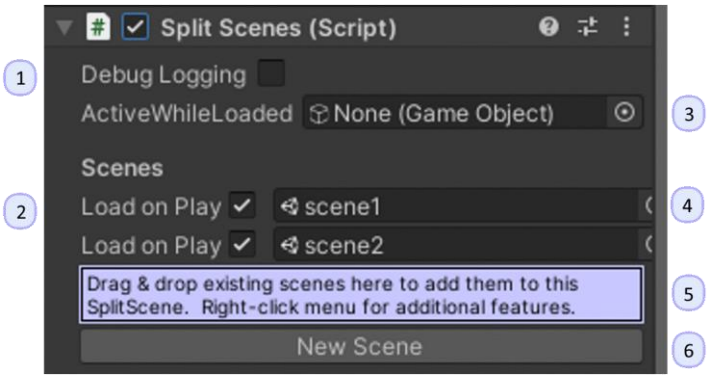
To set up **SplitScenes**, do the following:

1. Import **SplitScenes** asset package
2. Create a new scene to be the root of a set of scene layers
3. Typically, delete the default objects in the root scene unless you want them there.
4. Click *Add Component* and select *SplitScenes* to create a management object in the root scene.
5. Click the *New Scene* button or drag in existing scenes
6. Remember to load the root scene to see the whole set at once, although you can simply open and work on a sub-scene anytime.

FEATURES

SplitScenes is a great tool for getting teams to work together without worrying about stomping each other's work. It also helps with revision control merging problems by separating out content into different scene files that can be organized by content type or individual contributor or whatever method suits your workflow. Something similar could be done with prefabs in a scene, but not with the convenience or clarity of additive scenes, nor with the flexibility of dynamic progressive scene loading. With a scenes-as-layers workflow, you can easily build levels where different people can be working simultaneously and constantly sync each other's work, without worrying about conflicts—*live*. That's the holy grail.

HOW TO USE THE GUI



Although the custom inspector for this tool is relatively simple, here's a thorough explanation of each feature you can see, plus a few you might have to hunt for.

1. **Debug Logging**: If you are having any problems with SplitScenes, check this box and you will see every action it takes and exhaustive details about what phases Unity is operating in during transitions. This is probably not useful information except if you want to submit a bug report, then please do turn this on and send a log.

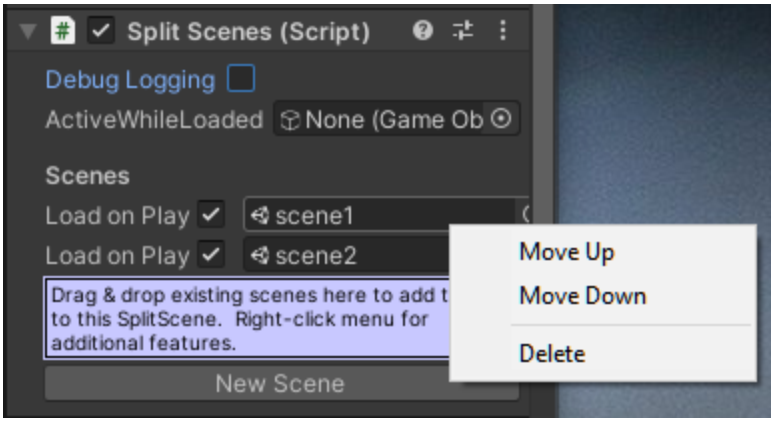
2. **Load on Play**: If checked, the scene on this row will be loaded at runtime. If unchecked, this scene is only loaded in the Editor. This is useful for situations where you have reference material or associated concept art that is needed during development only. You can also use this to have debugging tools in a scene that can quickly be included or excluded with a single checkbox. Also useful if you place your enemies in one scene and your environment in another, so your artists can toggle off enemies while inspecting their work. And dozens more purposes…

3. **Active While Loading**: You can create a GameObject, connect it here, and when the full set of scenes are loaded, it will be activated, giving you a chance to initialize things. When the set of scenes is unloading, it will be deactivated first, allowing you to save any state or perform transitions you need. Just remember to **leave it disabled** otherwise it will trigger immediately on start, which defeats the purpose.

4. **Scene selector**: This conveniently accepts SceneAsset types. Be aware that in Unity, all scenes are actually loaded by path or index. SplitScenes loads everything by path, since the Build Scene list changes all the time. **If you change the path of a scene**, whether dragging or renaming, make sure you open any SplitScenes object that references it. The path is automatically updated under the hood when it activates, so just clicking on it should suffice.

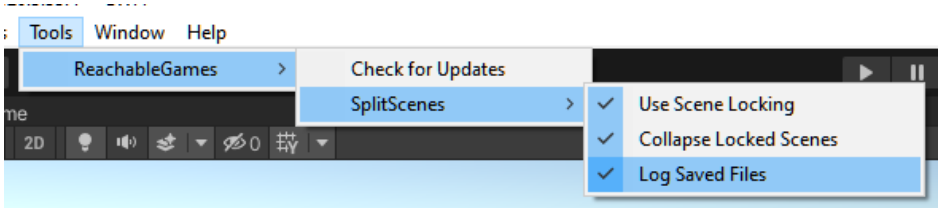
5. **Add-by-drag-and-drop**: To add an existing scene to a SplitScenes, just drag it to the blue square. The inspector is smart enough to reject duplicate scenes or non-scene objects. This will add the scene to the bottom of the list. You can adjust the order with the Right-Click context menu.

6. **New Scene**: To add a new scene quickly, just click this button. It automatically names the new scene, saves it to disk, and adds it to the list. Very convenient.



7. **Right-Click Menu**: To arrange the order of scenes, just right-click anywhere on that row and select Move Up or Move Down. To remove that entry, select Delete. Don't worry, Delete only removes the scene from this list, no actual scene files are harmed.

TOOLS MENU



8. **Check for Updates** allows you to quickly pop up a browser window that shows what the latest versions are for all the Reachable Games tools, including SplitScenes. It's also a shortcut to our website and blog, which I promise is as thrilling as it sounds.

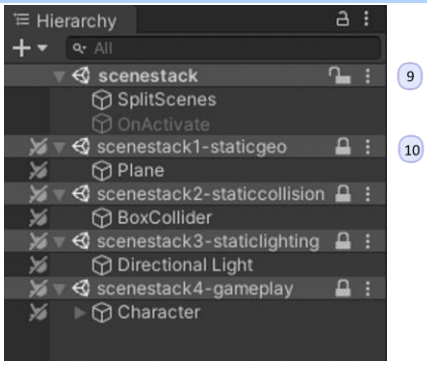
9. **Use Scene Locking** allows you to completely disable the Lock icon behavior added by SplitScenes. Although we think it's pretty great, you may disagree. We can still be friends.

10. **Collapse Locked Scenes** is what happens when you have 10 or so scenes with a thousand objects in them. It gets really noisy, and it turns out that when you've locked a scene, you

probably don't need to see the contents of those scenes on startup. But the **unlocked ones** do matter to you. This setting helps by quieting down the noise by collapsing any scene you left locked. Quality of Life is important.

11. **Log Saved Files** is a handy tool to see what files are being saved. Sometimes it's useful to glance down and see that you actually did save a scene or did not save a scene that you had locked, and navigated the popups properly. But honestly, more than anything, it helps track down badly-behaved assets (not mine!) that keep modifying files at random times. Feel free to toggle this off if you don't find it useful.

HIERARCHY WINDOW



12. **Unlocked**: This scene's contents are allowed to change, be picked in SceneView, and saved without alerting you. This setting persists across sessions so you don't need to constantly unlock the same scenes.

13. **Locked**: This scene's contents cannot be picked in the SceneView, and if it does somehow change, you will be alerted if you try to save the contents. You can still save, but you have the option to ignore the save operation for this scene. Also, if you have a modified scene and try to Lock it, you will be asked whether you want to save the scene before locking it, or discard the changes and revert back to whatever is on disk.

MULTI-SCENE ACTIVATION

With *SplitScenes*, all the sub-scenes are loaded asynchronously at the same time, which may cause them to complete in random order. Unity provides a method for delaying the activation of a scene, and we take advantage of this to ensure none of the scenes are activated until they are all done loading. The previous version of SplitScenes had trouble activating all the scenes at once, but this new version **triggers activations of all the scenes in order, on the same frame!** You can verify this by looking at the Console output on the *scenestack* example scene when you hit Play. For now, though, consider using a loading screen if you are likely to load scenes directly in front of the player camera.

Worth noting, however, is that objects in each scene go through the Awake→OnEnable→Start process together, and scenes activate one after another. So you may find that code that is dependent upon scripts or objects in other scenes could fail to start correctly. All I can suggest is put related objects in the same scene, reorder scenes so the objects doing the Finding come later (this is fragile), or make use of the **ActiveWhileLoading** object's **OnEnable** function to do cross-scene hookups.

REFERENCE COUNTED SCENES

The *roaming* example demonstrates a very simple trigger volume-based scene loading system. This works because the set of scenes loaded at any given time is reference counted, meaning multiple SplitScenes objects can ask for a scene to be loaded and as long as at least one remains, the scene does not get unloaded. Feel free to come up with more clever ways to use this for open world

exploration games, or portal-loading methods in shooters, etc.

ALL SOURCE IS INCLUDED

If you want to change the way it works, knock yourself out! It's a great starting point for building your own multi-scene workflow. Please do not expect support for any code changes you make.

SUPPORT

Please read all the documentation. We put a lot of effort into it, and hope that it exceeds your expectations. In the event you have further questions, please check the following web pages for more details about this asset:

Our Website: https://reachablegames.com/unity-assets/

Unity Forum: https://forum.unity.com/threads/624757/

If you find that none of the above can answer your question, you may contact us at support@reachablegames.com, but know that we always handle support requests first that include:

1. **Invoice Number**
2. **Unity Version**
3. **Asset Version**
4. Links to screenshots or small sample projects on DropBox or similar sharing site describing the problem.
5. Kindness. If you are mean, rude, harassing, or hateful, do not expect a response.

As a matter of common sense, we do not offer support to free customers except as time permits.

Finally, if you are looking for a feature that is not currently supported, understand that we are a business and get many such requests. If you need a custom feature that is important enough to your product to pay for its development, contact us about it.

ABOUT US

Reachable Games is located in the beautiful hill country of Austin, Texas USA. It was founded by Jason Hughes, who has been a professional game developer since 1995, at Origin Systems. He has worked on many AAA games and with many recognizable companies. As a generalist, Hughes has worked on nearly every kind of game platform in every capacity—from graphics tools to AI to UI/UX to game design to shader writing to database management to networking and server development. It turns out this is the right skillset to help improve the Unity development experience for other developers.

We currently have several other assets on the Asset Store. For what it's worth, they are all built because we are working on projects of our own and both need and use them on a daily basis. If you think this one is great, chances are the others will help you out too.

RELEASE NOTES

V2.4 JANAURY 22, 2023

- Added auto-collapsing locked scenes. QoL improvement for large teams.
- Added logging of files as they save.
- Fixed a couple of minor bugs with renaming scenes.
- Removed some code that occasionally gets tricked into removing non-deleted scenes from the scene build list. SplitScenes doesn't try to help so hard anymore.

V2.3 MAY 1, 2022

- Fixed a severe bug where a modified scene would ask if you wanted to save the scene. If you said "Save" or "Don't Save" it meant save, only "Cancel" meant do not save. This sometimes would clobber your data, obviously. This is a bug in Unity not my code: `EditorSceneManager.SaveCurrentModifiedScenesIfUserWantsTo()`

V2.2 JULY 12, 2021

- Reworked the Scene Locking logic to prevent an occasional bug where scenes would be marked modified when changing the selection status of the scenes at the wrong times.

V2.0

- Huge rewrite, including requiring Unity 2019 as the minimum version.
- The SplitScenes inspector now does all the work and the old SplitScenes Editor Window is gone.
- Added a Lock button over the Hierarchy Window, for more convenient locking of scenes.
- Moved the storage of Lock status out of the root game scene and into EditorPrefs, so your settings don't affect other people or dirty your scene.
- Added a Menu toggle to disable Lock entirely, also just stored on your machine.
- Added proper reference counting for loaded scenes, which makes big world async loading possible.
- Too many quality-of-life improvements to list. Check out the video when it's posted in the store.

V1.5

- Revised the error handling for UnityWebRequest now that the interface changed in 2020.2.

V1.4

- Simplified version checking popup.

V1.3

- Worked around a full editor crash introduced by a bug in Unity 2019.3.

V1.2

- Added better styling for Pro skin.
- Added synchronization and ordering to scene initialization after all loads are finished.
- Added DoLogging checkbox and swept all the noisy console logs into conditionals.
- Cleaned up namespaces around the code, upgraded the web popup.

- Got SplitScenes working in 2017.4LTS.
- Updated documentation.

V1.1

- Documentation
- Fixed various bugs relating to deleting scenes, moving them in projects leaving broken path links, and so forth.
- Added the plus button for rapid new-scene creation.
- Fixed window size and layouts so the trash can button isn't hidden by default.

V1.0

- Initial Release