Namespacer

Manual

v1.1 August 2016

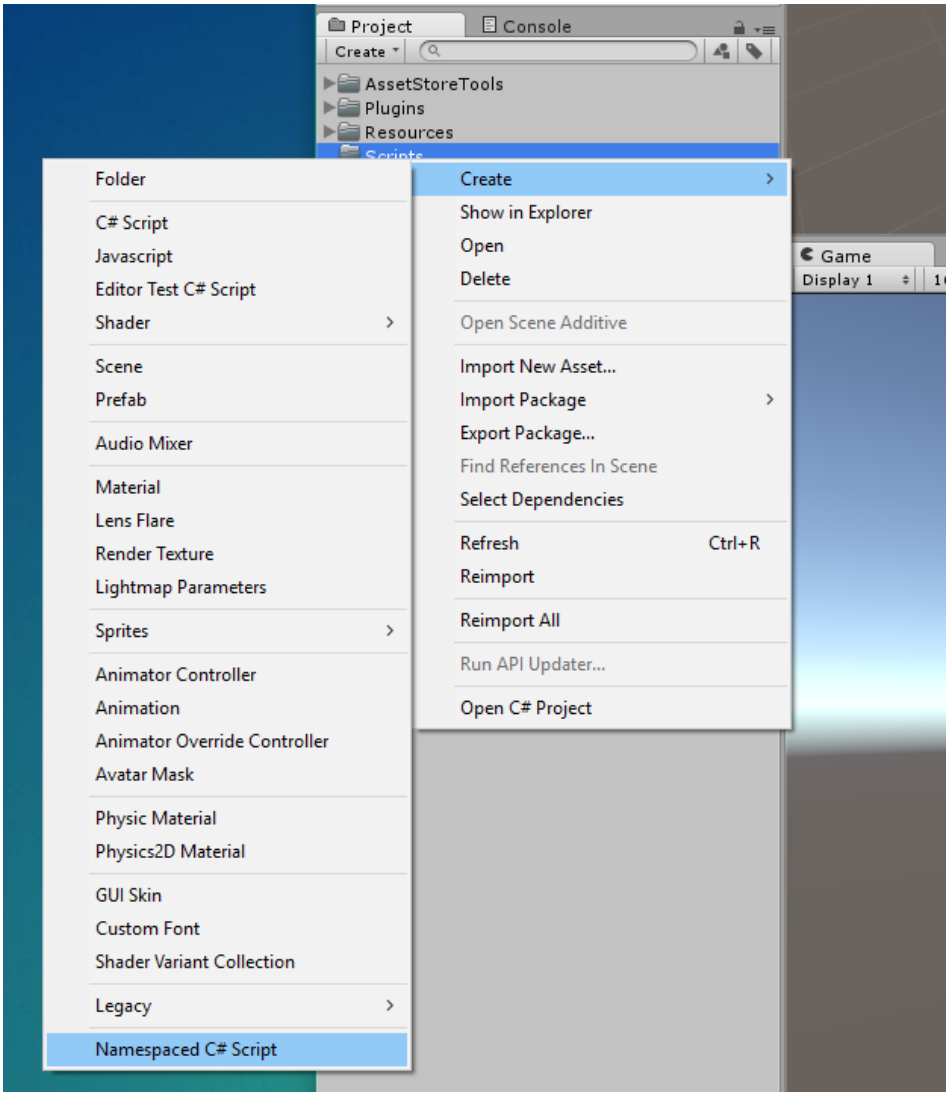
Yo!

Thanks for using Namespacer! Chances are you already have a pretty good idea what Namespacer does, and our hope is you find it pretty straightforward. If you still are not sure what Namespacer is, it is a simple editor extension that enables you to create C# Monobehaviour scripts with your own namespaces built in. The ultimate goal is to save you time by avoiding the annoying task of adding in your namespaces every time you create a script. Generally, Namespacer follows the same rules as Visual Studio does when creating C# files in a .NET project (i.e. namespaces mimicking the folder directory structure).

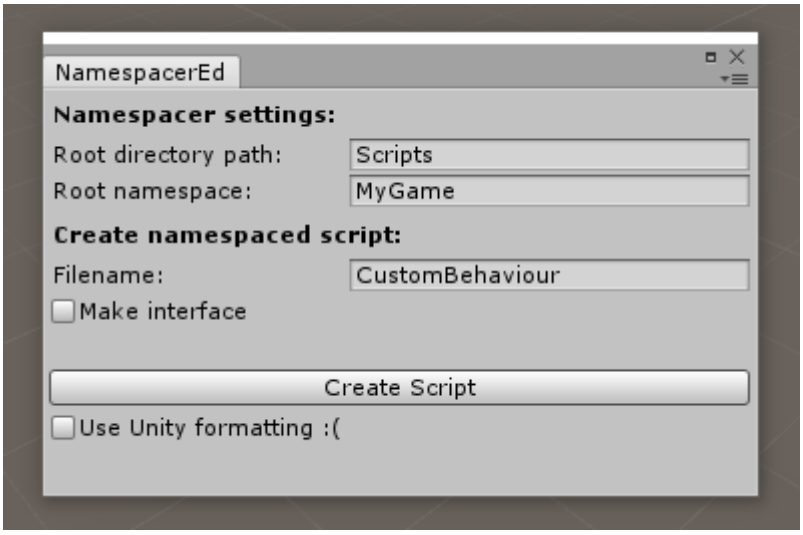
Namespacer is meant to be as simple and light as possible, so our hope is it helps to make your life easier in nice tiny increments. With all that being said, we would still love to hear from you if you have any comments, bugs, or ideas for improvement. You can hit us up at **support@searinggames.com**.

Getting Started

Namespacer is pretty simple: Just import the asset into your project, and you should immediately have an option in the project window under “Create” > “Namespaced C# Script”.



From there, you will see an editor window pop that should look something like this:



Once you are happy with all your info, just click “Create Script”, and you have a shiny new script with a namespace!

Settings

Generally, any value (aside from the Filename) you edit in the editor window is saved as soon as you create a script with Namespacer. All the settings for Namespacer are containted right in the editor window. Here is what they do:

Root directory path:

This is the root relative path, starting after the Assets folder in your Unity project, that you want to be ignored in your namespace path. This is optional, and if left blank, your namespace structure will mimic the exact folder structure you see in your project window.

For example, if you create a script in the folder “Scripts” with the root directory path setting left blank you should see the namespace *MyGame.Scripts*.

However, if you have you, a root directory path set to “Scripts” and you create a script in the folder “Scripts”, you will see a new script with the namespace *MyGame*. Any scripts created in that folder will mimic the directory structure with the folder “Scripts” ignored.

If a directory path cannot be resolved, the directory is ignored and the namespace will just mimic the folder structure. Additionally, the root directory path must be a complete one, so if you wanted to ignore the “Player” folder from within the “Scripts” folder you would need a root directory path of “Scripts/Player”.

Root namespace:

This is the primary namespace in all of your scripts. By default, or if left blank the root namespace is the name of your Unity project.

Make interface:

Checking this toggle creates an empty C# interface with a namespace. If you are using namespaces, chances are you might need an interface or two!

Use Unity formatting:

Leaving this unchecked means your scripts will be created with the same formatting Visual Studio uses when creating C# files. Check this box off if you want to use the way Unity formats regular Monobehaviours.

That's It!

Like we said before, Namespacer is meant to be a simple tool for a simple job. Hopefully it starts saving you time immediately!

If you have any questions, comments, rage, etc., send us an email at **support@searinggames.com**.

Happy namespacing!

- *Searing Games*