SpeedScript

Made by Star Banner Games.

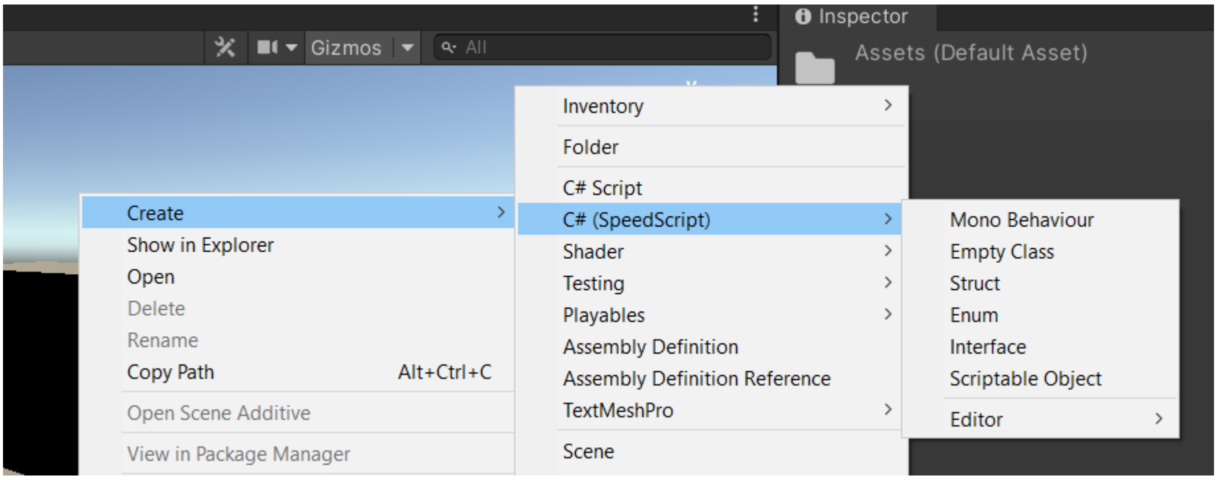
Itch.io: Star Banner Games **LinkedIn:** Sanandana Michael Ithaler **Twitter:** @starbannergames

Content

1 QuickStart

1.1 Creating Scripts

In the Project Window, right click and navigate to "**Create/C# (SpeedScript)**". Here you can create a Script using all available Templates.

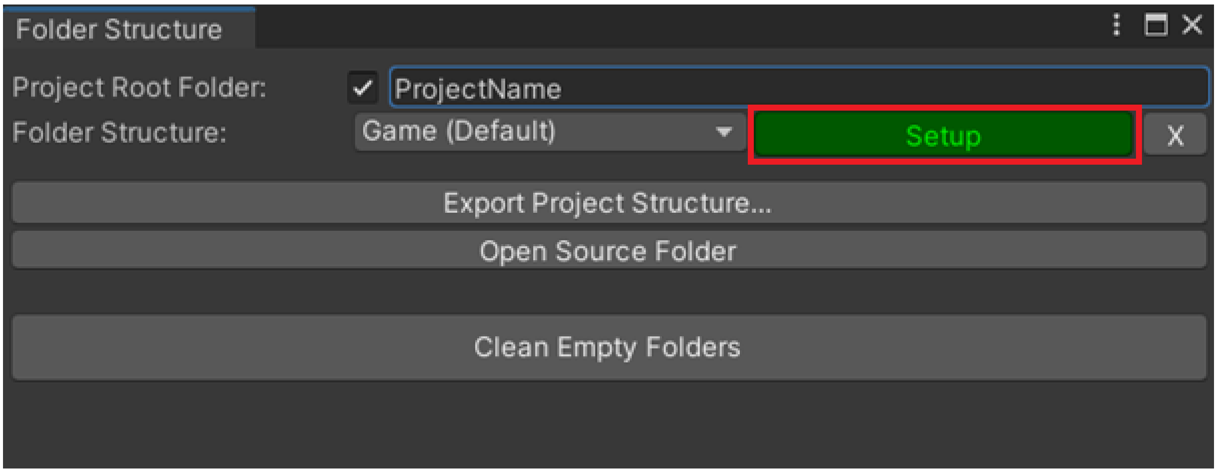


1.2 Folder Structure

To quickly set up the Default Folder Structure, navigate to:

“**Window/Star Banner Games/SpeedScripts/Folder Structure**”.

There, just click on the green “**Setup**” Button. Done!



Project Root Folder: ☑ ProjectName

Folder Structure: Game (Default)

2 Advanced Settings

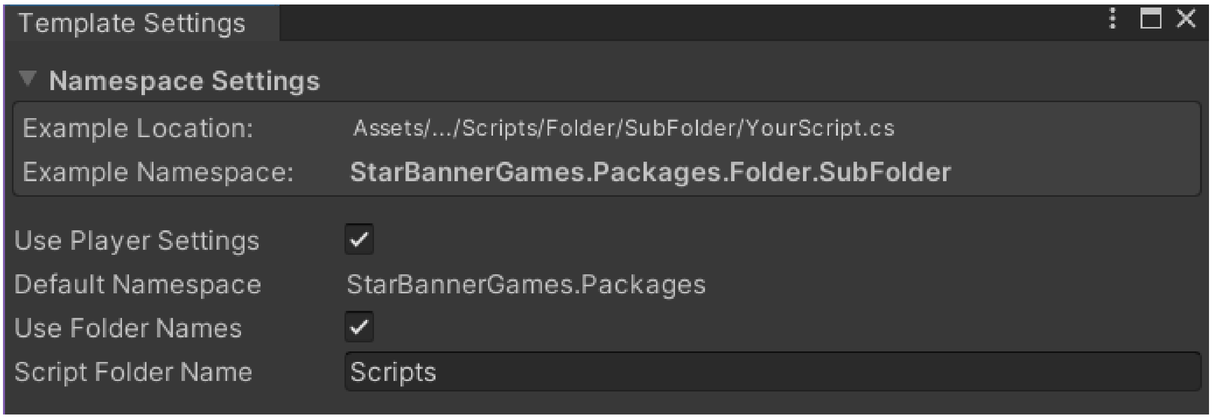
2.1 Template Settings

Navigate to "**Window/Star Banner Games/SpeedScript/Template Settings**".

The Template Settings allow you to define what Namespaces your Scripts will be assigned to, as well as manage custom Script Templates.

2.1.1 Namespace Settings

The Namespace Settings are used to define what Namespace a created script will be assigned to. Let's go through all elements from top to bottom:



Template Settings

▼ Namespace Settings

Example Location: Assets/.../Scripts/Folder/SubFolder/YourScript.cs
Example Namespace: StarBannerGames.Packages.Folder.SubFolder

Use Player Settings ☑
Default Namespace StarBannerGames.Packages
Use Folder Names ☑
Script Folder Name Scripts

2.1.1.1 *Namespace Preview:*

Example Namespace shows you what the namespace of a script might look like using the current settings. **Example Location** also gives an example of the path that script is located in, as the path can influence the Namespace.

2.1.1.2 *Use Player Settings*

If **enabled**, the **Default Namespace** will be set using the Company Name and Product Name defined in the Player Settings. It will have the format "***Company.Product***".

If **disabled**, the **Default Namespace** can be defined through a text field.

2.1.1.3 *Default Namespace*

This is the base Namespace that will always be used for scripts.

2.1.1.4 Use Folder Names

If **enabled**, the Asset Folder Path of a created script will be appended to the **Default Namespace** to create the full namespace.

Let’s say we have a script at the path “***Assets/AISystem/Pathfinding***”. The resulting namespace would be: “**DefaultNamespace.AISystem.Pathfinding**”.

2.1.1.5 Script Folder Name

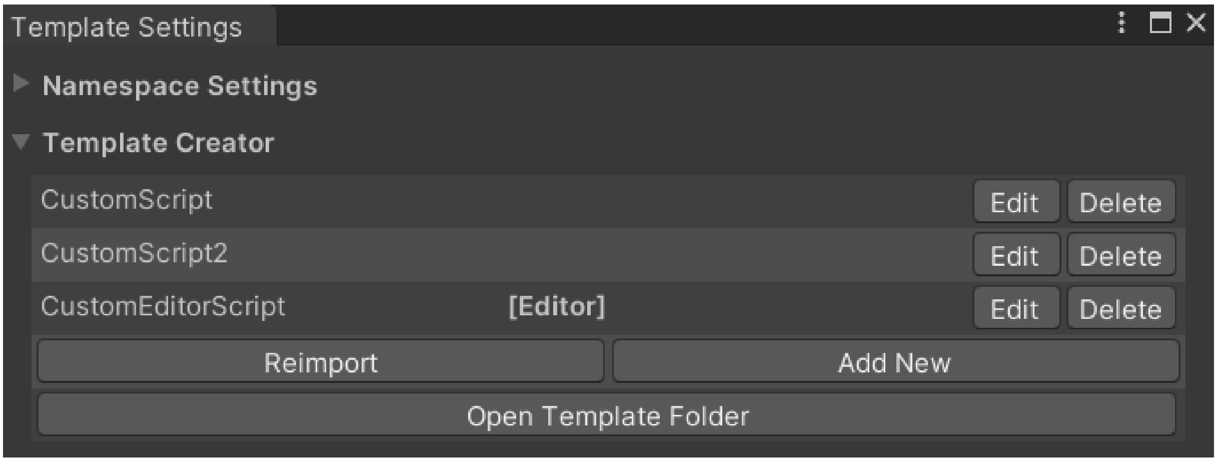
*(This control is only active after selecting **Use Folder Names**.)*

The name of your Scripts folder. Defaults to “Scripts”. This is used to ignore unnecessary folders in your namespaces.

Example: A script is in the folder “*Assets/CoolGame/Scripts/SaveSystem*”. The resulting Namespace would be: “**DefaultNamespace.SaveSystem**”.

2.1.2 Template Creator

With the Template Creator you can create your own Custom Templates to create Scripts exactly the way you need them across all of your projects. Let’s go through all elements from top to bottom:



2.1.2.1 Custom Template List

Here you can edit or delete all your custom Templates. It also shows if a given Template is used for Editor Scripts. Pressing edit will simply open the template in your default program for **.txt** files. Deletion might take a few seconds as Unity needs to recompile the changes.

2.1.2.2 Reimport

When tinkering with files a lot it can happen that the Context Menu does not Update accordingly and throws errors. Pressing the Reimport Button should fix these issues. A Unity restart should also have the same effect.

2.1.2.3 Add New

This button will let you create a new Script Template. After creation it should open the Template as a **.txt** file for you to edit. Save the file and come back to Unity. After compiling it should now appear in the "**Create/C# (SpeedScript)**" Context Menu.

Note that Editor Script Templates appear in the "**Create/C# (SpeedScript)/Editor**" Context Menu.

2.1.2.4 Open Template Folder

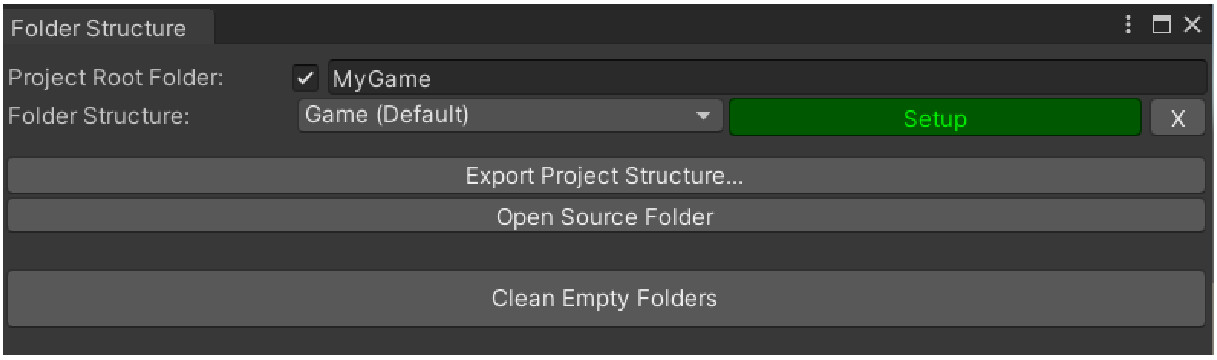
This will open the Folder with your Custom Templates in the File Explorer for easy access.

WARNING: Creating/Deleting Template Files by hand can break References. If you decide to do so, either close Unity or make sure to press the Reimport Button afterwards.

2.2 Folder Structure

Navigate to "**Window/Star Banner Games/SpeedScript/Folder Structure**".

Using the Folder Structure Window, you can manage your Projects Folder Hierarchy. This allows you to keep new Projects organized from the start.



Folder Structure

Project Root Folder: ☑ MyGame

Folder Structure: Game (Default)

2.2.1 Project Root Folder

When **disabled**, your defined Folder Structure will be created right in the Assets Folder.

When **enabled**, you can define a Folder name. Now when setting up a Folder Structure, all folders will be created as subfolders of "*Assets/RootFolder/*".

2.2.2 Folder Structure Selection

Here you can select any available Folder Structure. Clicking **Setup** will create the selected Folder Structure in your project. Pressing **X** will delete the selected Folder Structure. Note that the Folder Structure File itself will be deleted, not the actual Folders in your project.

2.2.3 Export Project Structure

If you want to save your own custom Folder Structure, set all the folders up in your project as you need them and then press this button. You will be able to select a root folder that marks the start of your structure. If you do not have a Root Folder, select the Asset Folder. It doesn't matter if you have files within any folders. After exporting, your Folder Structure should be selectable in the **Folder Structure Selection**.

2.2.4 Open Source Folder

This will open the Source Folder where your custom Folder Structures are stored, so you can make changes easier.

2.2.5 Clean Empty Folders

This button will delete every empty folder in your Project that lies withing "*Assets/…*". This has two major use cases:

1. To quickly delete a Folder Structure after Setup, in case the wrong one was selected.

2. To clean unnecessary folders after assets have been added to the project (e.g: a "Models" folder in a 2D game).