Scription – Easy Script Generator

Current version 1.0.2

Table of Contents

Introduction:

Thank you for downloading Scription! First, let me tell you what Scription is not. While the title suggests that this Editor Tool can generate scripts, it is important to specify that it does create properly formatted .cs files that contain the basic structure of a C# script. The Generation, sadly, does not delve into the realm of actual code. But who knows, maybe in the future I can add basic script creation as well– I sure would love to do that, and it may be possible thanks to your support.

Scription Quick-Start:

To get stated using Scription, you can simply click on *Tools* > *Cerebrynth* > *Scription*, from the top bar of your Unity Editor.

Once you do, the Scription's Editor Window will open. This window behaves like any other Unity tab/Window. You may move, resize and dock it anywhere you want. If you don't want it around you can close it and reopen it from the menu at any time.

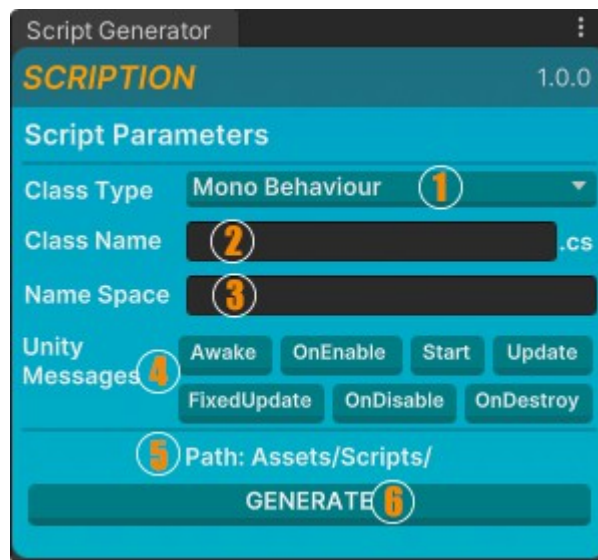To create a script, simply perform the following actions:

1. Select the Type of class you would like to create from the ClassType drop down.

2. You must type in a valid* name for your class.

3. You may type in a custom Namespace for your application – if you leave the window blank, a default Namespace will be added. (it's good practice)

4. You may choose to add any of the basic Unity Messages by clicking on the toggles with their respective names.

5. Click on the Generate button to have the script be created in your Assets/Scripts/ folder. If this folder does not Exist, Scription will create it for

you.

6. Get to coding your awesome game!

I have included some amount of filename validation and correction. If you should create a script with an invalid filename/class name(including invalid characters or other) the resulting file can simply be deleted without causing any issues in your project. Unity's console window will inform you of any compile errors.

The Editor Window

The editor window is composed of the following items:

1. The Class Type drop-down: inside of this drop-down you can select the type of class to create. As of version 1.0.0 you have access to:
   1. MonoBehavour
   2. Base Class
   3. Abstract Class
   4. Interface
   5. ScriptableObject Class
   6. Struct
   7. Enum

2. Class Name: The input of this text field will represent both the Class name and

the file name in your Scripts Folder.

3. Namespace: The input of this text field will be assigned to the script's Namespace declaration.

4. Unity Messages: These toggles control which, if any, basic Unity Messages the script will contain. The names on the toggles represent the names of the methods to include.

5. Assets/Scripts/: A label to inform the user where the scripts will be saved. If this directory does not exist in your project, Scription will create it automatically.

6. Generate: This button will begin the file generation and saving process. Keep an eye on your Scripts folder for the newly created .cs file.

Conclusion and Notes:

I hope you enjoy using Scription and that you find it as helpful as I have in my own projects. I have plans on expanding the tool to support a few more functions but it is important to underline that adding granular controls could potentially achieve the opposite effect of slowing down your workflow.

I will always be happy to receive feedback and answer questions you may have. You may contact me through www.cerebrynth.com or by email at: cerebrynth@gmail.com

Thanks again for your support and *HAPPY CODING!*

Troubleshooting:

- An issue may result where Newtonsoft Json is not included in the project packages. To correct this problem, you can open *Window > Package Manager* and click on the plus icon below the window's title. Then you can select "*Add package from git URL…*" and paste the following text:

  com.unity.nuget.newtonsoft-json

  Once installed you should be able to use Scription without a problem.