Sublime Text Project’s Integration Tool for Unity3D (Version 1.0)

By Lif

Introduction

Voidless’ Sublime Text Integration into Unity is a tool that lets the user automatically create a Sublime Project to quickly navigate through scripts in the project.

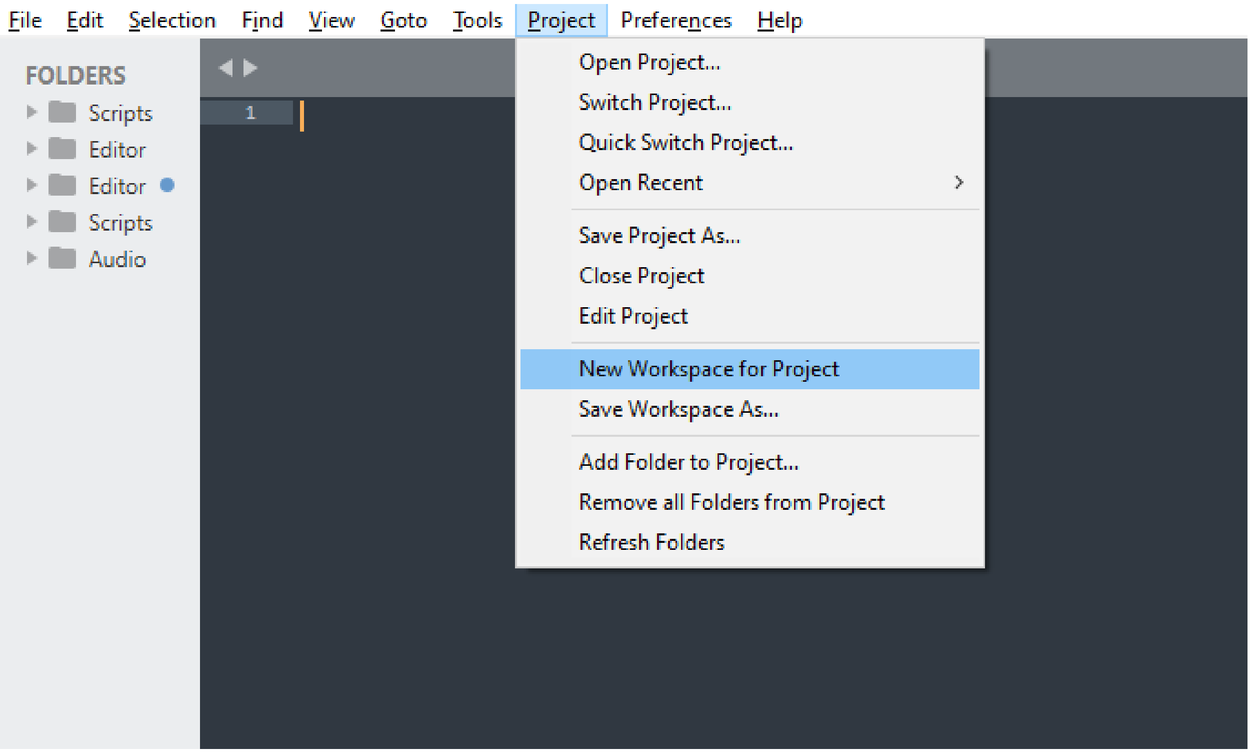
Why Sublime Text?

Sublime Text is a lightweight text editor that tends to be used for scripting. It is a great option if you wanna trade-off features from IDEs (such as compilation, debugging on the IDE's console, IntelliSense, etc.) for fast and efficient typing.

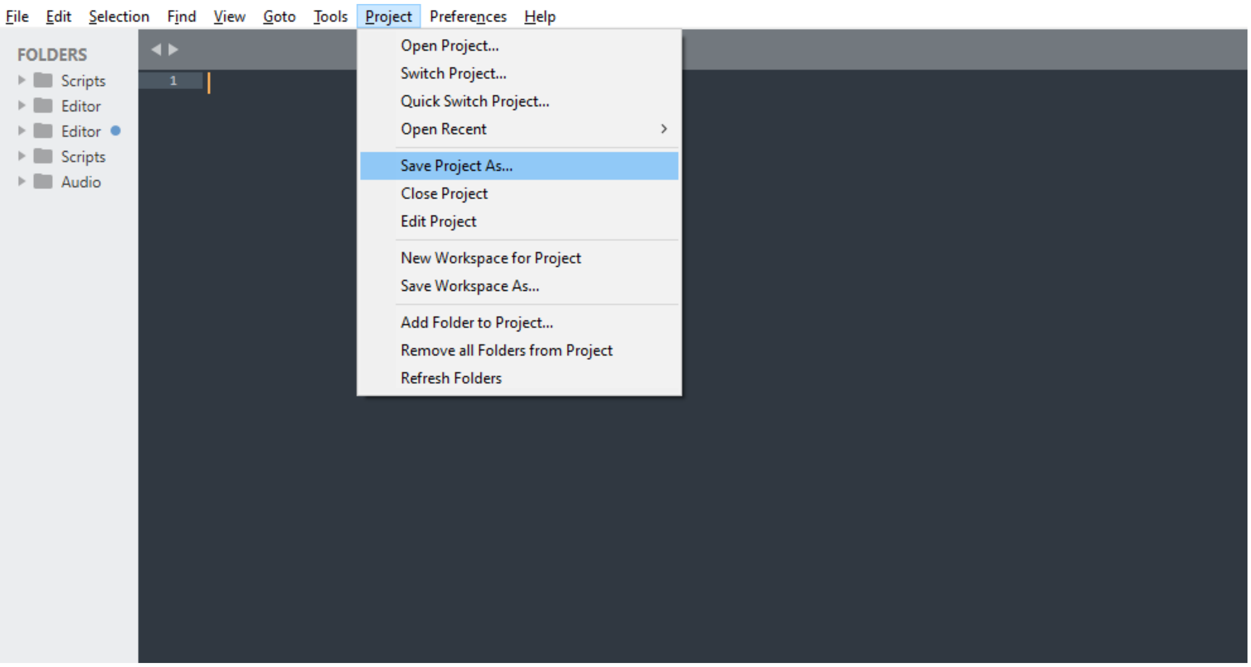
Setup

Without this Tool

To create a Sublime Project, you must first need to create a new project workspace. For that, go to Project/New Workspace for Project on the menu bar:

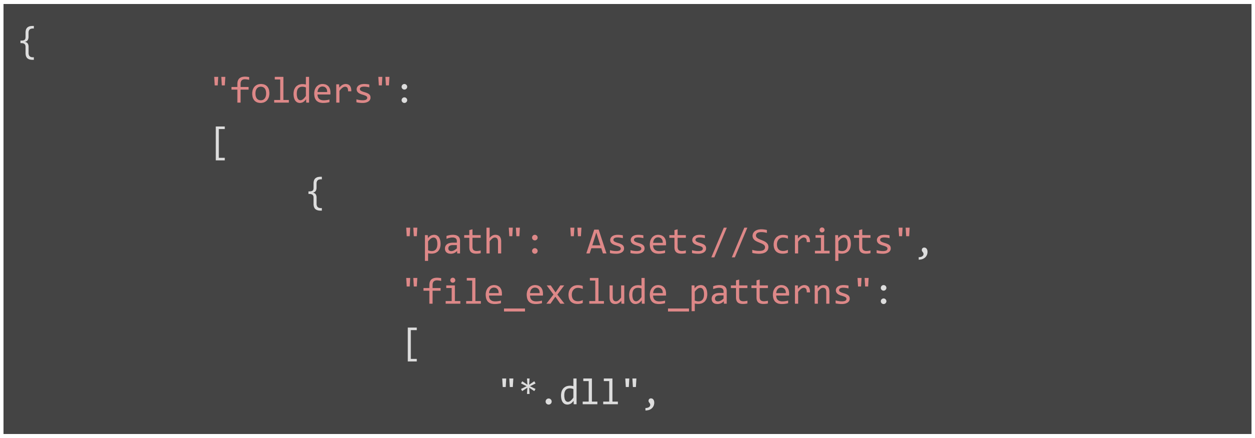


After that, you’ll have to create a Sublime Project by going into Project/Save Workspace As…:



This will create an empty Sublime Project file with the extension “.sublime-project”.

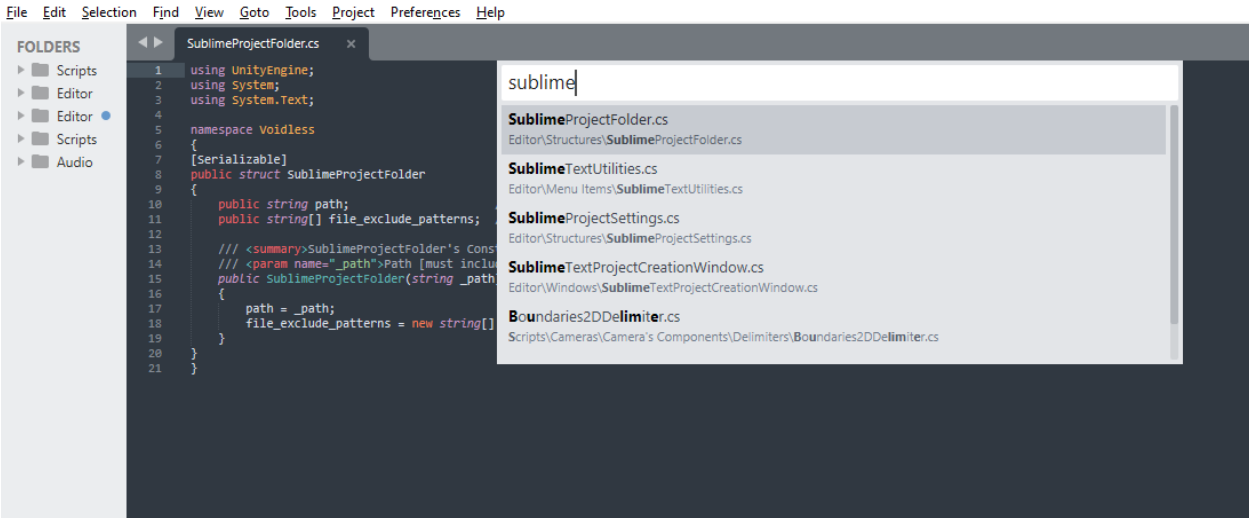
Though you’d have to manually modify that empty file in order for Sublime Text to let you quickly navigate your scripts. This is the structure of a Sublime Project fitting for Unity3D:



```
{
          "folders":
          [
               {
                    "path": "Assets//Scripts",
                    "file_exclude_patterns":
                    [
                         "*.dll",
                    "*.meta"
                ]
            },
            {
                "path": "Assets//Editor",
                "file_exclude_patterns":
                [
                    "*.dll",
                    "*.meta"
                ]
            }
        ]
}
```

It has the syntax of a JSON with the extension *.sublime-project*. The “path” field refers to the root folder from which Sublime Text will retrieve files, the “file_exclude_patterns” field is an array that refers to file exceptions, since script folders tend to include “.meta” and “.dll” files, we omit those.

If you correctly setted up the Sublime Project, you will be able to quickly navigate through defined folders with Ctrl+P, as shown here:



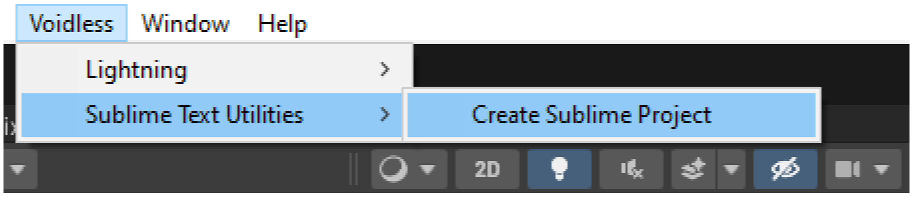
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Scripts
Editor
Editor
Scripts
Audio
SublimeProjectFolder.cs
sublime
SublimeProjectFolder.cs
Editor\Structures\SublimeProjectFolder.cs
SublimeTextUtilities.cs
Editor\Menu Items\SublimeTextUtilities.cs
SublimeProjectSettings.cs
Editor\Structures\SublimeProjectSettings.cs
SublimeTextProjectCreationWindow.cs
Editor\Windows\SublimeTextProjectCreationWindow.cs
Boundaries2DDelimiter.cs
Scripts\Cameras\Camera's Components\Delimiters\Boundaries2DDelimiter.cs

```
using UnityEngine;
using System;
using System.Text;

namespace Voidless
{
[Serializable]
public struct SublimeProjectFolder
{
    public string path;
    public string[] file_exclude_patterns;

    /// <summary>SublimeProjectFolder's Cons
    /// <param name="_path">Path [must inclu
    public SublimeProjectFolder(string _path
    {
        path = _path;
        file_exclude_patterns = new string[]
    }
}
}
```

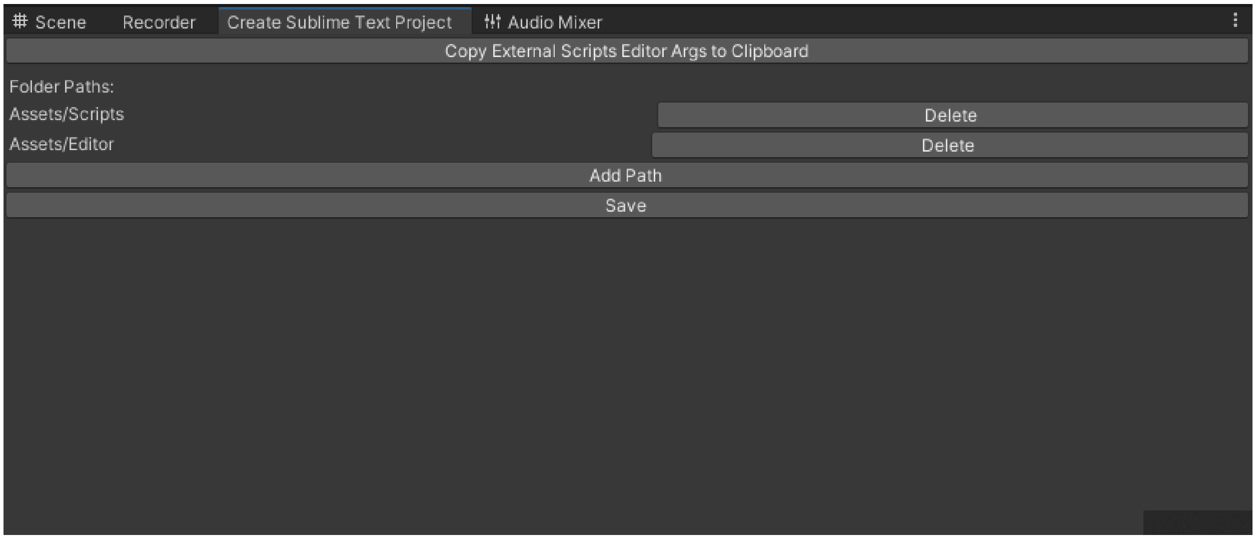
This certainly makes scripting in Sublime Text more efficient than it already is. However, and given that I have made plenty of Sublime Projects, I find this process quite tedious; therefore this personal need to develop a tool to automate this process.

With this Tool

In your Editor window navigate through Voidless/Tools/Sublime Text/Project Creation in the bar menu:



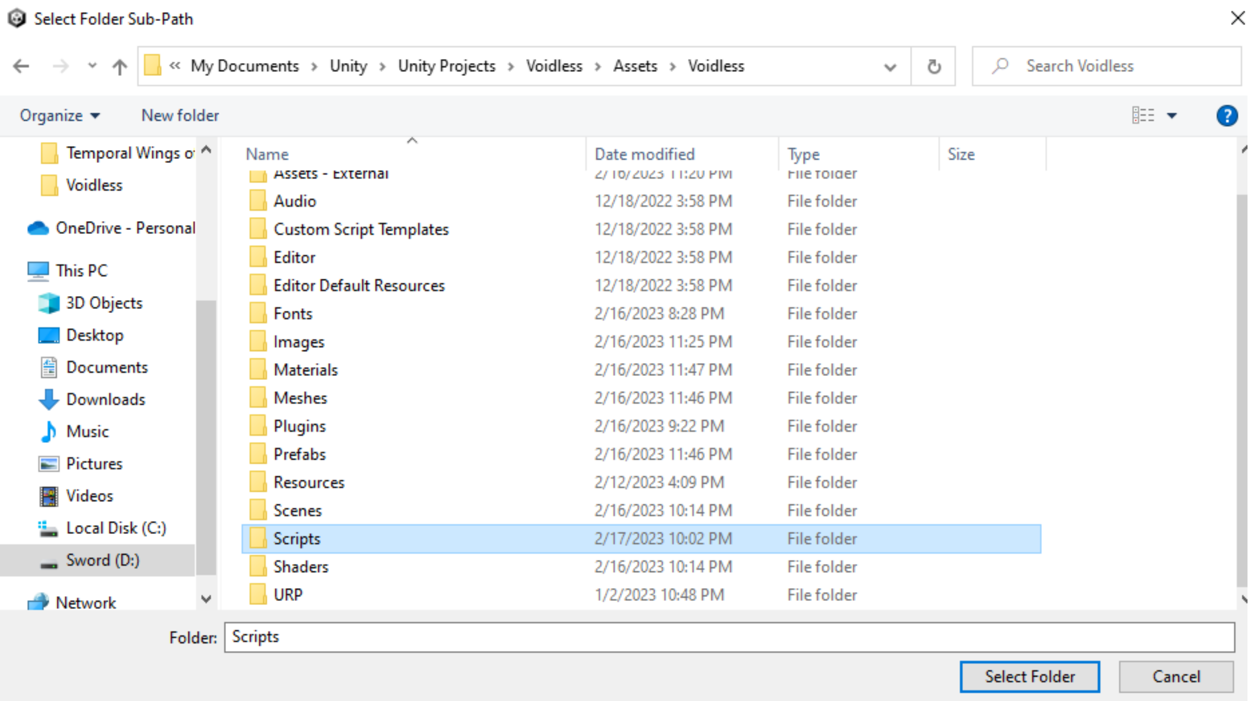
A window for creating a Sublime Project will appear. By default, 2 paths for Scrips and Editor will be created (since most projects have those folders). You can delete either of the paths and add more paths if you wish with the button “Add Path”.


Scene
Recorder
Create Sublime Text Project
Audio Mixer
Copy External Scripts Editor Args to Clipboard

Folder Paths:
Assets/Scripts
Assets/Editor

Delete
Delete
Add Path
Save

When you click “Add Path” a window will open where you will have to select the a new root folder:

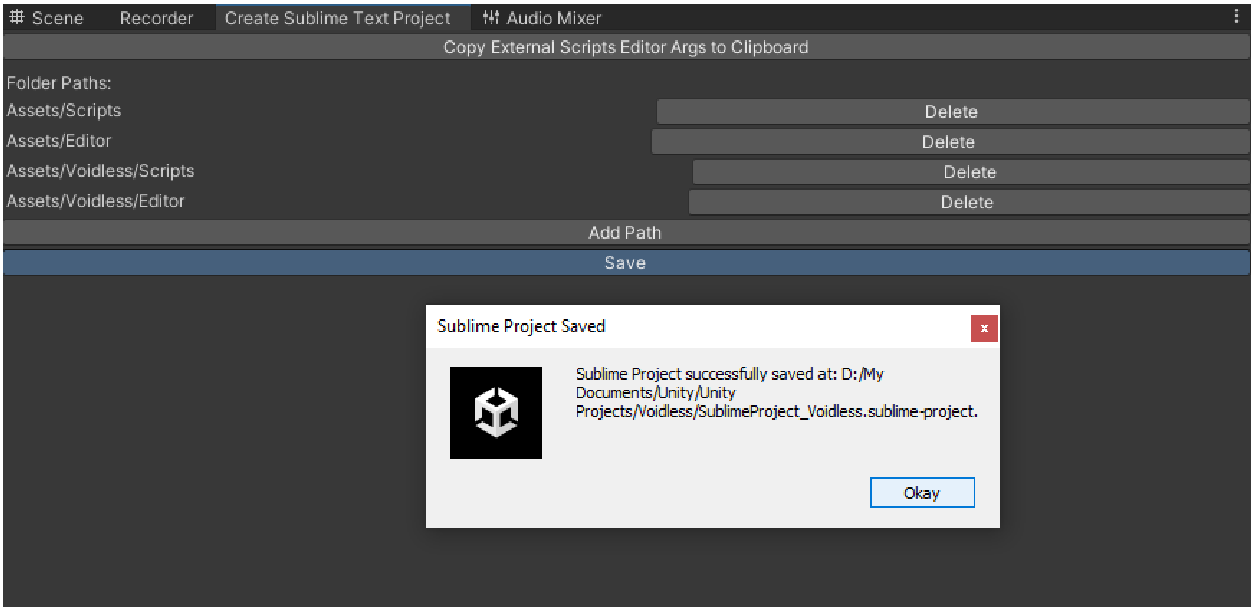

Select Folder Sub-Path

« My Documents › Unity › Unity Projects › Voidless › Assets › Voidless

Name	Date modified	Type	Size
Assets - External	2/16/2023 11:20 PM	File folder	
Audio	12/18/2022 3:58 PM	File folder	
Custom Script Templates	12/18/2022 3:58 PM	File folder	
Editor	12/18/2022 3:58 PM	File folder	
Editor Default Resources	12/18/2022 3:58 PM	File folder	
Fonts	2/16/2023 8:28 PM	File folder	
Images	2/16/2023 11:25 PM	File folder	
Materials	2/16/2023 11:47 PM	File folder	
Meshes	2/16/2023 11:46 PM	File folder	
Plugins	2/16/2023 9:22 PM	File folder	
Prefabs	2/16/2023 11:46 PM	File folder	
Resources	2/12/2023 4:09 PM	File folder	
Scenes	2/16/2023 10:14 PM	File folder	
Scripts	2/17/2023 10:02 PM	File folder	
Shaders	2/16/2023 10:14 PM	File folder	
URP	1/2/2023 10:48 PM	File folder	

Folder: Scripts

After you are done adding your desired paths, click the button “Save”, this will create a Sublime Project with the name “SublimeProject_PROJECTNAME.sublime-project”, where PROJECTNAME will be the particular name of your project. The file will be saved under the project's folder (the same folder where Assets is located).


Scene
Recorder
Create Sublime Text Project
Audio Mixer
Copy External Scripts Editor Args to Clipboard
Delete
Delete
Delete
Delete
Add Path
Save
Okay

Folder Paths:
Assets/Scripts
Assets/Editor
Assets/Voidless/Scripts
Assets/Voidless/Editor

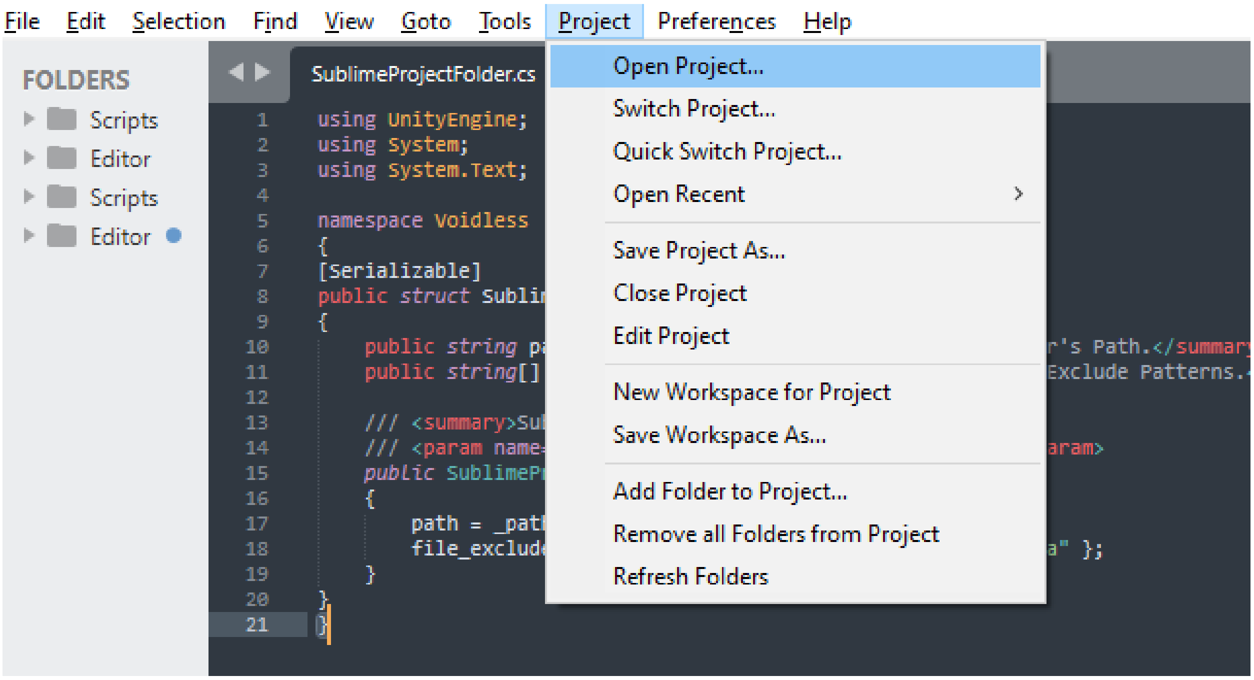
Sublime Project Saved

Sublime Project successfully saved at: D:/My Documents/Unity/Unity Projects/Voidless/SublimeProject_Voidless.sublime-project.

In the particular case of this example, the resulting Sublime Project ended up being like this:

```
{
    "folders": [
        {
            "path": "Assets/Scripts",
            "file_exclude_patterns": [
                "*.dll",
                "*.meta"
            ]
        },
        {
            "path": "Assets/Editor",
            "file_exclude_patterns": [
                "*.dll",
                "*.meta"
            ]
        },
        {
            "path": "Assets/Voidless/Scripts",
            "file_exclude_patterns": [
                "*.dll",
                "*.meta"
            ]
        },
        {
            "path": "Assets/Voidless/Editor",
            "file_exclude_patterns": [
                "*.dll",
                "*.meta"
            ]
        }
    ]
}
```

Now, as a final step, go to Sublime Text, and open your project under the path Project/Open Project on the menu bar. You’ll have to select the file created by this tool.

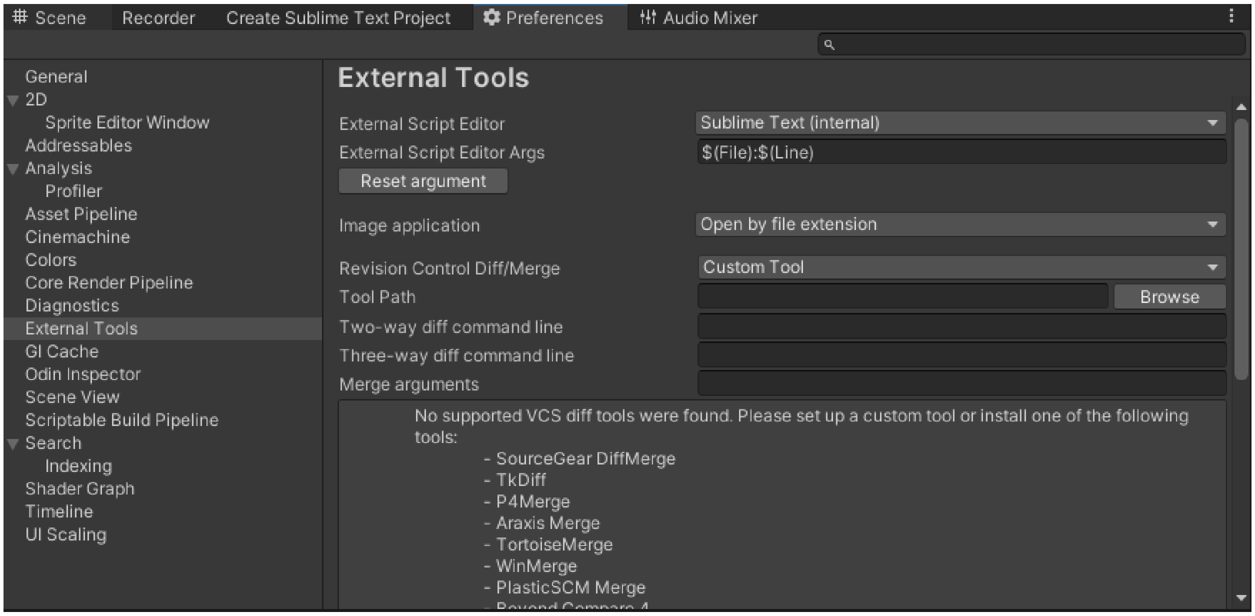


```
using UnityEngine;
using System;
using System.Text;

namespace Voidless
{
[Serializable]
public struct Sublir
{
    public string pa                                  r's Path.</summary
    public string[]                                   Exclude Patterns.

    /// <summary>Su
    /// <param name=                                  aram>
    public SublimePr
    {
        path = _path
        file_exclude                                  a" };
    }
}
}
```

Additional Features

Setting Script Editor Arguments for Sublime Text

On the same Sublime Project creation window, there is a button named “Copy External Scripts Editor Args to Clipboard”, if you click that button, the path “$(File):$(Line)” will be copied into your clipboard. Now go to Edit/Preferences/External Tools, and paste that path on the “External Script Editor Args” field:



External Tools

External Script Editor: Sublime Text (internal)
External Script Editor Args: $(File):$(Line)
Image application: Open by file extension
Revision Control Diff/Merge: Custom Tool
Tool Path
Two-way diff command line
Three-way diff command line
Merge arguments

No supported VCS diff tools were found. Please set up a custom tool or install one of the following tools:
- SourceGear DiffMerge
- TkDiff
- P4Merge
- Araxis Merge
- TortoiseMerge
- WinMerge
- PlasticSCM Merge
- Beyond Compare 4

With this setup, you will be able to navigate directly into the file and line of a script when you double-click a Console Log’s entry that refers to a script (whether it is a normal warning or error log).

.gitignore Gets Updated too

When you save a Sublime Project with this tool, additional exception entries are added to the .gitignore automatically (“*.sublime-project” and “*.sublime-workspace”). These files are ignored just in case it may cause conflicts with other members that pull changes from the repository that contains your project.

Further Improvements

As of right now, I can think of the following improvements to implement on further versions:

- Better visual representation of the window.
- Automatically add the editor script arguments to the External Tools' window.
- Et cetera (if you can think of any features/bugs that I may have omitted, do not hesitate to contact me).

Contact

Have any questions/suggestions/comments? Don't hesitate to contact me through the following:

- E-mail: lif.gamedev@gmail.com
- Instagram: @lif.gamedev

Bonus

If you like Sublime Text and haven't integrated any snippets yet, make sure to checkout my snippets' repository at Github: https://github.com/LifGwaethrakindoII/TemplatesAndSnippets