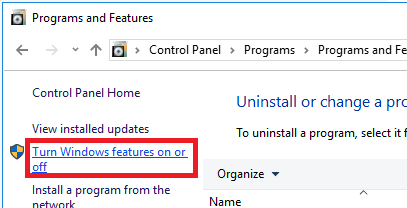
**For unity2017:**

To be able to compile .net libraries for unity, you need to use .net framework 3.5, or less. Since this is legacy .net, you need to configure Windows in a certain way.

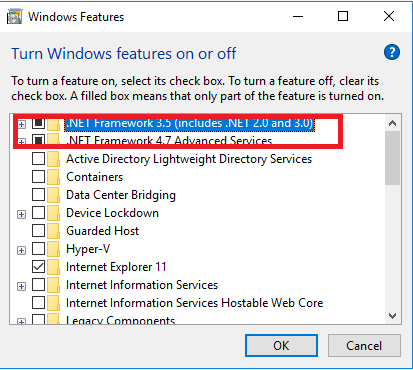
1. Make sure you have .net 3.5 installed.

This is not available directly from the visual studio installer, you will need to do this:

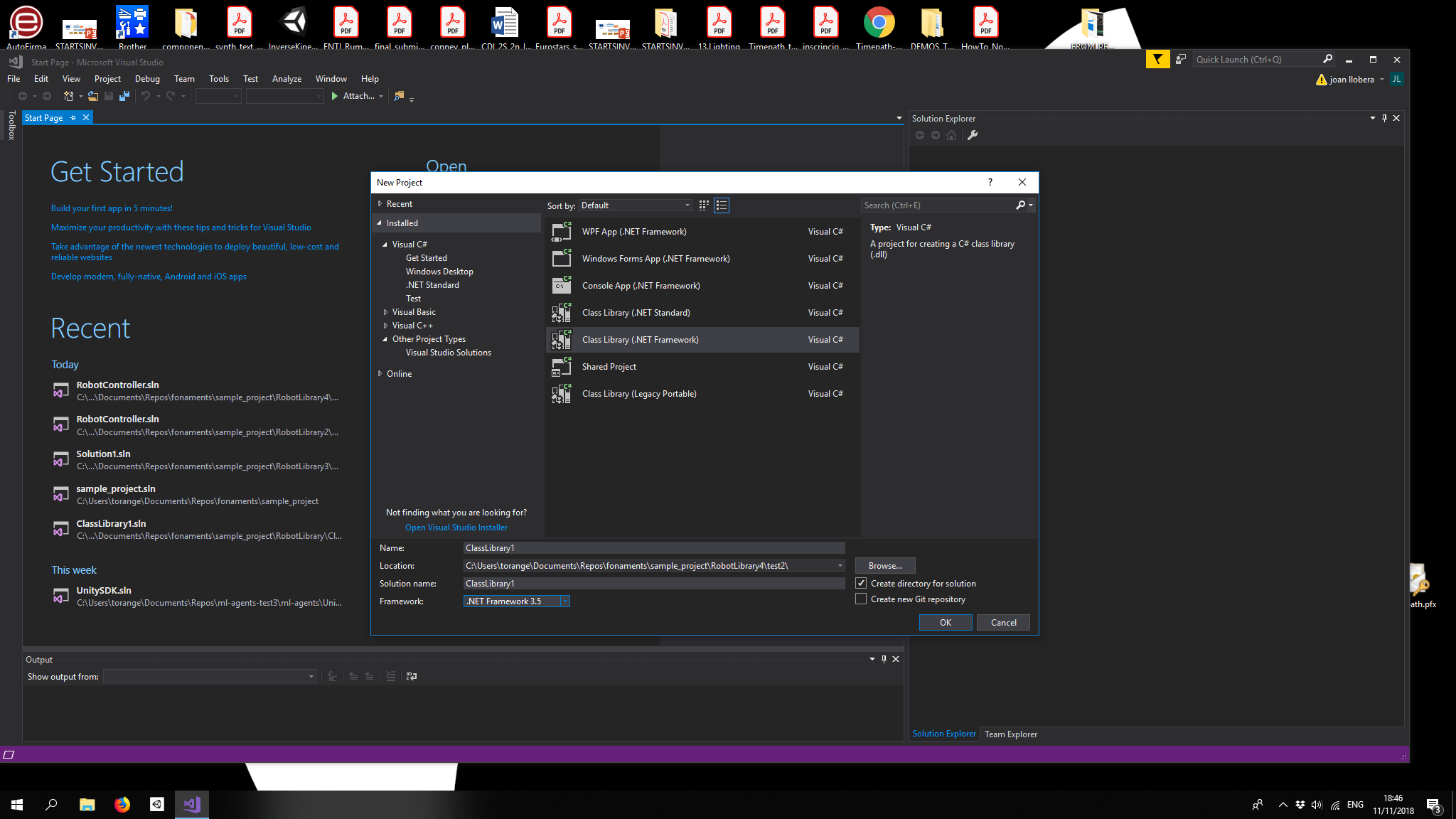
1. Open your control panel. Click Uninstall a program in Programs. Choose Turn Windows features on or off.



1. Make sure .net 3.5 is well checked



1. Restart the computer
2. Create a new project using the .net 3.5 framework



1. Drag and drop the resulting .dll into the Unity asset folders and it should work out of the box.

References:<https://social.msdn.microsoft.com/Forums/es-ES/64b19bac-b5e2-40ce-aea2-efdfd53b1768/install-net-framework-35-for-visual-studio-c-class?forum=netfxbcl> <https://docs.unity3d.com/Manual/UsingDLL.html> <https://docs.microsoft.com/en-us/dotnet/core/tutorials/library-with-visual-studio>

**For unity2019:**

The previous is solved, due to the fact that in 2019 versions unity targets .net 4.0, instead of .net3.5

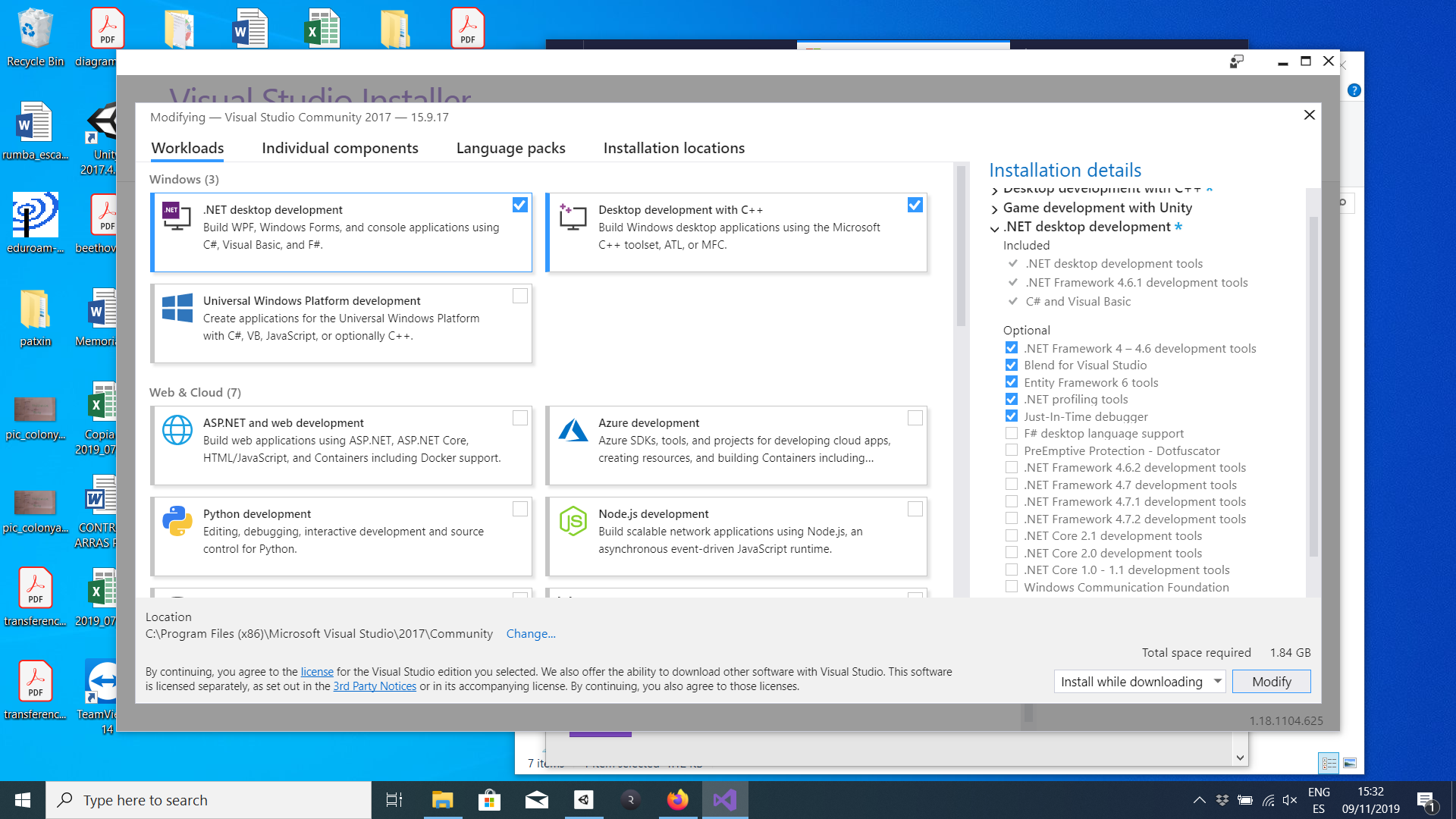
However, in cases where you want to use **UnityEngine.dll** inside your library, you may find the following error:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Severity | Code | Description | Project | File | Line | Suppression State |
| Warning |  | The primary reference "UnityEngine" could not be resolved because it has an indirect dependency on the .NET Framework assembly "mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" which has a higher version "4.0.0.0" than the version "2.0.0.0" in the current target framework. | OctopusController |  |  |  |

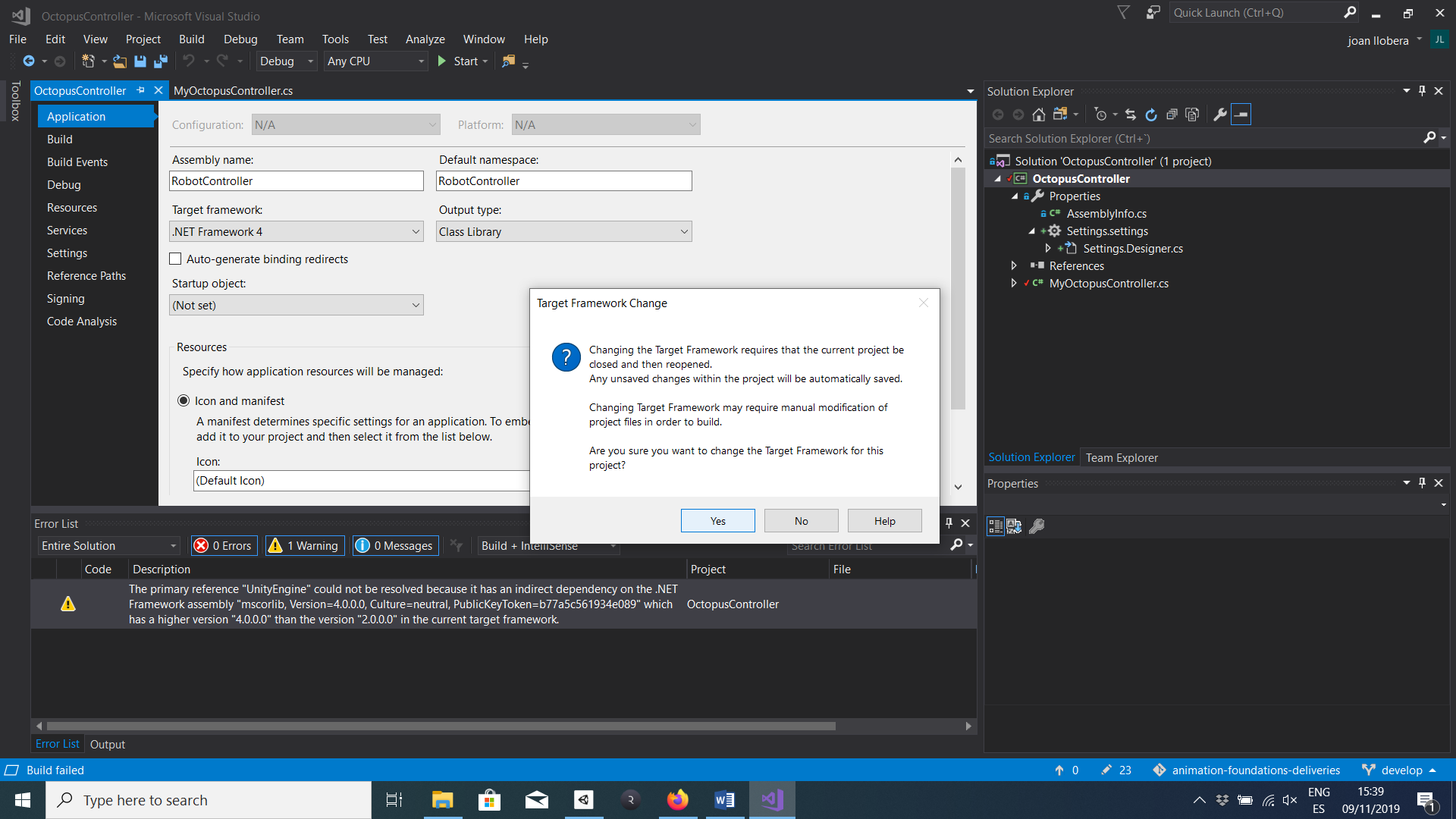
To address this (reference [here](https://developercommunity.visualstudio.com/content/problem/80670/could-not-load-file-or-assembly-microsoftvsdesigne-1.html)).

Please do the following to fix your issue:

1. Quit VS 2017
2. Launch VS 2017 Installer (update it and update visual studio, if necessary)
3. Click Modify.
4. Check the ".NET desktop development" workload.



1. Click Modify.
2. Reload your project
3. Right click on the project, open the properties panel, and change the .net target version to 4.0 (See below)



1. Close and reload your project
2. Confirm whether it compiles

Once you are done, you will still have to add a dependency. The simplest way to do this is described here:

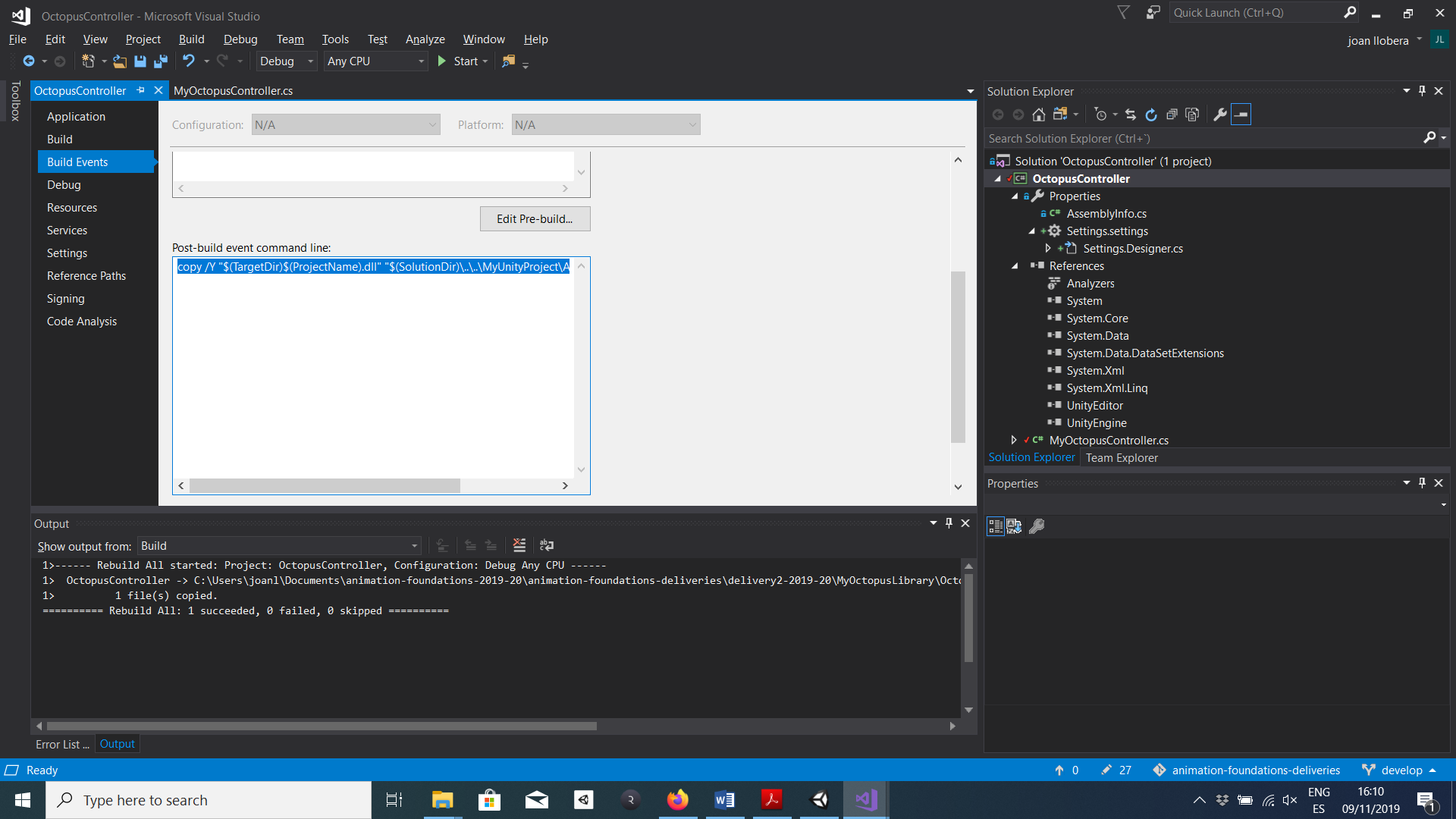
<https://docs.unity3d.com/2018.4/Documentation/Manual/UsingDLL.html>

For encapsulation purposes, in this project we have chosen to simply add them in a folder, in order the project is self-contained across dependencies

In addition, to automate the copy of the .dll generated inside the unity project, you can add this instruction:

copy /Y "$(TargetDir)$(ProjectName).dll" "$(SolutionDir)\..\..\MyUnityProject\Assets\$(ProjectName).dll"

This should be added to the post-build events, inside the project properties (see screenshot below).



**Debugging in unity2019:**

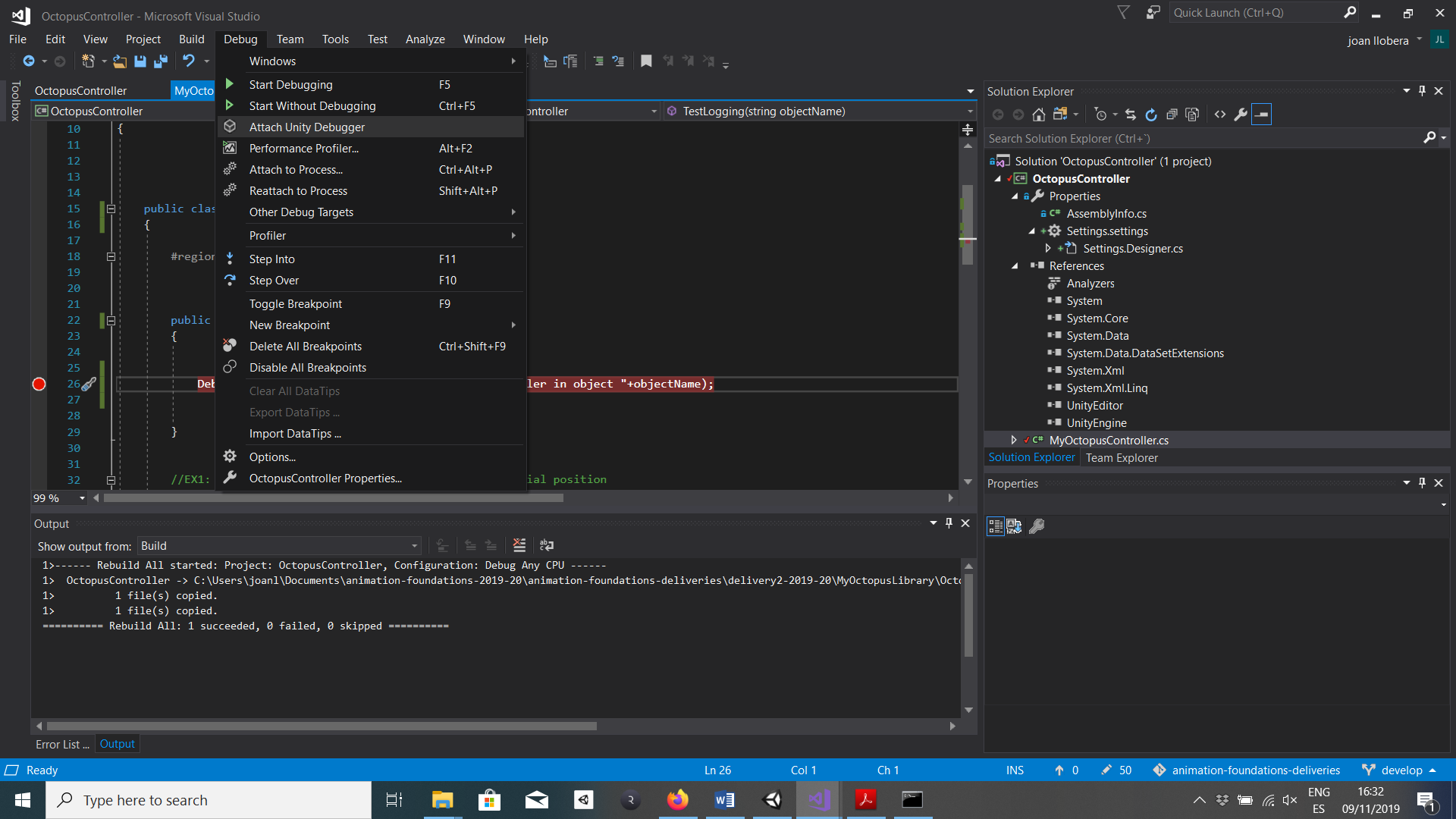
If you also want to debug the .dll using the Unity project within the Unity editor, the documentation is a bit confusing (see here: <https://docs.unity3d.com/Manual/UsingDLL.html> )

Something that has worked for me in the past is to also copy the .pdb inside the Plugin library. You can use the following post build instructions:

*copy /Y "$(TargetDir)$(ProjectName).dll" "$(SolutionDir)\..\..\MyUnityProject\Assets\Plugins\$(ProjectName).dll"*

*copy /Y "$(TargetDir)$(ProjectName).pdb" "$(SolutionDir)\..\..\MyUnityProject\Assets\Plugins\$(ProjectName).pdb"*

Then Attach the unity debugger (see screenshot below):



In the popup that follows select the Unity Editor version with which you want to debug, and finally press play in the editor if not done already.