Author: Hooman Salamat

Installing SFML 2.5.1 in Visual Studio 2017

First, you must download the SFML SDK from the download page.

You must download the package that matches your version of Visual C++. Indeed, a library compiled with VC++ 10 (Visual Studio 2010) won't be compatible with VC++ 12 (Visual Studio 2013) for example. If there's no SFML package compiled for your version of Visual C++, you will have to build SFML yourself.

We will be using 64 bit version VC++ 15 (2017). On Windows, choosing 32 or 64-bit libraries should be based on which platform you want to compile for, not which OS you have. Indeed, you can perfectly compile and run a 32-bit program on a 64-bit Windows. So you might want to target 32-bit platforms, to have the largest possible audience. Choose 64-bit packages only if you have good reasons.

You can then unpack the SFML archive wherever you like. Copying headers and libraries to your installation of Visual Studio is not recommended, it's better to keep libraries in their own separate location, especially if you intend to use several versions of the same library, or several compilers. In my case, I renamed the unpacked folder to SFML-2.5.1 and leave it outside my project folder that I am going to create "next".

GBC > S4-GAME2002-T163 > GAME2002-T163-2019 > Week1			Search Week1	
Name	Date modified	Type	Size	
SFML-2.5.1	2019-01-05 12:28 PM	File folder		
SFMLWeek1- Project 0	2019-01-05 12:53 PM	File folder		

| GAME2002-T163-2019 > Week1 > SFML-2.5.1			Search SFML-2.5.1	
Name	Date modified	Type	Size	
bin	2019-01-05 12:28 PM	File folder		
doc	2019-01-05 12:28 PM	File folder		
examples	2019-01-05 12:28 PM	File folder		
include	2019-01-05 12:28 PM	File folder		
lib	2019-01-05 12:28 PM	File folder		
license.md	2019-01-05 12:27 PM	MD File		
readme.md	2019-01-05 12:27 PM	MD File		

Creating and configuring a SFML project

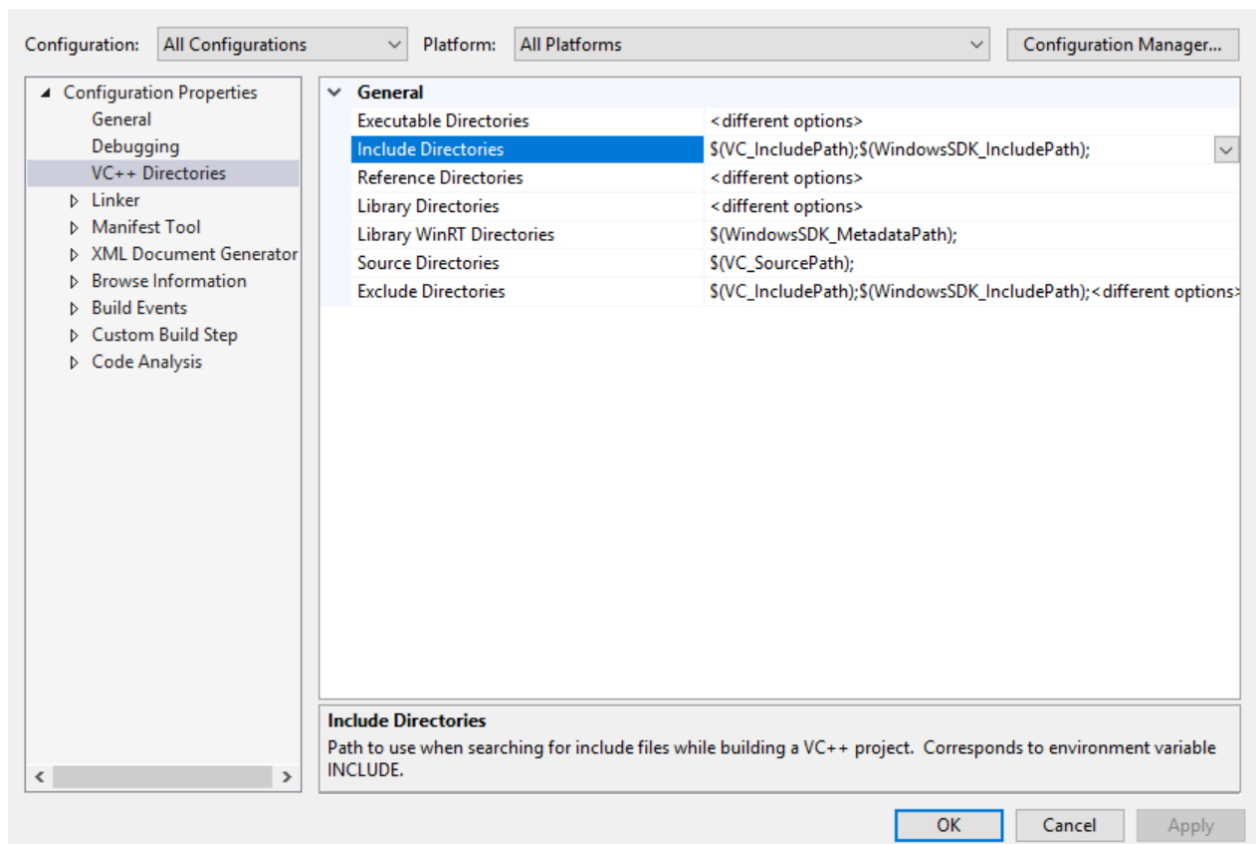1. Open Visual Studio 2017 and Create a new Visual C++ → General → Empty Project
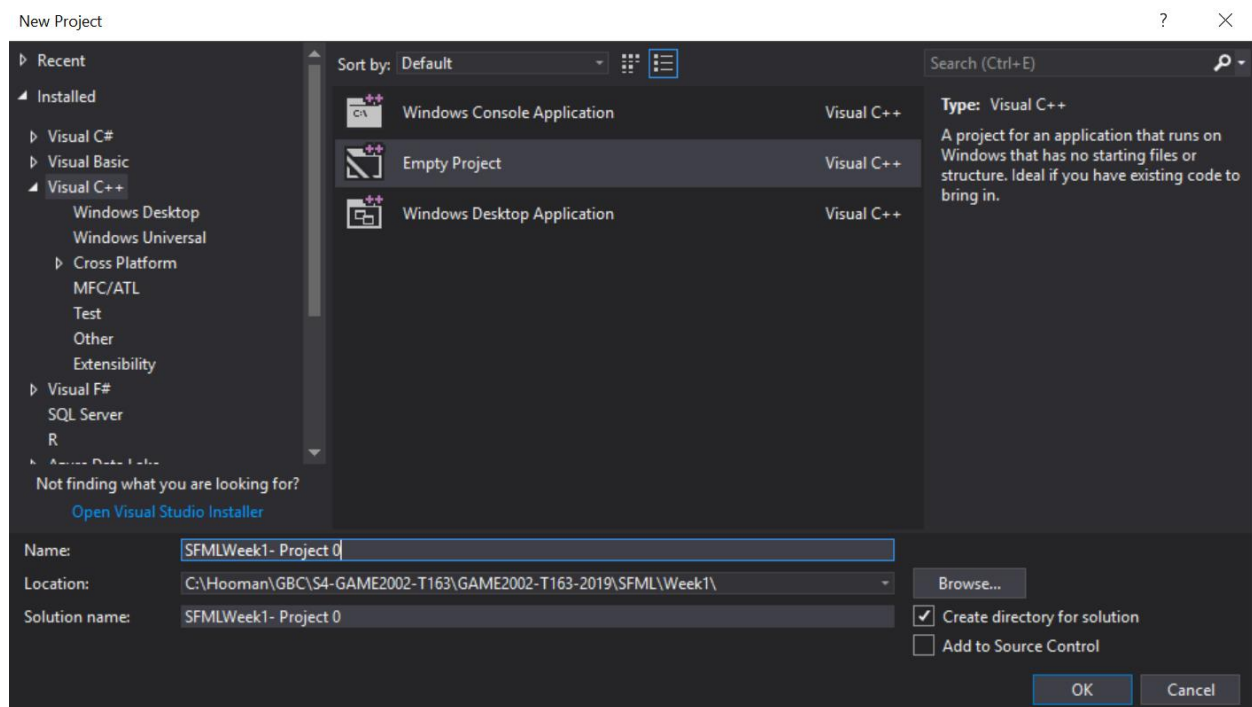
2. Right Click on the Project → Select Properties → Select "All Configuration" in Configuration combo box → Select "All Platforms" in Platform combo box:
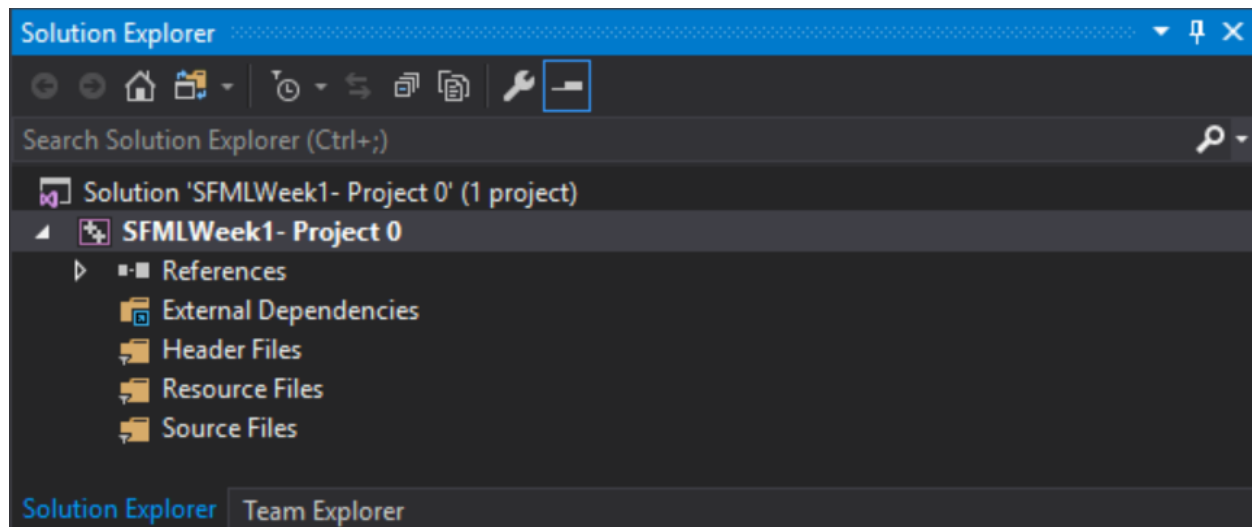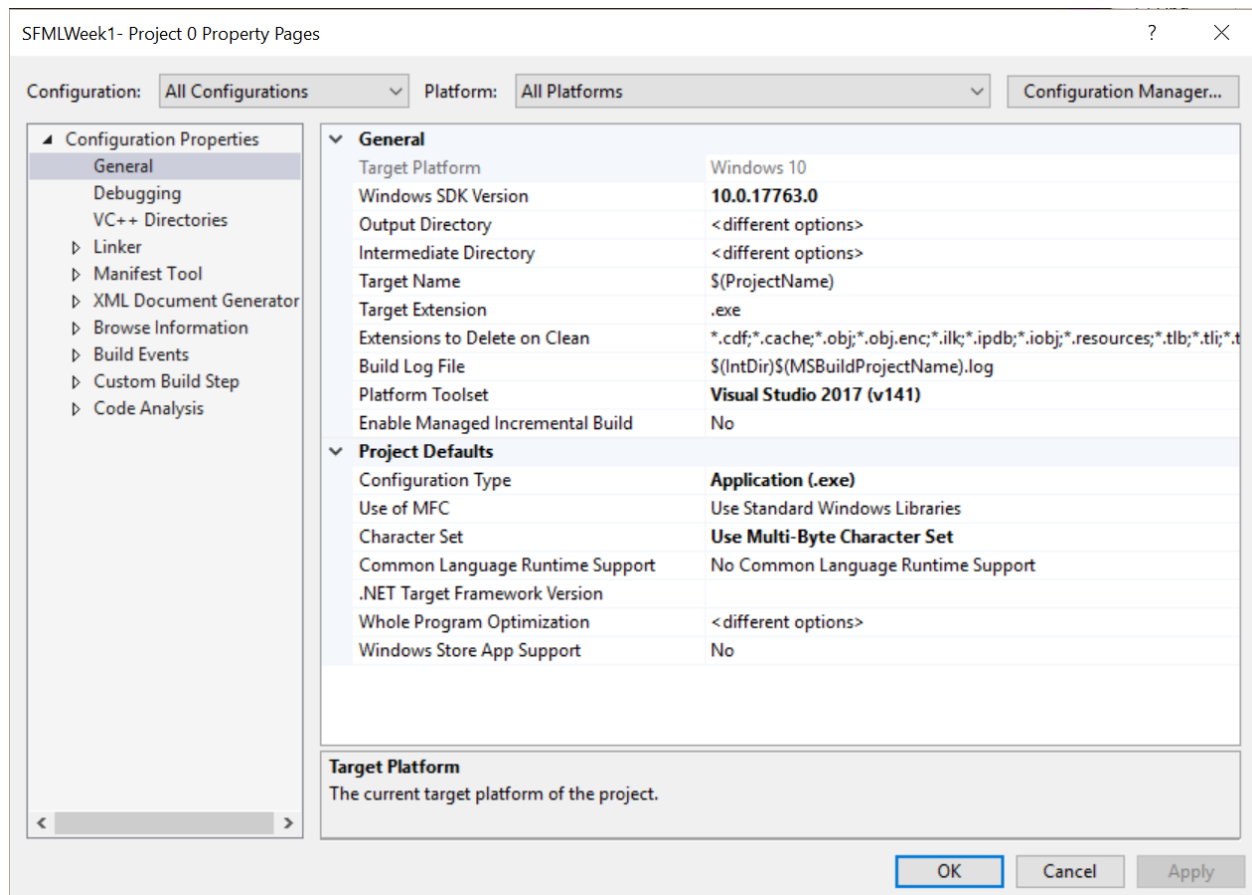
3. Under Configuration Properties , VC++ Directories → Add the sfml include folder to "Include Directories"

SFMLWeek1- Project 0 Property Pages   ?   ✕

Configuration: [All Configurations   ▼] Platform: [All Platforms   ▼] [Configuration Manager...]

◢ Configuration Properties | ∨ **General**
   General | Executable Directories   <different options>
   Debugging | Include Directories $(VC_IncludePath);$(WindowsSDK_IncludePath);   [▼]
   VC++ Directories | Reference Directories   <different options>
  ▷ Linker | Library Directories   <different options>
  ▷ Manifest Tool | Library WinRT Directories $(WindowsSDK_MetadataPath);
  ▷ XML Document Generator | Source Directories $(VC_SourcePath);
  ▷ Browse Information | Exclude Directories $(VC_IncludePath);$(WindowsSDK_IncludePath);<different options>
  ▷ Build Events |
  ▷ Custom Build Step |
  ▷ Code Analysis |

Include Directories
Path to use when searching for include files while building a VC++ project. Corresponds to environment variable
INCLUDE.

[OK]   [Cancel]   [Apply]

Include Directories   ?   ✕

$(SolutionDir)/../../SFML-2.5.1/include

Evaluated value:

E2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\/../../SFML-2.5.1/include
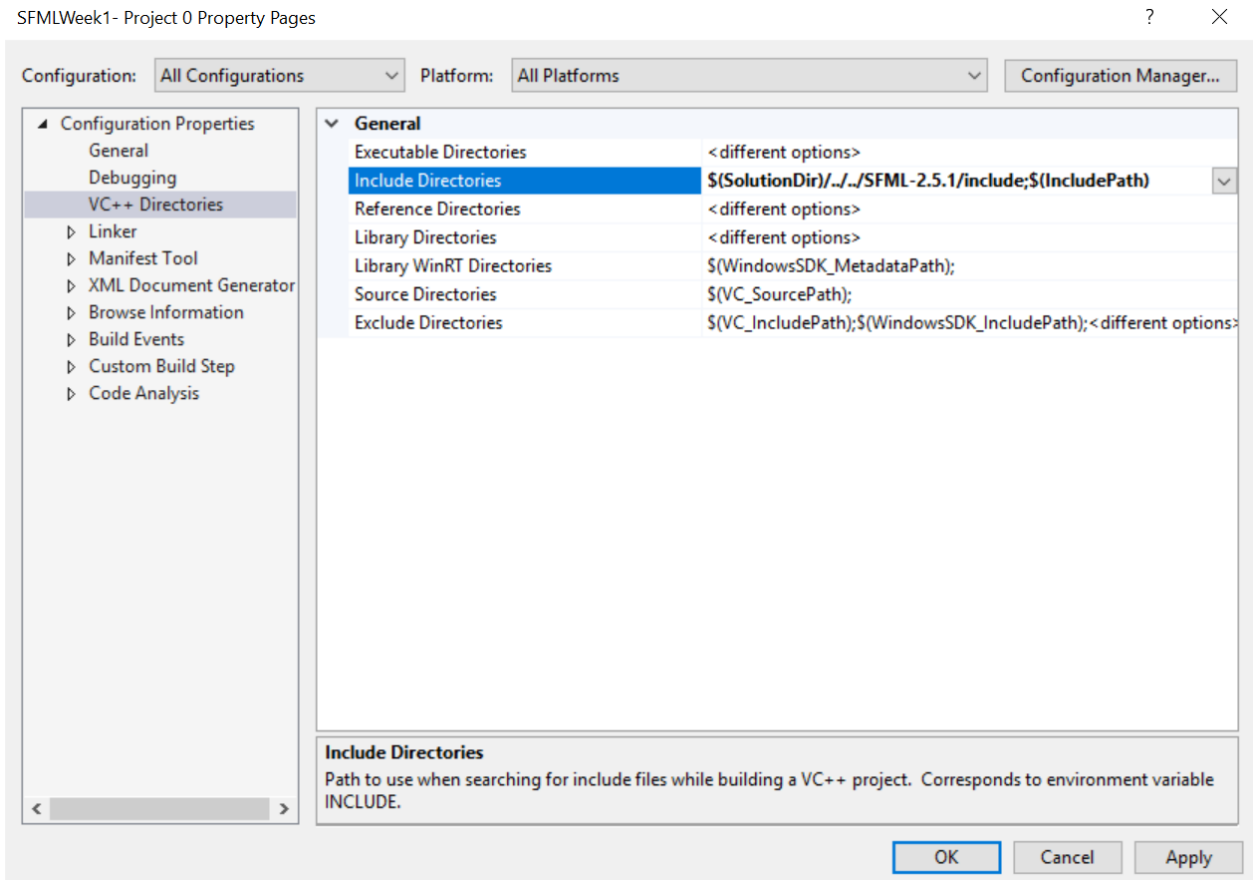
Inherited values:

$(VC_IncludePath)
$(WindowsSDK_IncludePath)
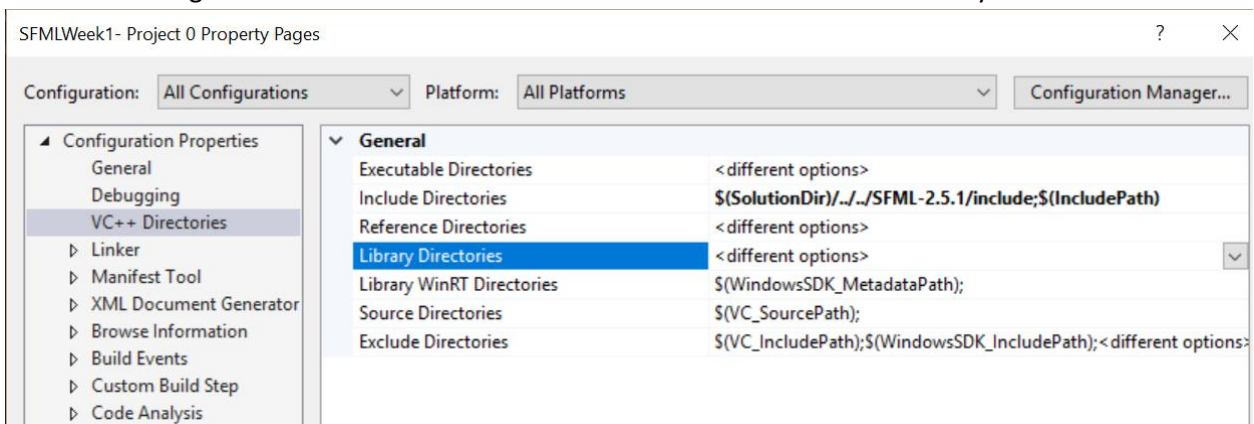
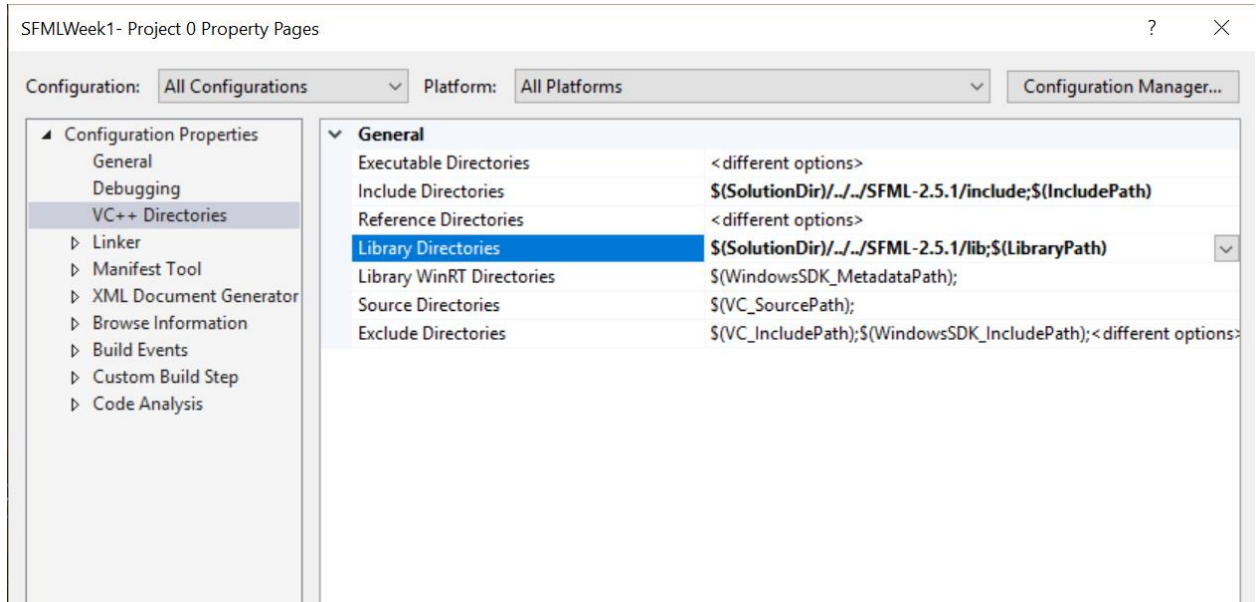☑ Inherit from parent or project defaults   Macros>>

                                              OK   Cancel

4. Go to "All Configuration" → VC++ Directories → Add the sfml lib folder to "Library Directories"
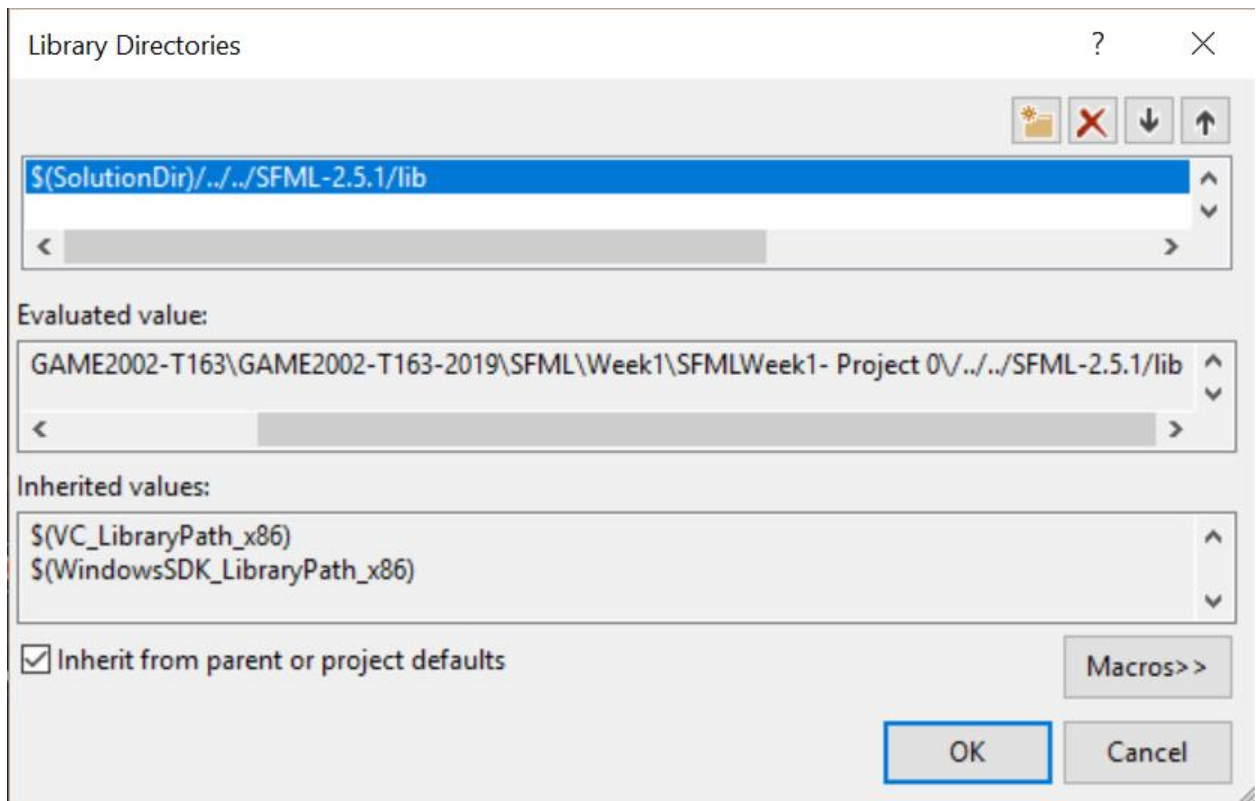
5. Choose "Debug" Configuration from "Configuration" combo box → Go to "Configuration Properties" ➔ "Linker" → "Input" → Additional Dependencies → And Add all the SFML libraries that you need, for example "sfml-graphics-d.lib", "sfml-window-d.lib" and "sfml-system-d.lib". We are going to add all the debug SFML libraries (hence -d).

- sfml-system-d.lib
- sfml-window-d.lib
- sfml-graphics-d.lib
- sfml-audio-d.lib
- sfml-network-d.lib

Additional Dependencies   ?   ✕

```
sfml-system-d.lib
sfml-window-d.lib
sfml-graphics-d.lib
sfml-audio-d.lib
sfml-network-d.lib
```

Evaluated value:

```
sfml-system-d.lib
sfml-window-d.lib
sfml-graphics-d.lib
sfml-audio-d.lib
sfml-network-d.lib
%(AdditionalDependencies)
```

Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
advapi32.lib
shell32.lib
```

☑ Inherit from parent or project defaults   Macros>>

OK   Cancel
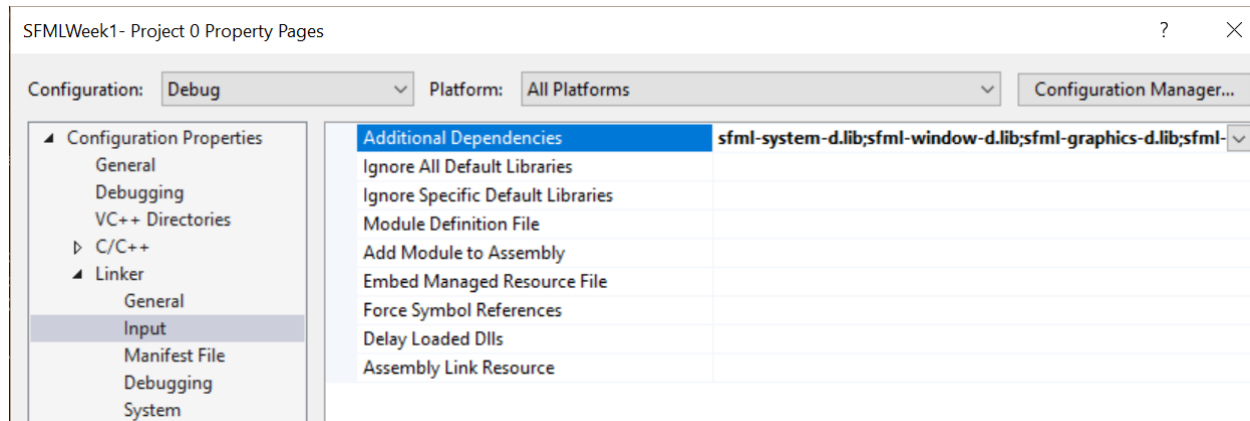
Note: It is important to link to the libraries that match the configuration: "sfml-xxx-d.lib" for Debug, and "sfml-xxx.lib" for Release.

6. Go to "Release Configuration" ➔ "Linker" → "Input" → Additional Dependencies → And Add all the SFML libraries that you need, for example "sfml-graphics.lib", "sfml-window.lib" and "sfml-system.lib".

Additional Dependencies   ?   ✕

```
sfml-system.lib
sfml-window.lib
sfml-graphics.lib
sfml-audio.lib
sfml-network.lib
```

Evaluated value:

```
sfml-system.lib
sfml-window.lib
sfml-graphics.lib
sfml-audio.lib
sfml-network.lib
%(AdditionalDependencies)
```

Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
advapi32.lib
shell32.lib
```

☑ Inherit from parent or project defaults   Macros>>

OK   Cancel
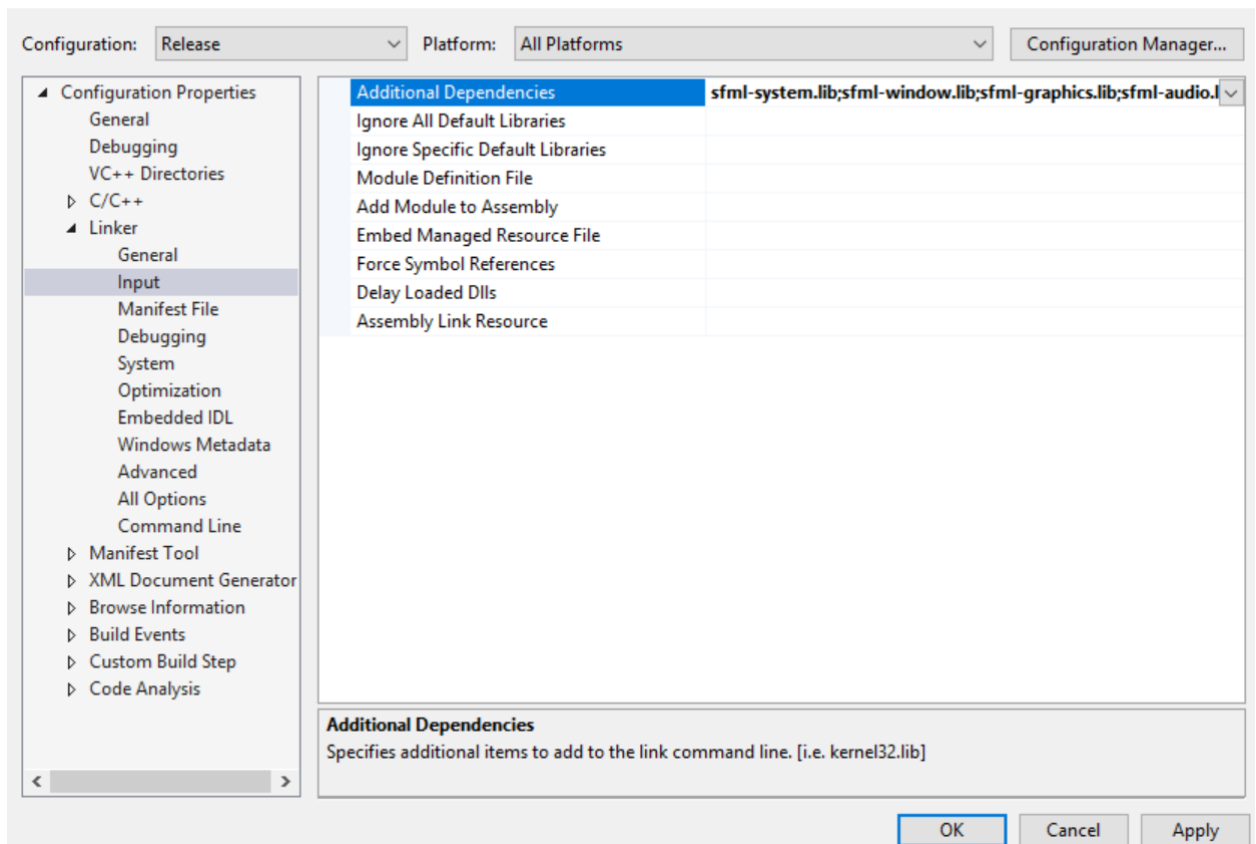
7. Make sure all the SFML dll files are copied to your project folder. Go to SFML library "bin" folder and copy everything under your project folder.

GAME2002-T163-2019 > SFML > SFML-2.5.1 > bin

Search bin

Name	Date modified	Type	Size
openal32.dll	2019-01-05 12:28 PM	Application extension	
sfml-audio-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-audio-d-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-graphics-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-graphics-d-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-network-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-network-d-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-system-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-system-d-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-window-2.dll	2019-01-05 12:28 PM	Application extension	
sfml-window-d-2.dll	2019-01-05 12:28 PM	Application extension	

8. Go to "Source Files" folder in Visual Studio "Solution Explorer" and add a C++ file, call it game.cpp
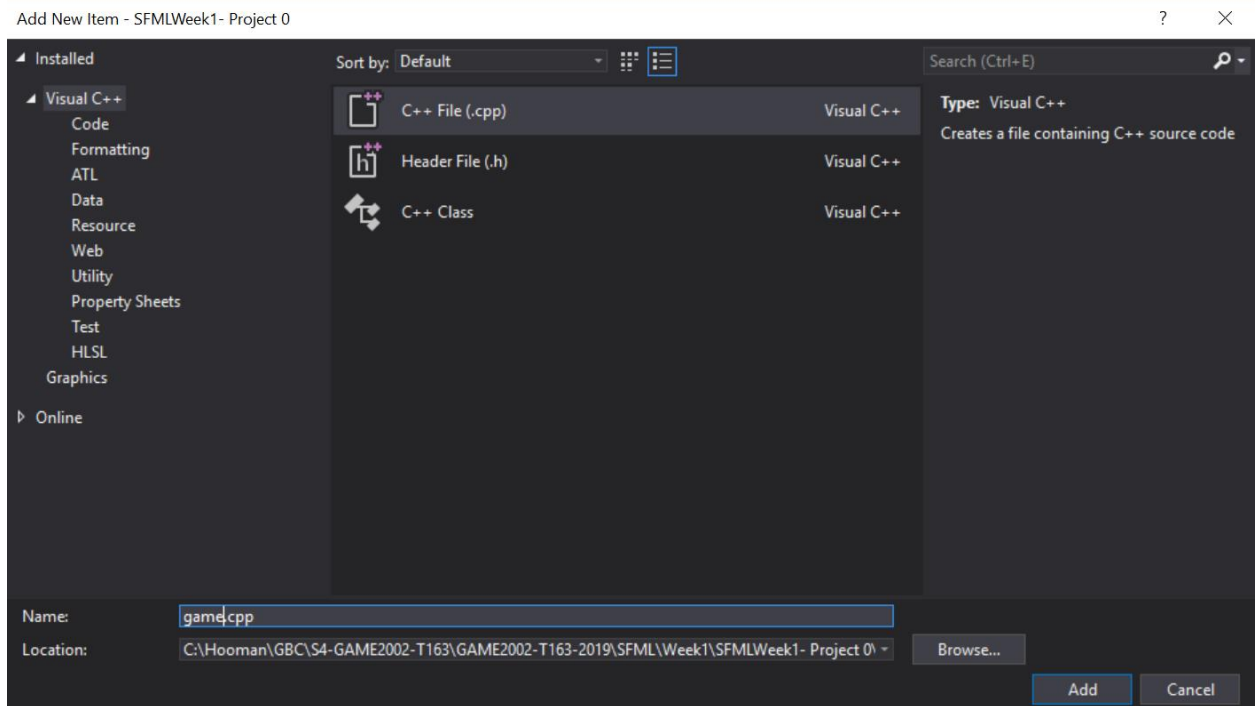
9. Copy and Paste the following code inside the game.cpp file:

```cpp
#include <SFML/Graphics.hpp>

int main()
{
    sf::RenderWindow window(sf::VideoMode(200, 200), "SFML works!");
    sf::CircleShape shape(100.f);
    shape.setFillColor(sf::Color::Green);

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
                window.close();
        }

        window.clear();
        window.draw(shape);
        window.display();
    }

    return 0;
}
```
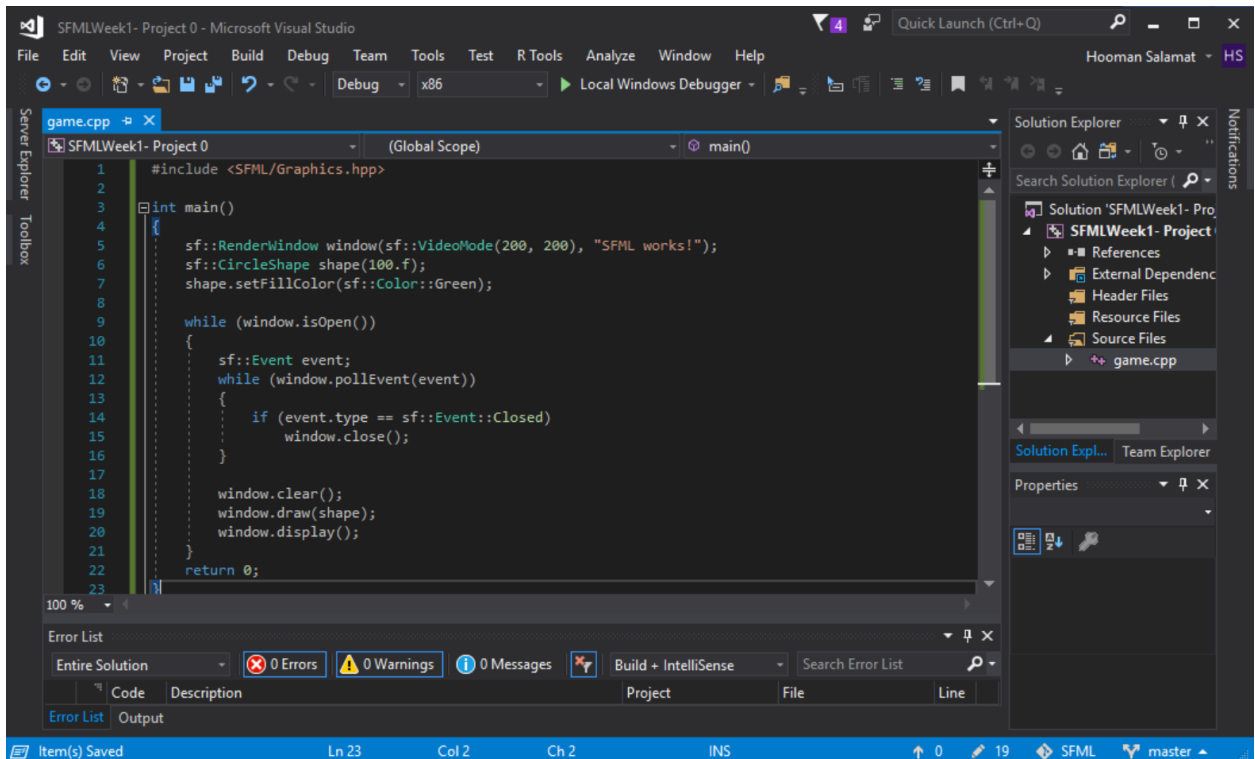
10. Compile and run the project. Oops, we are getting an error. Well, remember, we downloaded the 64-bit version of SFML library.
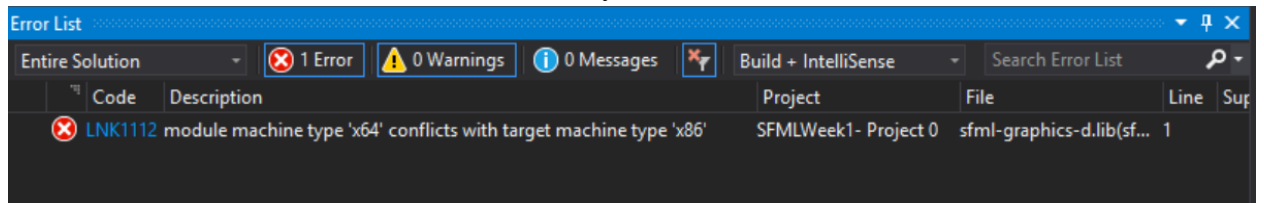
11. Now, change the target platform to 64 bit (x64)
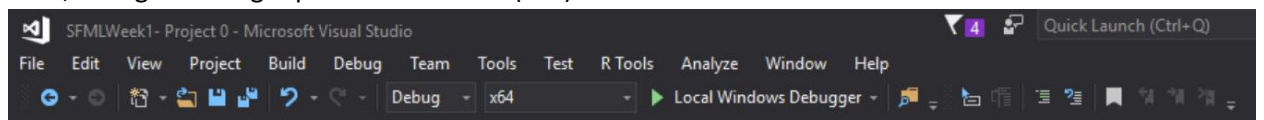
12. Now compile and run.
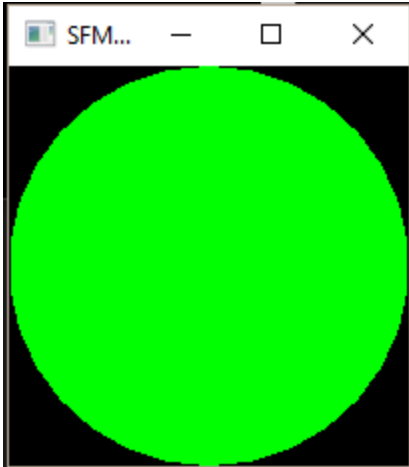
13. If you are still getting an error, because you just copy the project from github and didn't follow the above steps, check your Windows SDK version! Probably yours is different than mine. Nota problem. Just change it to the one you have!
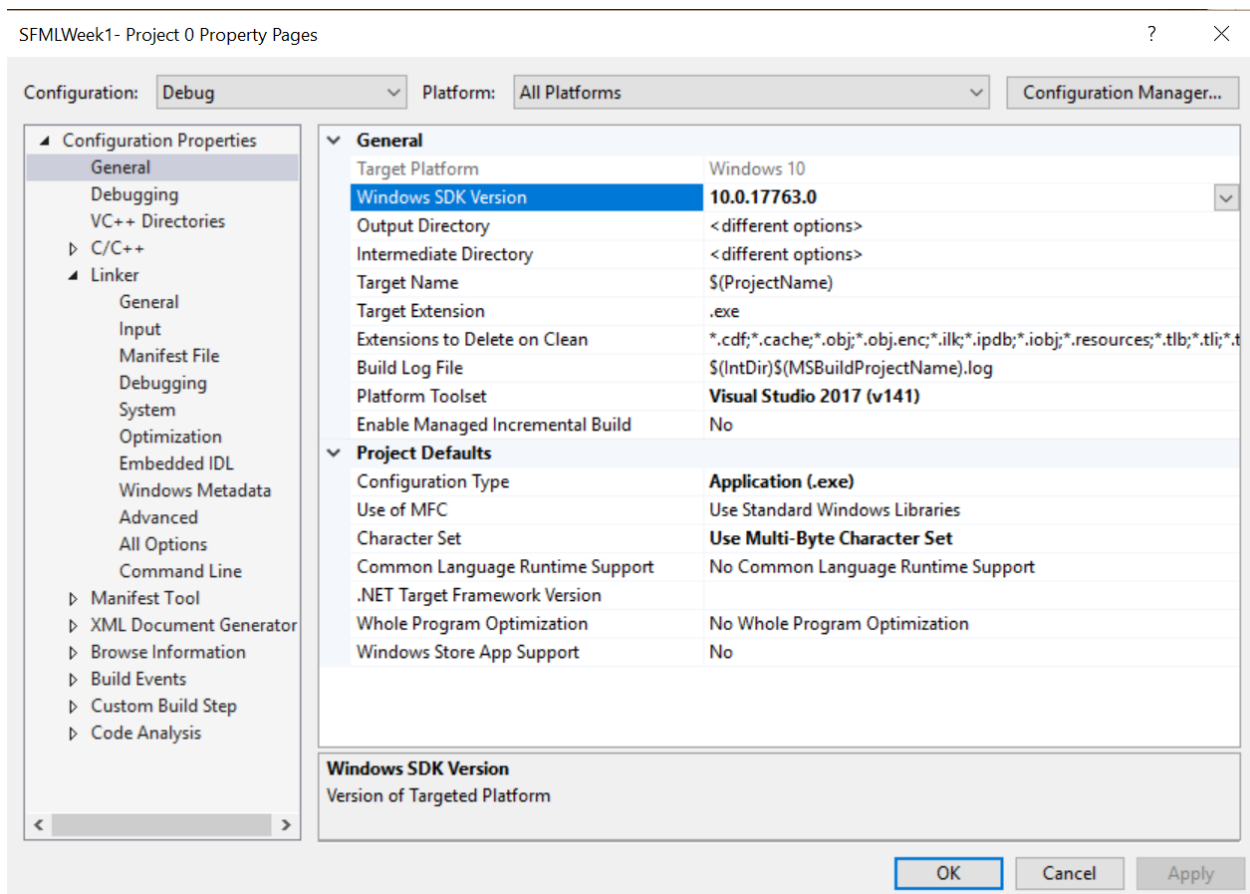
14. Since you linked to the dynamic version of SFML, you copied the SFML DLLs (they are in *<sfml-install-path/bin>*) to the project directory. You can also use a post-build action in your project, and add the commands to copy the bin folder DLLs to the "debug folder". The post-build action will be written as a batch script.

Configuration: All Configurations ∨   Platform: All Platforms ∨   Configuration Manager...

▲ Configuration Properties   Command Line
    General   Description
    Debugging   Use In Build   Yes
    VC++ Directories
    ▷ C/C++
    ▷ Linker
    ▷ Manifest Tool
    ▷ XML Document Generator
    ▷ Browse Information
    ▲ Build Events
        Pre-Build Event
        Pre-Link Event
        Post-Build Event
    ▷ Custom Build Step
    ▷ Code Analysis
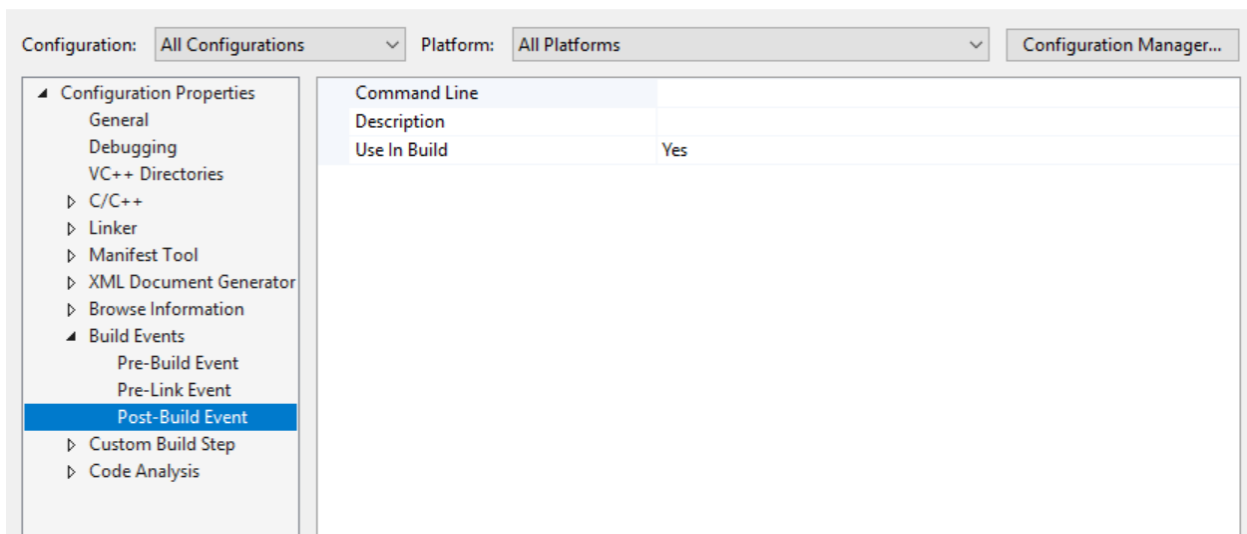
15. Create a new command line and add the following line:

xcopy "$(SolutionDir)\..\..\SFML-2.5.1\bin*.dll" "$(SolutionDir)\x64\Debug\" /S /Y <NUL:

/s	Copies folders and subfolders except for empty ones.
/y	Overwrites existing files without prompting you.

<NUL: is there to fix a bug 😊

Note: **xcopy** is an external program, while **copy** is part of the interpreter (cmd.exe, command.com). This means that **xcopy** might not be present on another machine or a rescue disk.

Command Line   ?   ✕

xcopy "$(SolutionDir)\..\..\SFML-2.5.1\bin*.dll" "$(SolutionDir)\x64\Debug\" /S /Y <NUL:

Evaluated value:

xcopy "C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project

                                                              Macros>>

                                            OK   Cancel

SFMLWeek1- Project 0 - Microsoft Visual Studio

File   Edit   View   Project   Build   Debug   Team   Tools   Test   R Tools   Analyze   Window   Help

Hooman Salamat   HS

Quick Launch (Ctrl+Q)

Debug   x64   ▶ Local Windows Debugger

game.cpp

SFMLWeek1- Project 0   (Global Scope)   main()

```
1       #include <SFML/Graphics.hpp>
2
3     □int main()
```

100 %

Solution Explorer

Search Solution Explorer (
- Solution 'SFMLWeek1- Pro...
  - SFMLWeek1- Project
    - References
    - External Dependenc
    - Header Files
    - Resource Files
    - Source Files
      - game.cpp

Solution Expl...   Team Explorer

Output

Show output from: Build

```
1>------ Rebuild All started: Project: SFMLWeek1- Project 0, Configuration: Debug x64 ------
1>game.cpp
1>SFMLWeek1- Project 0.vcxproj -> C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Proje
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\opena
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>C:\Hooman\GBC\S4-GAME2002-T163\GAME2002-T163-2019\SFML\Week1\SFMLWeek1- Project 0\\..\..\SFML-2.5.1\bin\sfml-
1>11 File(s) copied
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

Properties

SFMLWeek1- Project 0 Pro

- Misc
  (Name)   SFMLWeek1-
  Project Dep
  Project File C:\Hooman\

(Name)
Specifies the project name.

Error List   Command Window   Output

Ready   ↑ 0   60   SFML   master