GP2 Frameworks – Setup

Introduction

During Graphics Programming 2 we will be using two different frameworks.
The frameworks are setup to built for x64 processor architectures, using the Visual Studio v143 Platform Toolset (This means you'll need Visual Studio 2022 in order to compile the frameworks).

1. **PhysXFramework** → A simple framework that we will be using to get used to the PhysX API
2. **Overlord Engine** → A more complex framework that you will expand with various Graphics Programming Techniques, and ultimately used for your end project.

Both frameworks rely on a set of third-party libraries such as:

1. **DirectX** (part of the Windows SDK)
2. **DirectXTex** (helper library to convert different image formats into textures)
3. **FX11** (helper library that mimics the DX10 effect system for DX11)
4. **FMOD** (FMOD 1.10, a library that helps us with sound and sound effects)
5. **PhysX** (Nvidia PhysX 4.1, a library for accurate physics simulations)
6. **Vld** (Visual Leak Detector, memory leak detection library)
7. **Imgui** (A lightweight and super useful UI framework for debugging purposes)

All these third-party libraries come with different files (includes, libraries, binaries) that need to be included in the frameworks mentioned on top. Depending on the build setup used for the frameworks, a matching set of library builds need to be created.

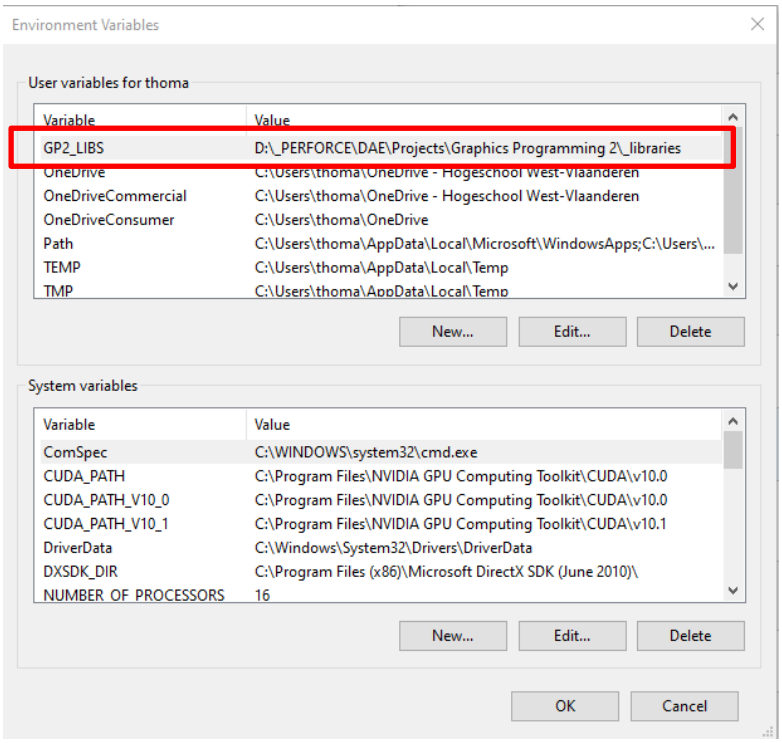
The total (uncompressed) size of all these libraries (including debug, release and profile versions) is close to 500MB. So, it wouldn't be very useful to copy all the libraries each time you create a new project with one of the GP2 frameworks. The ideal setup is to have a single instance of these libraries somewhere on your local drive and setup a mechanism that Visual Studio knows where to find these required library files.

Making sure that Visual Studio (or any other IDE) knows where to find the necessary files to build the frameworks can be done by setting an 'Environment Variable' on your system. This environment variable can be set on a system or user level and contains the absolute path to our library directory. Programs on your PC (such as Visual Studio) can easily retrieve and evaluate such values.

The Environment Variable that we want to set is called GP2_LIBS and will contain the absolute path to the root of our GP2 libraries folder.

Setting up the environment

The only thing necessary to be able to correctly build the GP2 frameworks, is the creation of the GP2_LIBS environment variable.



1. Extract the **GP2_Libraries.rar** file somewhere on a logical place on your local hard drive
2. Run the **Set-GP2_LIBS-EnvVariable.bat** script which is located inside the libraries folder. This will automatically create the **GP2_LIBS** variable on a user-level using the current directory (so if you move the libraries folder somewhere else, make sure to run the script again).

   (Optional) You can also set this variable manually (GP2_LIBS), or even as a system variable (in case you are using multiple accounts on your device that need this variable)

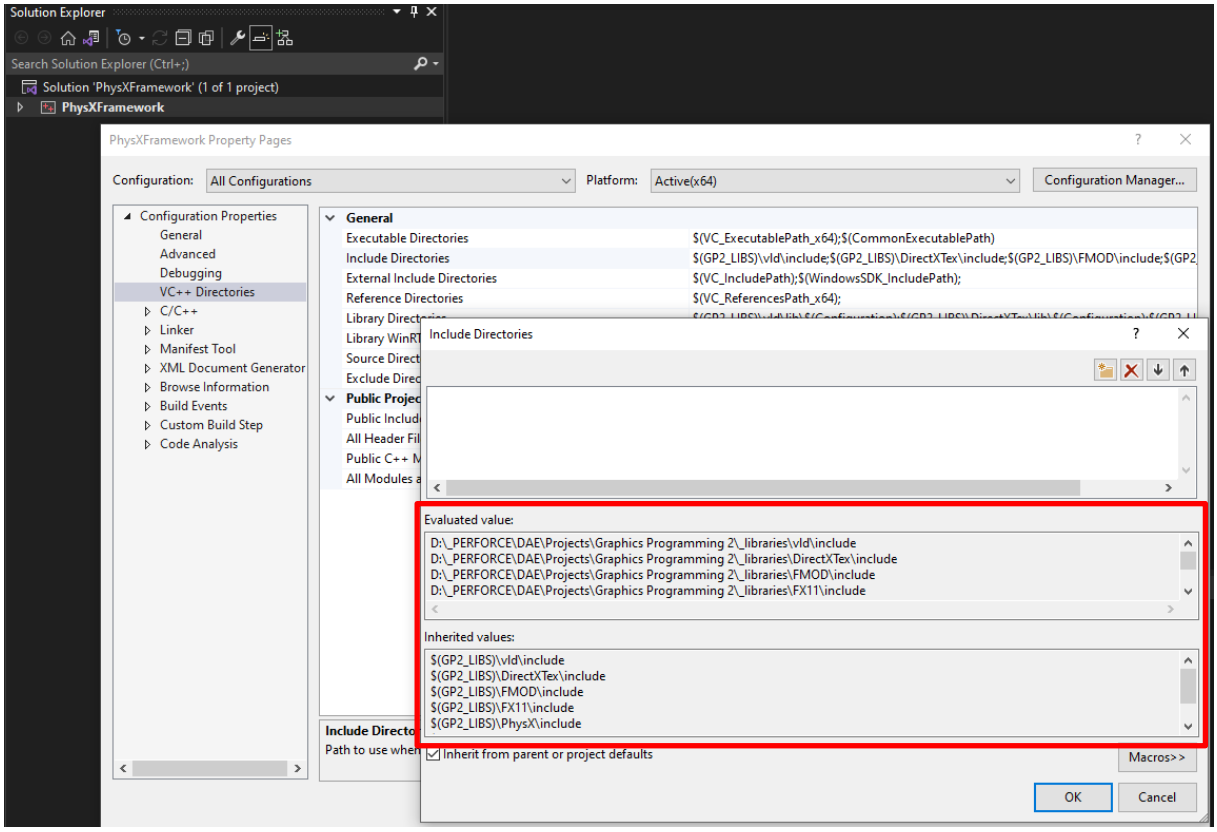
3. Extract a GP2 Framework and verify that it builds correctly in debug and release mode.

So, you only must perform these steps once. The moment this variable is created (and points the correct library root folder) you should be able to build any instance of the GP2 frameworks (without creating additional copies of these library files).

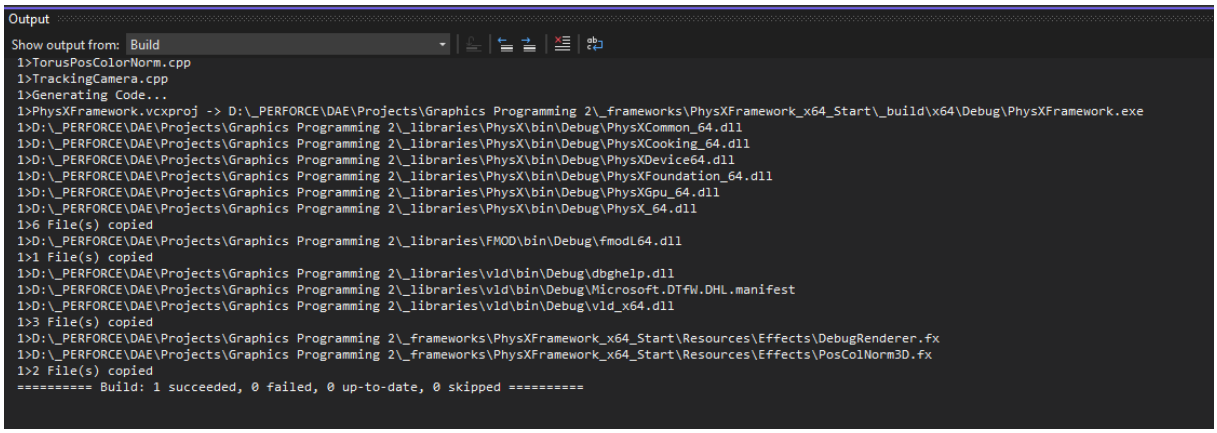
This also means that whenever you must share your project, do not sent the library files along! If the other person has the GP2_LIBS variable, the project should compile just fine without any (library related) issues.

How It Works

If you have a look at the Property Pages of one of the frameworks, you'll see that the location of all the necessary directories to build the project (include, library, … directories) is using this GP2_LIBS environment variable. And automatically evaluated as the absolute path to the root directory of the required libraries.



Also, the frameworks will automatically copy the required **binaries** (DLL) and **resource folder** to the build directory (for both Debug and Release).



That's it, see you in class and enjoy the ride!