Compiling Vulkan from command line

I am a boomer, and I compile my programs from the command line. This is how I was raised, and this is what I am more comfortable with. You are strongly encouraged to use an IDE and more modern development approaches. However, in this folder, I have put the scripts with which I was able to successfully compile the code in the command line environment under Windows, MacOS and Linux.

Windows

Under Windows, you must locate where you have put both Vulkan and GLFW. Then, replace in the script `compileWin.bat` the four directories color coded below, with the correct locations.

```
cl /MD /I C:\VulkanSDK\glfw-3.3.9.bin.WIN64\include
/I .\headers\ /I C:\VulkanSDK\1.3.275.0\Include /std:c++20
/EHsc %1.cpp gdi32.lib opengl32.lib kernel32.lib user32.lib
shell32.lib glfw3.lib vulkan-1.lib /link
/LIBPATH:C:\VulkanSDK\glfw-3.3.9.bin.WIN64\lib-vc2022
/LIBPATH:C:\VulkanSDK\1.3.275.0\Lib
```

In the command prompt you then simply call the batch script passing the name of the assignment. For example, for assignment one:

```
> compileWin A01
```

Linux

Under Linux, you probably do not need to modify the script, since the installation should have already correctly set the environment for you.

```
g++ -std=c++17 -Iheaders -O2 -o $1 $1.cpp -lglfw -lvulkan -
ldl -lpthread -lX11 -lXrandr
```

However, you should first make the script executable using the corresponding command:

```
> chmod u+x compileLinux
```

In the command prompt you then simply call the command script passing the name of the assignment. Remember to add "./" before to help finding the command. For example, for assignment one:

```
> ./compileLinux A01
```

MacOS

Under MacOS, you probably have just to locate the directory where Vulkan was installed, and properly patch the file path in the color-coded area below. In my case, GLFW was able to correctly setup its path in the development environment, while Vulkan SDK did not succeed.

```
export VP=/Users/gribaudo/VulkanSDK/1.3.239.0/macOS

g++ -std=c++17 -Iheaders -I$VP/include/ -L/usr/local/lib
-lglfw -L$VP/lib -lvulkan $1.cpp -o $1
```

Then, you should make the script executable using the corresponding command:

```
> chmod u+x compileMac
```

In the command prompt you then simply call the command script passing the name of the assignment. As for Linux, remember to add "./" before to help finding the command. For example, for assignment one:

```
> ./compileMac A01
```

Concluding remarks

The pervious commands work in my environments: unfortunately, there are too many ways to setup a development environment, and the commands above might not work for you. However, I hope they can be helpful at least in giving you some hints on how to setup your IDE for running the assignments.

Also, note that all the versions add the folder "`headers/`" to the path where the compiler looks for the included headers. That is the folder where header-only libraries, such as `GLM` for vector-matrix computations, or `tinyObj` for 3D assets loading are stored.