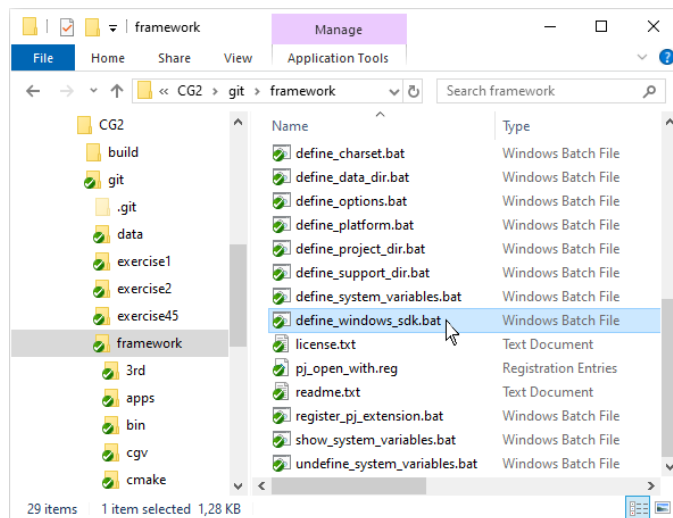


CGV Framework

Setup & Building

Step 1 – setup the environment

- Navigate to the directory `framework` in the repository root
- You will find several Windows batch scripts that take care of the setup (they basically set required environment variables)
- First, you have to select which of the installed Windows SDKs to use:



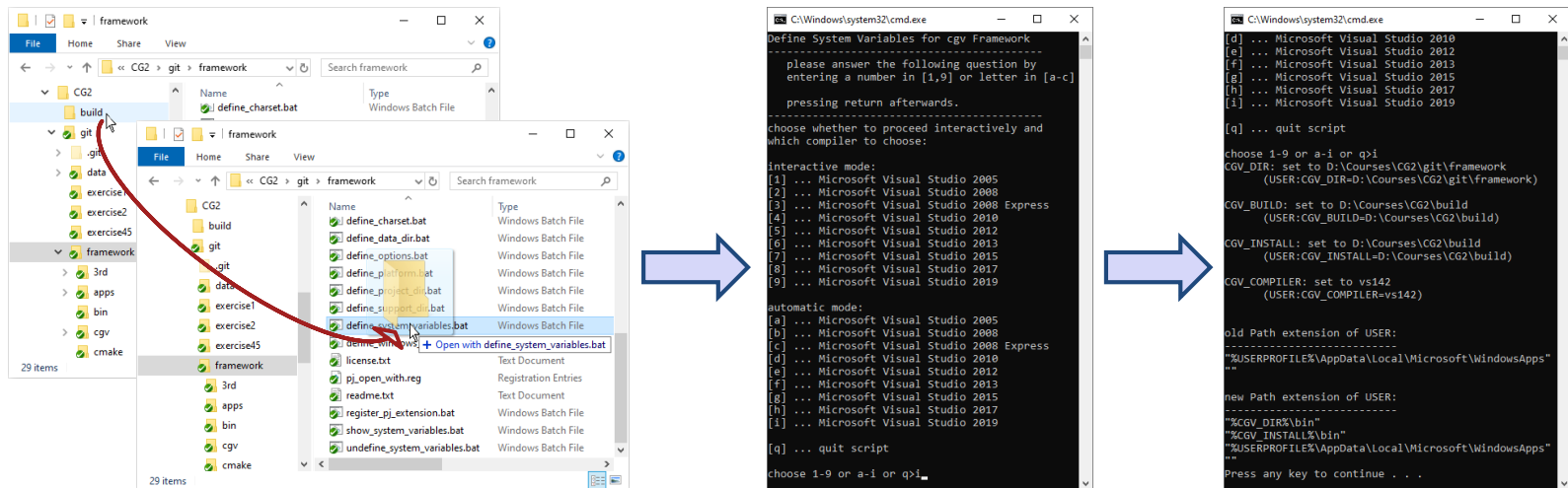
```
C:\Windows\system32\cmd.exe

current windows sdk version: 10.0.18362.0
-----
installed windows sdk versions:
[0] ... 10.0.10240.0
[1] ... 10.0.18362.0
-----
choose version [0-1] or press enter to select 10.0.18362.0:
pressed enter
(USER:CGV WINDOWS_SDK=10.0.18362.0)
Press any key to continue . . .
```

- If you hit enter without selecting anything at the prompt, the newest version will be chosen

Step 1 – setup the environment

- Next, you have to provide a build folder and select your Visual Studio Version (both are done in a single step)
- Just drag your build folder onto the script `define_system_variables.bat`:



The first screenshot shows a file explorer window with the 'framework' directory selected. The 'build' folder is being dragged onto the 'define_system_variables.bat' file. The second screenshot shows the command prompt running the script, which prompts for a Visual Studio version choice. The third screenshot shows the command prompt displaying the resulting environment variables.

```
Define System Variables for cgV Framework

please answer the following question by
entering a number in [1,9] or letter in [a-c]

pressing return afterwards.

-----
choose whether to proceed interactively and
which compiler to choose:

interactive mode:
[1] ... Microsoft Visual Studio 2005
[2] ... Microsoft Visual Studio 2008
[3] ... Microsoft Visual Studio 2008 Express
[4] ... Microsoft Visual Studio 2010
[5] ... Microsoft Visual Studio 2012
[6] ... Microsoft Visual Studio 2013
[7] ... Microsoft Visual Studio 2015
[8] ... Microsoft Visual Studio 2017
[9] ... Microsoft Visual Studio 2019

automatic mode:
[a] ... Microsoft Visual Studio 2005
[b] ... Microsoft Visual Studio 2008
[c] ... Microsoft Visual Studio 2008 Express
[d] ... Microsoft Visual Studio 2010
[e] ... Microsoft Visual Studio 2012
[f] ... Microsoft Visual Studio 2013
[g] ... Microsoft Visual Studio 2015
[h] ... Microsoft Visual Studio 2017
[i] ... Microsoft Visual Studio 2019
[q] ... quit script

choose 1-9 or a-i or q>i_
```

```
[d] ... Microsoft Visual Studio 2010
[e] ... Microsoft Visual Studio 2012
[f] ... Microsoft Visual Studio 2013
[g] ... Microsoft Visual Studio 2015
[h] ... Microsoft Visual Studio 2017
[i] ... Microsoft Visual Studio 2019
[q] ... quit script

choose 1-9 or a-i or q>i_

CGV_DIR: set to D:\Courses\CG2\git\framework
(USER:CGV_DIR=D:\Courses\CG2\git\framework)

CGV_BUILD: set to D:\Courses\CG2\build
(USER:CGV_BUILD=D:\Courses\CG2\build)

CGV_INSTALL: set to D:\Courses\CG2\build
(USER:CGV_INSTALL=D:\Courses\CG2\build)

CGV_COMPILER: set to vs142
(USER:CGV_COMPILER=vs142)

old Path extension of USER:
"%USERPROFILE%\AppData\Local\Microsoft\WindowsApps"
""

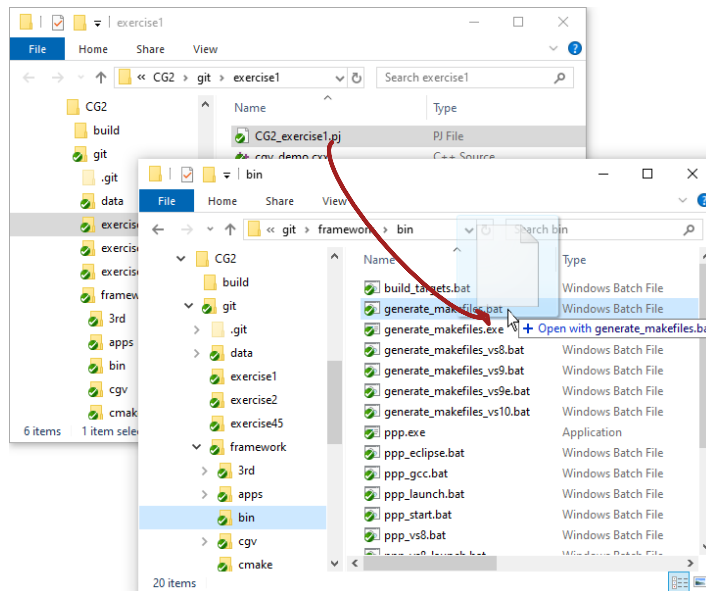
new Path extension of USER:
-----
"%CGV_DIR%\bin"
"%CGV_INSTALL%\bin"
"%USERPROFILE%\AppData\Local\Microsoft\WindowsApps"
""

Press any key to continue . . .
```

- Choose your version of Visual Studio from the options under „automatic mode“ – for this tutorial, we use Visual Studio 2019, so here the choice was „i“

Step 2 – generate the Visual Studio solution

- The Framework is now set up – next, we generate a VS solution for exercise 1
- For this, it's easiest if you open two explorer windows: one should show the folder exercise1 from the repository root, the other the folder framework\bin
- Drag the “project file” CG2_exercise1.pj onto the batch script generate_makefiles.bat:



```
C:\Windows\system32\cmd.exe
generating makefiles in D:\Courses\CG2\build\vs142\cgv_math
generating makefiles in D:\Courses\CG2\build\vs142\glew
generating makefiles in D:\Courses\CG2\build\vs142\jpeg
generating makefiles in D:\Courses\CG2\build\vs142\zlib
generating makefiles in D:\Courses\CG2\build\vs142\glslu
generating makefiles in D:\Courses\CG2\build\vs142\cgv_type
generating makefiles in D:\Courses\CG2\build\vs142\png
generating makefiles in D:\Courses\CG2\build\vs142\ftlk
generating makefiles in D:\Courses\CG2\build\vs142\cgv_reflect
generating makefiles in D:\Courses\CG2\build\vs142\cgv_ppp
generating makefiles in D:\Courses\CG2\build\vs142\cgv_data
generating makefiles in D:\Courses\CG2\build\vs142\cgv_base
generating makefiles in D:\Courses\CG2\build\vs142\cgv_os
generating makefiles in D:\Courses\CG2\build\vs142\cgv_media
generating makefiles in D:\Courses\CG2\build\vs142\cg_icons
generating makefiles in D:\Courses\CG2\build\vs142\cgv_gui
generating makefiles in D:\Courses\CG2\build\vs142\cgv_render
generating makefiles in D:\Courses\CG2\build\vs142\cgv_reflect_types
generating makefiles in D:\Courses\CG2\build\vs142\cgv_viewer
generating makefiles in D:\Courses\CG2\build\vs142\cg_ext
generating makefiles in D:\Courses\CG2\build\vs142\cgv_gl
generating makefiles in D:\Courses\CG2\build\vs142\cg_fltk
generating makefiles in D:\Courses\CG2\build\vs142\crg_stereo_view
generating makefiles in D:\Courses\CG2\build\vs142\crg_light
generating makefiles in D:\Courses\CG2\build\vs142\CG2_exercise1
successfully processed script D:\Courses\CG2\git\framework\make\ppp\templates\make.ppp
to open solution, press any key (otherwise close window)
Press any key to continue . . .
```

Step 3 – build the code

- If you hit any key in the previous console window, Visual Studio will open automatically. Otherwise, you can find your solution inside your build folder; e.g. `build\vs142\CG2_exercise1` in case of Visual Studio 2019 and 32-bit builds
- Before you build the code, make sure either of the two „... Dll“ configurations is selected
- The other two build types („Debug/Release“ and „Debug/Release Exe“) are for advanced usage and outside of the scope of the exercise
- Done! You can now build the solution.

