Bremen

CG
VR

Advanced Computer Graphics

Tutorium 1

Assignment Sheets

- **C++ hint sheets**
  - If you are not familar with C++ yet, there are some hint sheets from the course „Computergrafik 1“ which describes some relevant difference to Java (german):
    - https://cgvr.informatik.uni-bremen.de/teaching/cg1/uebungen/cpp_hints02.pdf
    - https://cgvr.informatik.uni-bremen.de/teaching/cg1/uebungen/cpp_hints03.pdf
      - Some parts of this sheet might not be that relevant for this course, since we focus more on raytracing.

Assignment Sheets

- **Assignment Sheet 1**
  - Already published

    (see website https://cgvr.cs.uni-bremen.de)

Instructions: Programming tasks

- **Prerequisites**:
  1. An **IDE** of your choice
  2. A **C++-compiler** of your choice
  3. **CMake**
  4. **OpenGL**

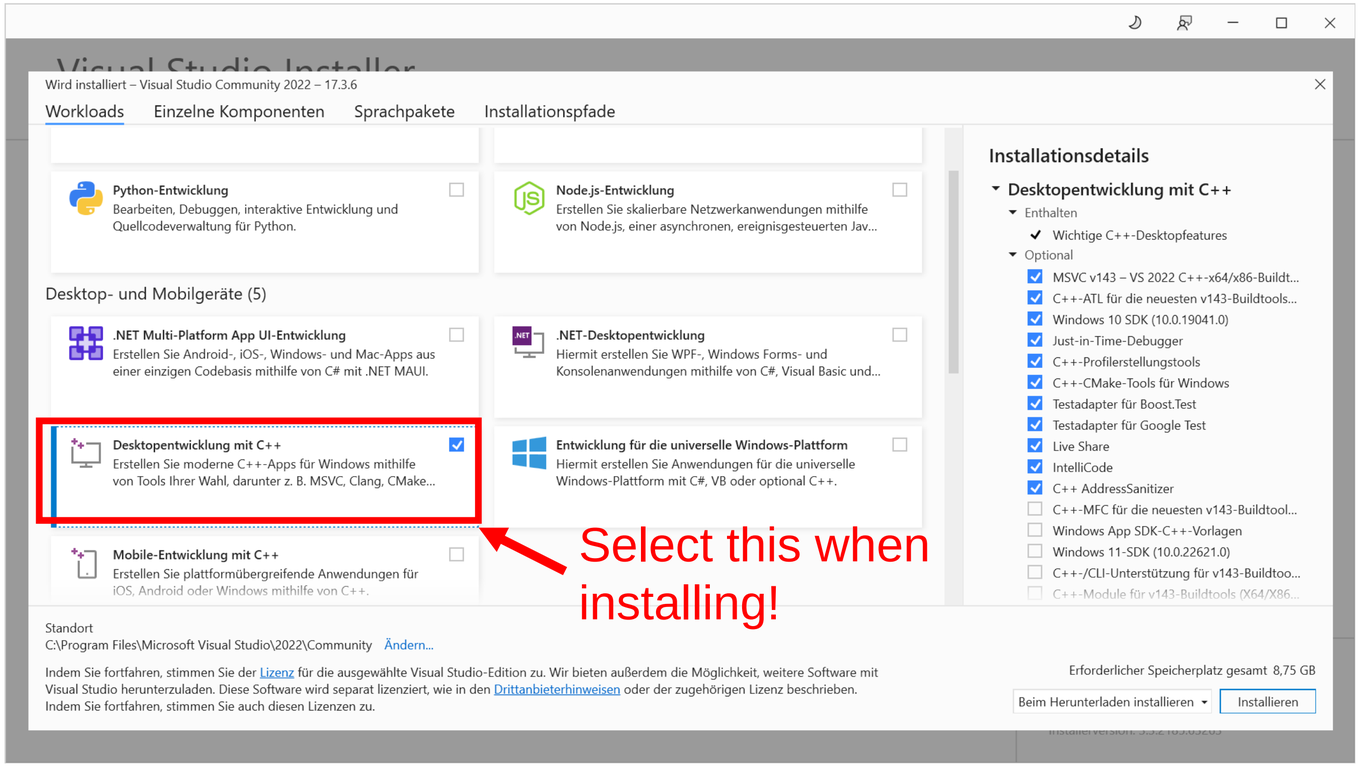
Instructions: Programming tasks

1. **Possible IDEs:**
   - Visual Studio *(Windows)*
     - https://visualstudio.microsoft.com/de/downloads/

Instructions: Programming tasks

1. Possible IDEs:

- Visual Studio *(Windows)*
  - https://visualstudio.microsoft.com/de/downloads/



Instructions: Programming tasks

1. **Possible IDEs:**
   - Visual Studio *(Windows)*
     - https://visualstudio.microsoft.com/de/downloads/
   - Xcode *(Mac)*
     - https://developer.apple.com/xcode/

Instructions: Programming tasks

1. Possible IDEs:

- Visual Studio *(Windows)*
  - https://visualstudio.microsoft.com/de/downloads/
- Xcode *(Mac)*
  - https://developer.apple.com/xcode/

At the first start opens:

These two checkboxes are sufficient

At the first start Install SDKs

Instructions: Programming tasks

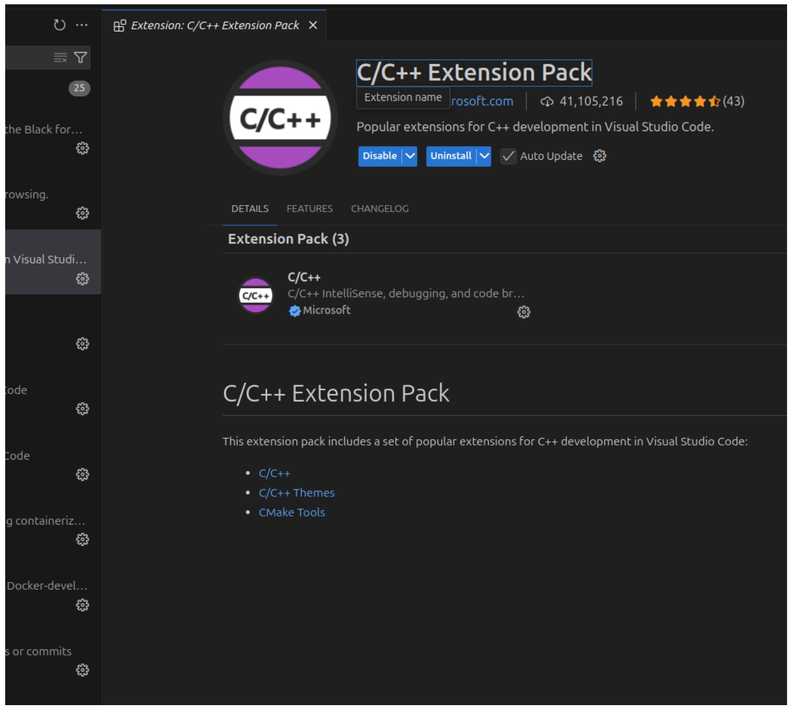
1. Possible IDEs:

- Visual Studio *(Windows)*
  - https://visualstudio.microsoft.com/de/downloads/
- Xcode *(Mac)*
  - https://developer.apple.com/xcode/
- Visual Studio Code (all platforms, recommended)
  - https://code.visualstudio.com/
  - We recommend the “C++-Entension Pack” for easier development

Instructions: Programming tasks

1. Possible IDEs:

- Visual Studio *(Windows)*
  - https://visualstudio.microsoft.com/de/downloads/
- Xcode *(Mac)*
  - https://developer.apple.com/xcode/
- Visual Studio Code (all platforms, recommended)
  - https://code.visualstudio.com/
  - We recommend the “C++-Entension Pack” for easier development



Instructions: Programming tasks

1. Possible IDEs:

- Visual Studio *(Windows)*
  - https://visualstudio.microsoft.com/de/downloads/
- Xcode *(Mac)*
  - https://developer.apple.com/xcode/
- Visual Studio Code (all platforms, recommended)
  - https://code.visualstudio.com/
  - We recommend the “C++-Entension Pack” for easier development
- Other (including Linux): QTCreator, CodeLite, Code::Blocks, etc.

Instructions: Programming tasks

2. Possible C++ Compiler:

- MSVC *(Windows)*
  - Already included in Visual Studio on Windows
- Clang *(Mac / Linux):*
  - Already included in Xcode on the Mac (AppleClang).
- GCC *(Linux & other platform):*
  - Installation by using a package manager (Ubuntu): sudo apt-get install gcc (is usually preinstalled on many Linux distributions)

Instructions: Programming tasks

3. CMake

- A quite simple tool: It generates project files for any IDEs / compilers from the information of a `CMakeLists.txt`.
  - This way it is possible that we can provide a single project for different IDEs and compilers with the same settings.
    - ➢ So by using CMake, you can use the IDE and the Operating System of your choice.
  - We provide the `CMakeLists.txt` in each C++ project.

Instructions: Programming tasks

3. CMake (Installation)

- Windows:
  - Download and install from: https://cmake.org/
- Mac:
  - By using Homebrew:
  - `If Homebrew is not installed yet, execute the following command in the terminal:`
    - `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
  - `Then install via Homebrew:`
    - `brew install --cask caskroom/cask/cmake cmake`
- Linux:
  - By using a package manager: `sudo apt-get install cmake`

Instructions: Programming tasks

4. OpenGL

- Windows / Mac: Already installed when you install the graphics card driver (e.g. Nvidia, Intel,…)

- Linux:
  - Installation using a package manager (Ubuntu): `sudo apt install mesa-utils`

Instructions: Programming tasks

5. OpenMP (only MacOs)

Run the following two commands in your home directory (~/)

- `curl -O https://mac.r-project.org/openmp/openmp-14.0.6-darwin20-Release.tar.gz`
- `sudo tar fvxz openmp-14.0.6-darwin20-Release.tar.gz -C /`

(https://mac.r-project.org/openmp/)

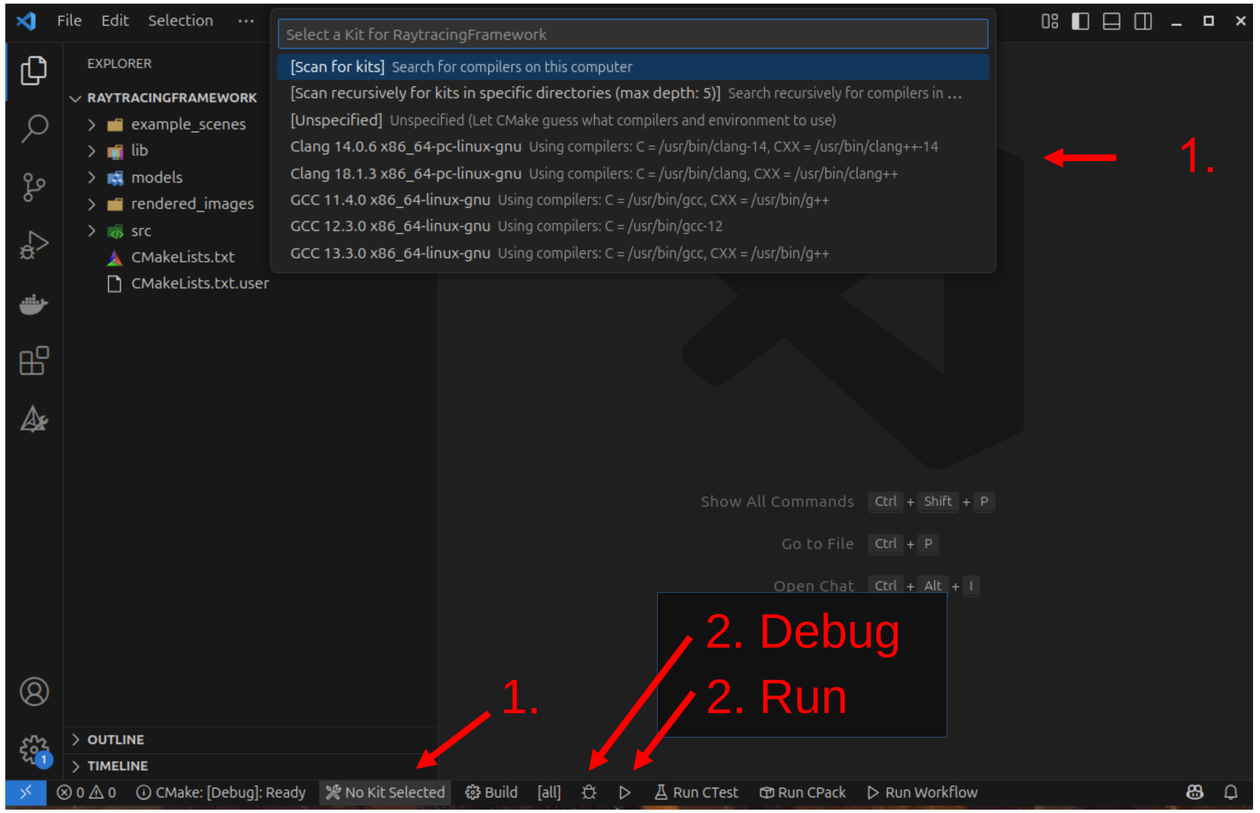
Instructions: Programming tasks

Once you have everything installed, you can download the sample project from the website and configure it with CMake - see the video on the webpage [Video Walkthrough Windows]

Instructions: Programming tasks

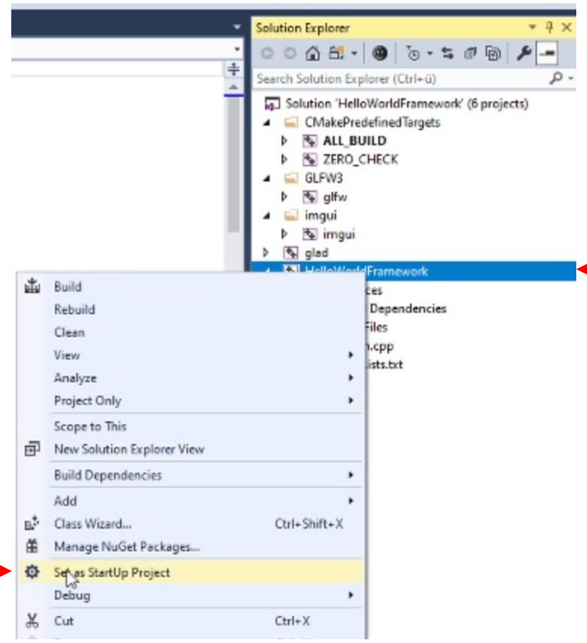
- **Note on compiling and running**
  - In VSCode:

    1. Select the compilation kit
    2. Run / Debug with the Cmake-Kit
    3. - Alternatively use launch-configurations



Instructions: Programming tasks

- **Note on compiling and running**
  - In Visual Studio:
    - You have to set the Framework as startup project, therefore do this:

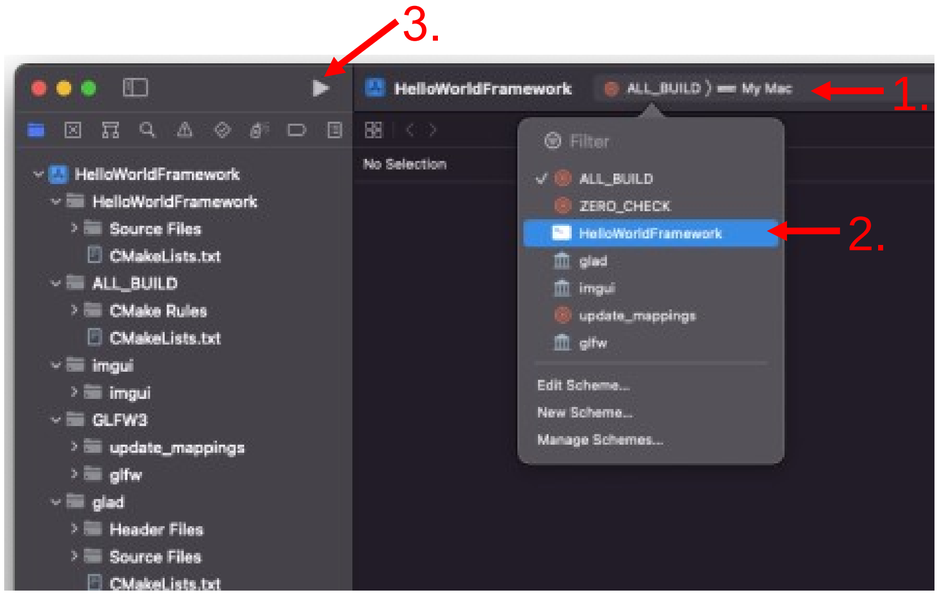


1. Right click

2. Left click

Instructions: Programming tasks

- **Note on compiling and running**
  - In Xcode:



By clicking play (3), the code is compiled and executed.