Advanced Computer Graphics

Tutorium 1

Bremen

Assignment Sheets

- Groups of 2-3 students

- Are to be submitted via Gitlab (https://gitlab.informatik.uni-bremen.de)!
  - Create a single repository for all submissions.
  - Invite both tutors (Roland Fischer & Navid Mirzayousef Jadid) so that we can check your submissions (select ‘Developer’ as role).
  - GitLab handles: s_8ix2ba & navid
  - Push your solution into the Git-Repo before the deadline is over.
    - ➢ `This counts as a submission.`

Assignment Sheets

- We use **C++**
  - But in a way which is quite similar to Java.
- **We expect**:
  - Clean and clear Code.
  - Some comments in code which describe what is done and why.
- We do **<u>not</u>** expect:
  - A description of your code in a separate document.
    - ➢ For programming tasks, it is sufficient to submit only your project files / source code.

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)
  - **Deadline** is 24.04.2024 at 11:59 pm (23:59 Uhr).

Instructions: Programming tasks

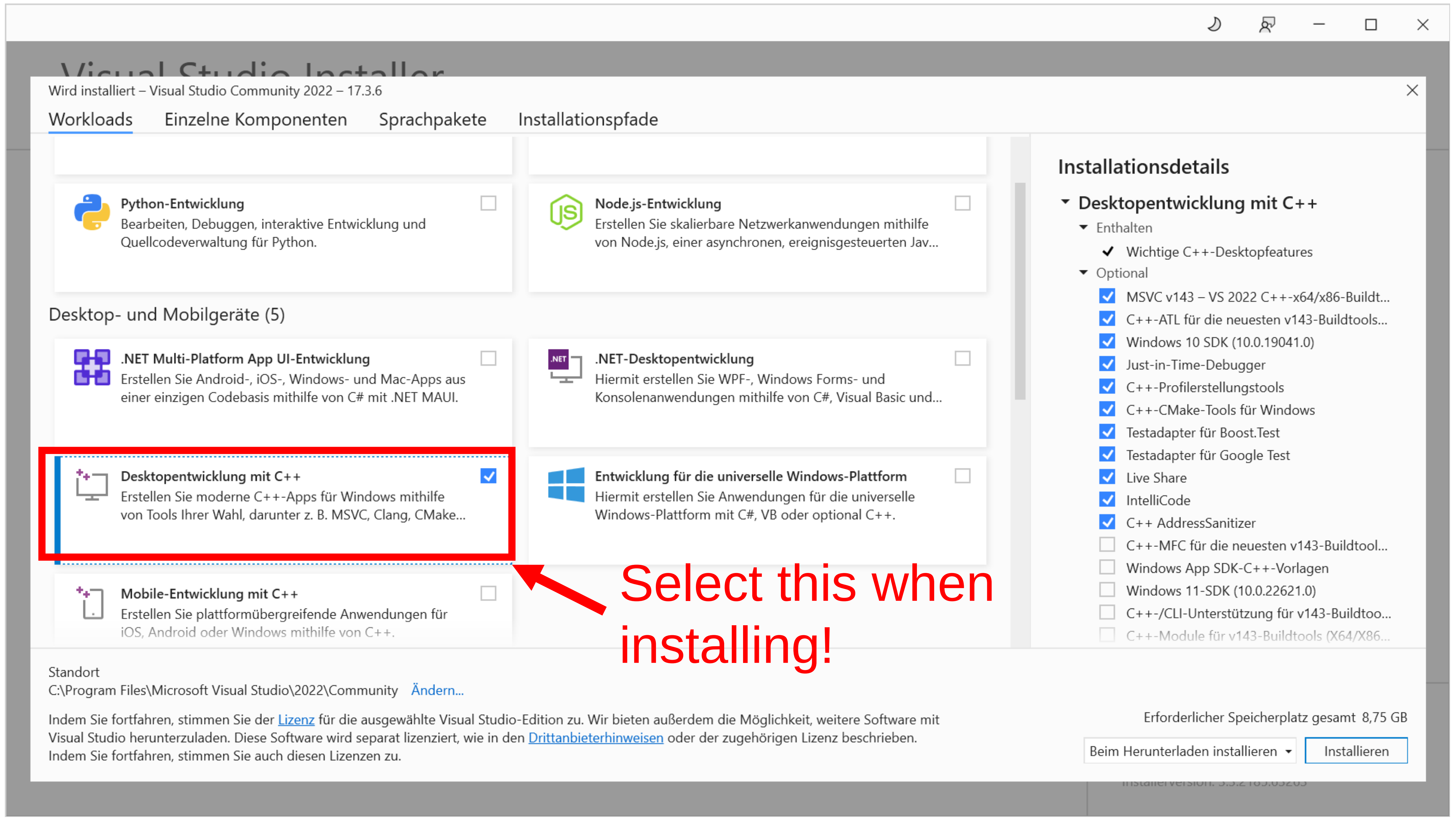
- **Prerequisites**:
  1. An **IDE** of your choice
  2. A **C++-compiler** of your choice
  3. **CMake**
  4. **OpenGL**

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/



Visual Studio Installer

Wird installiert – Visual Studio Community 2022 – 17.3.6

Workloads | Einzelne Komponenten | Sprachpakete | Installationspfade

Python-Entwicklung
Bearbeiten, Debuggen, interaktive Entwicklung und Quellcodeverwaltung für Python.

Node.js-Entwicklung
Erstellen Sie skalierbare Netzwerkanwendungen mithilfe von Node.js, einer asynchronen, ereignisgesteuerten Jav...

Desktop- und Mobilgeräte (5)

.NET Multi-Platform App UI-Entwicklung
Erstellen Sie Android-, iOS-, Windows- und Mac-Apps aus einer einzigen Codebasis mithilfe von C# mit .NET MAUI.

.NET-Desktopentwicklung
Hiermit erstellen Sie WPF-, Windows Forms- und Konsolenanwendungen mithilfe von C#, Visual Basic und...

[✓] Desktopentwicklung mit C++
Erstellen Sie moderne C++-Apps für Windows mithilfe von Tools Ihrer Wahl, darunter z. B. MSVC, Clang, CMake...

Entwicklung für die universelle Windows-Plattform
Hiermit erstellen Sie Anwendungen für die universelle Windows-Plattform mit C#, VB oder optional C++.

Mobile-Entwicklung mit C++
Erstellen Sie plattformübergreifende Anwendungen für iOS, Android oder Windows mithilfe von C++.

Select this when installing!

Installationsdetails

Desktopentwicklung mit C++
- Enthalten
  - ✓ Wichtige C++-Desktopfeatures
- Optional
  - [✓] MSVC v143 – VS 2022 C++-x64/x86-Buildt...
  - [✓] C++-ATL für die neuesten v143-Buildtools...
  - [✓] Windows 10 SDK (10.0.19041.0)
  - [✓] Just-in-Time-Debugger
  - [✓] C++-Profilerstellungstools
  - [✓] C++-CMake-Tools für Windows
  - [✓] Testadapter für Boost.Test
  - [✓] Testadapter für Google Test
  - [✓] Live Share
  - [✓] IntelliCode
  - [✓] C++ AddressSanitizer
  - [ ] C++-MFC für die neuesten v143-Buildtool...
  - [ ] Windows App SDK-C++-Vorlagen
  - [ ] Windows 11-SDK (10.0.22621.0)
  - [ ] C++-/CLI-Unterstützung für v143-Buildtoo...
  - [ ] C++-Module für v143-Buildtools (X64/X86...

Standort
C:\Program Files\Microsoft Visual Studio\2022\Community Ändern...

Indem Sie fortfahren, stimmen Sie der Lizenz für die ausgewählte Visual Studio-Edition zu. Wir bieten außerdem die Möglichkeit, weitere Software mit Visual Studio herunterzuladen. Diese Software wird separat lizenziert, wie in den Drittanbieterhinweisen oder der zugehörigen Lizenz beschrieben. Indem Sie fortfahren, stimmen Sie auch diesen Lizenzen zu.

Erforderlicher Speicherplatz gesamt 8,75 GB

Beim Herunterladen installieren | Installieren

Instructions: Programming tasks

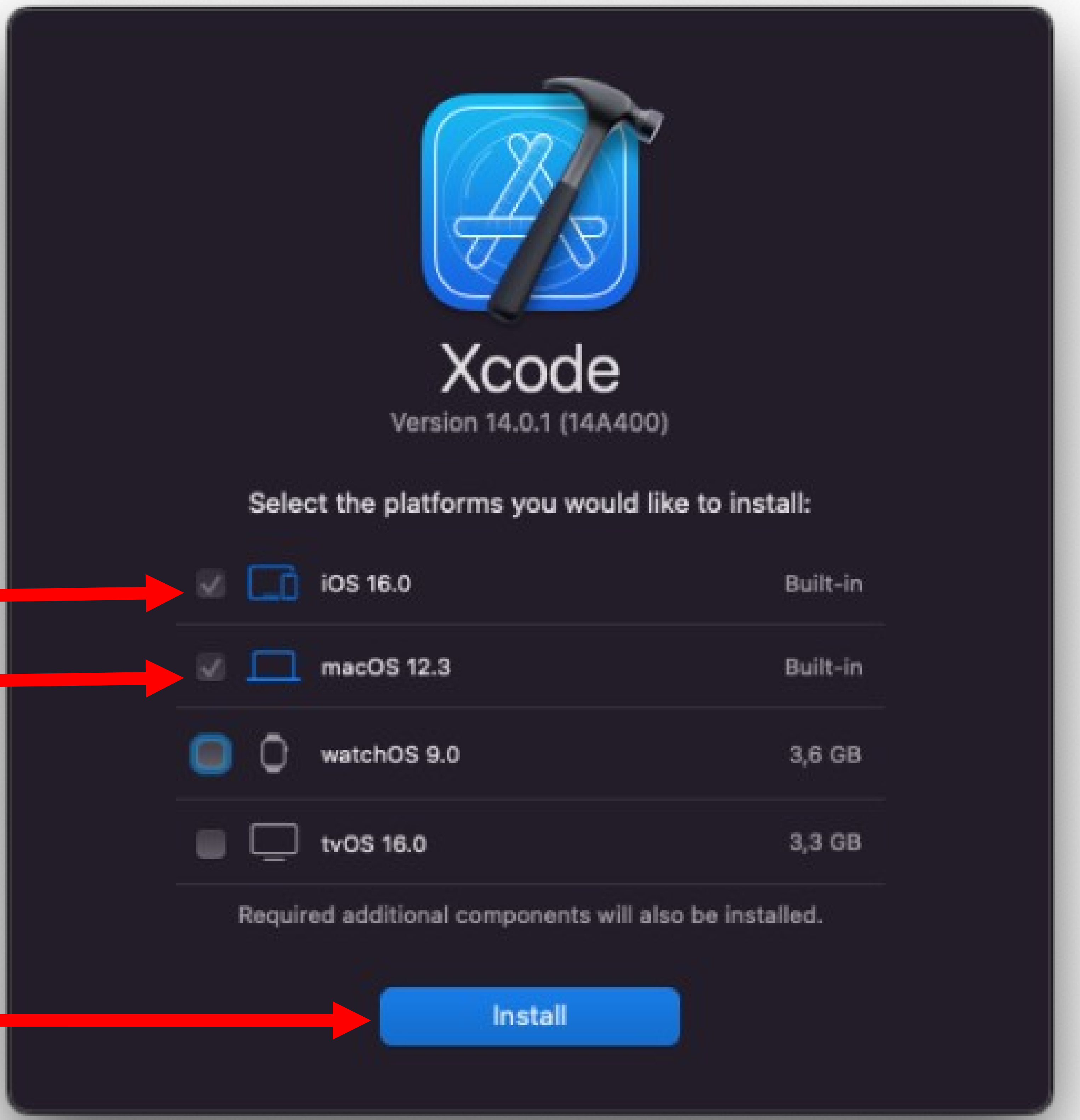
1. **Possible IDEs:**
   - Visual Studio *(Windows)*
     - https://visualstudio.microsoft.com/de/downloads/
   - Xcode *(Mac)*
     - https://developer.apple.com/xcode/

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:



Xcode

Version 14.0.1 (14A400)

Select the platforms you would like to install:

✓	iOS 16.0	Built-in
✓	macOS 12.3	Built-in
	watchOS 9.0	3,6 GB
	tvOS 16.0	3,3 GB

Required additional components will also be installed.

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/
   - Other (including Linux): QTCreator, CodeLite, Code::Blocks, Visual Studio Code 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 *(u.a. Linux):*
  - Installation by using a paket manager (Ubuntu): sudo apt-get install gcc

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 paket 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 paket 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 /
```

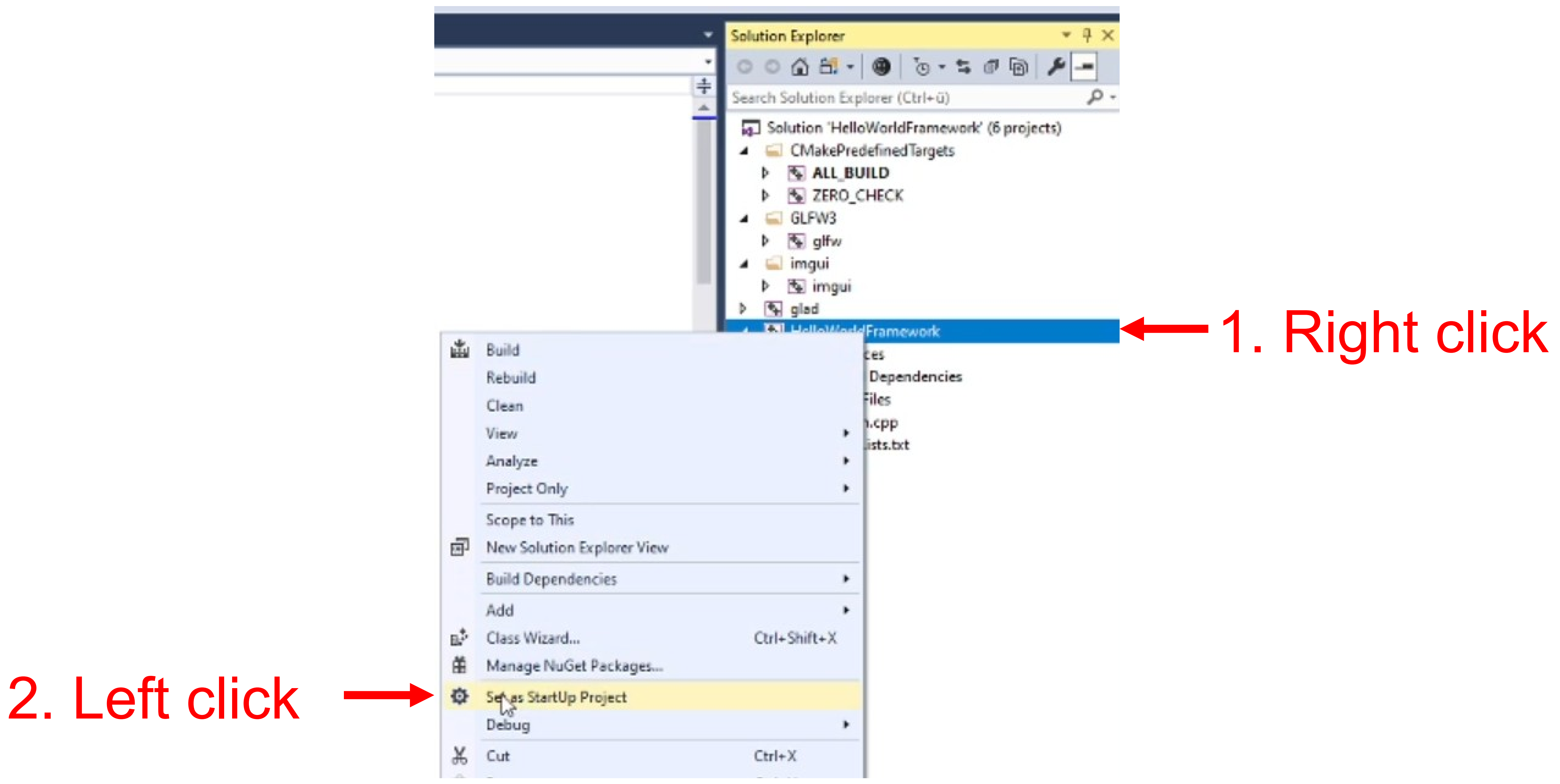
(https://mac.r-project.org/openmp/)

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

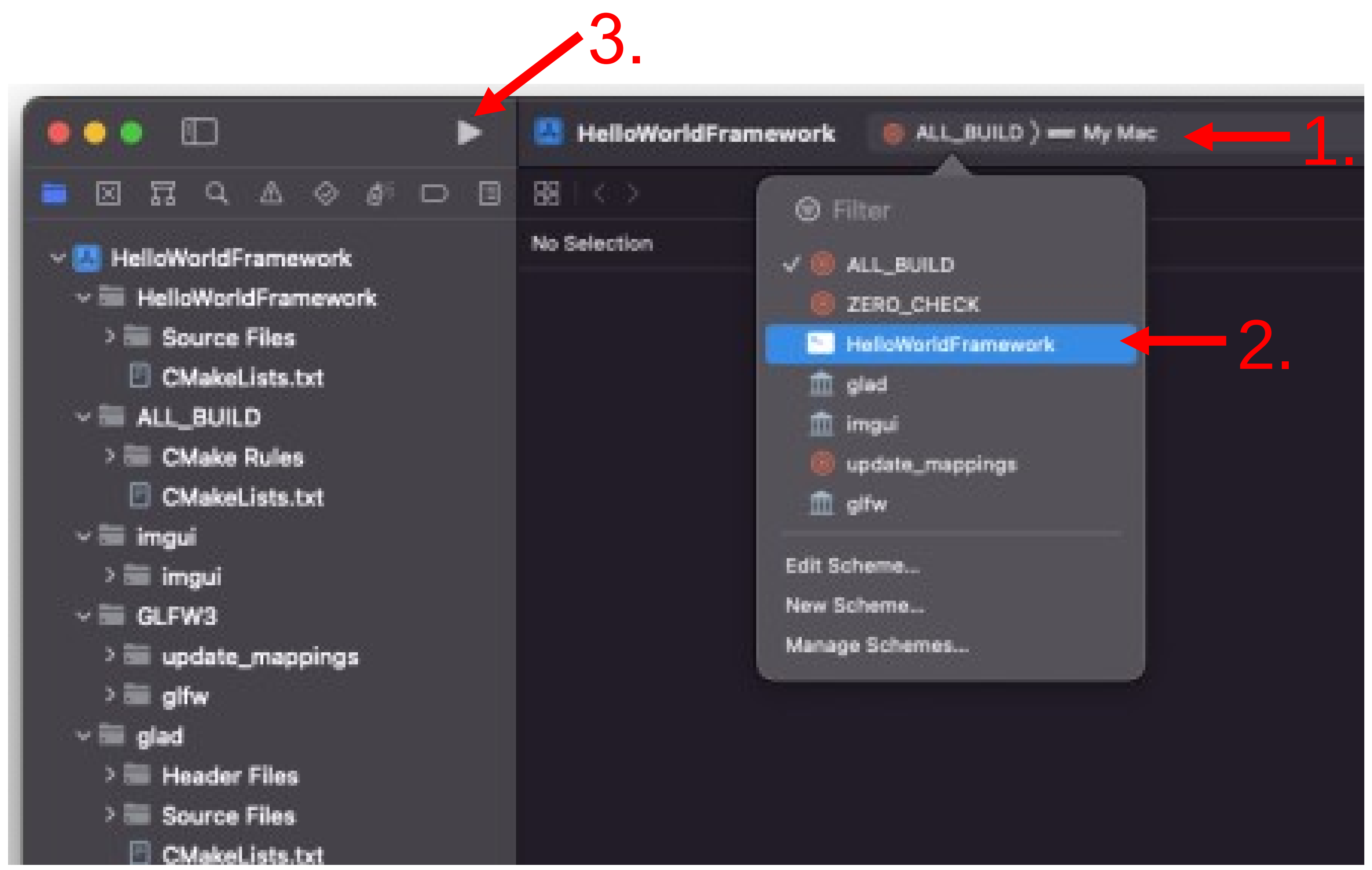
Instructions: Programming tasks

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



Instructions: Programming tasks

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



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