

Übung 1 PgrT: Code Compilieren mit und ohne Cmake

1. Ordnerstruktur

Üblicherweise verwendet man folgende Ordnerstruktur für C++ Projekte:

- 3rdPary / lib (statische Bibliotheken)
- bin /build (Binaries)
- cmake (Cmake Scripte)
- include (Header Dateien)
- src (Source Dateien)
 - main.cpp
 - CMakeLists.txt (Cmake Datei für die Sourcen)
- CMakeLists.txt (Cmake Datei für Projekteinstellungen)

Bauen sie die Ordnerstruktur in einem Projektorder ihrer Wahl auf und schreiben sie einen kleinen Code ihrer Wahl in die main.cpp. Verwenden sie ein C++11/14 Feature ihrer Wahl.

2. Compilieren „von Hand“

Compilieren Sie den Code über das Terminal ohne Cmake. Achten Sie darauf, dass die Binärdatei auch im build Verzeichnis landen.

3. CMake Dateien

Im Moment haben wir zwei CmakeLists.txt.

In die Datei im Hauptverzeichnis tragen sie folgende Elemente ein:

- Die benötigte CMake Version
- Den Projektnamen
- Eine Projektversion
- Den CMake Skript Pfad
- den Pfad für die Include Dateien
- den Pfad für die Source Dateien
- Den verwendeten C++14 Standard

Dafür benötigen Sie die Befehle: `cmake_minimum_required`, `project`, `set`, `include_directories` und `add_definition`. Des weiteren geben sie mit `message` Den Projektnamen und die Projektversion aus.

In die Datei im Source Verzeichnis tragen sie folgende Elemente ein:

- Die zu Compilierenden Source Dateien
- Die zu Linkenden Bibliotheken (im Moment noch leer)

Dafür benötigen Sie die Befehle: `add_executable` und `target_link_libraries`

Wenn Sie alles richtig Gemacht haben. Können Sie jetzt mit dem Terminal in das build Verzeichnis wechseln und mit dem Befehl „`cmake ..`“ die Make Dateien erstellen. Anschließen mit „`make all`“ ihre Datei Kompilieren.

4. Integration in eine IDE

Falls Sie für die weiteren Übungen eine Entwicklungsumgebung verwenden wollen, integrieren Sie das Projekt. Achten Sie darauf dass

1. Cmake als Buildtool verwendet wird
2. Die Source und Header Pfade korrekt sind
3. Der Build-Pfad korrekt ist

5. Vorbereitung für die nächsten Übungen

In den folgenden Übungen werden wir die Simple and Fast Media Library nutzen um grafische Anwendungen zu Programmieren. Installieren sie die mithilfe ihres Paketarmangers. Desweiteren kopieren sie diese Datei (

<https://github.com/SFML/SFML/blob/master/cmake/Modules/FindSFML.cmake>) in ihr cmake Verzeichnis.

Nun geben Sie in der CmakeLists.txt im Hauptverzeichnis „`find_package(SFML REQUIRED)`“ an und erstellen Sie die Make Dateien neu.