1 CMake Tutorial

Willkommen in der Computergrafik-Übung!

Dieses Dokument soll Ihnen bei der Erzeugung eines Visual Studio Projekts für die Übungsaufgaben helfen.

Da dies mit Hilfe des Programmes CMake¹ geschieht, wird im folgenden als Voraussetzung angesehen, dass Sie CMake sowie Visual Studio bereits installiert haben. Dieses Tutorial behandelt explizit CMake unter Windows. CMake selbst kann jedoch auch für andere Entwicklungsumgebungen wie z.B. Codeblocks oder Eclipse sowie unter anderen Betriebsystemen die Projektdateien erzeugen. Die meisten Schritte erfolgen dabei analog zu dieser Anleitung.

Hinweis: Von Visual Studio benötigen Sie die Version "Visual Studio 12 2013" oder neuer, da der Compiler der vorherigen Versionen noch kein $C++11^2$ unterstützt.

1.1 Grundlegendes

CMake dient dazu ein Softwareprojekt möglichst unabhängig von einer bestimmten Entwicklungsumgebung entwickeln zu können. Daher erzeugt CMake aus den mitgelieferten CMake-Dateien die Projektdateien für Ihre Entwicklungsumgebung.

Dabei verfolgt CMake den Ansatz den Quellcode und die daraus erzeugten Binärdateien getrennt zu halten. Dies resultiert in zwei getrennten Verzeichnissen. In einem liegt der Quellcode, in diesem Fall die Übungsaufgabe sowie das Framework. In einem seperaten Verzeichnis wird das Projekt der Entwicklungsumgebung erzeugt, sowie die Binärdateien erstellt.

1.2 Visual Studio Projekt erzeugen

Um mit der Bearbeitung einer Übungsaufgabe zu beginnen, entpacken Sie bitte zuerst das ZIP-Archiv, welches die nötigen Dateien der Aufgabe enthält. Achten Sie hier bitte darauf, dass Ihr Arbeitsverzeichnis **keine** Umlaute enthält.

Öffnen Sie nun CMake (cmake-gui) und wählen Sie als *source code* den Ordner in Ihrem Arbeitsverzeichnis mit der enthaltenen Aufgabe. In der ersten Übungsaufgabe heißt dieser "01_colors". Für den Ordner der *binaries* wählen Sie den in der Ordnerstruktur der Aufgabe bereits vorhandenen Ordner "build" aus. In diesem Ordner wird das Visual Studio Projekt erzeugt werden, als auch die Binaries erstellt.

Die folgende Grafik verdeutlicht wie dies in der CMake-GUI nun aussehen sollte.

 $^{^1\}mathrm{CMake}$ ist ein Programm, dass die Build-Umgebung für Ihre Plattform generiert. https://cmake.org/ $^2\mathrm{C}++11$ ist der "neue" C++ Standard

CMake 3.2.1 - Z:/cg_exercise_01/01_colors/build			
Hie loois Options Help			
Where is the source code: Z:/cg_exercise_01/01_colors			Browse Source
Where to build the binaries: Z:/cg_exercise_01/01_colors/build		•	Browse Build
Search:	Grouped 🗖 Advanced	🕂 Add Entry	💥 Remove Entry
Name	Value		
Press Configure to update and display new values in	red, then press Generate to gen	erate selected build	files.
Configure Generate Current Generator: None			

Nachdem Sie die Ordner angegeben haben klicken Sie auf den "Configure" Button. Es sollte ein Fenster erscheinen in welchem Sie den Generator für das Projekt auswählen können.

CMake 3.2.1 - Z:/cg_exercise_01/01_colors/build						
Where is the source	code:	Z:/cg_exercise_01/01_colors				Browse Source
Where to build the binaries: Z:/cg_exercise_01/01_colors/build					•	Browse Build
Search:			Grouped	Advanced	🕂 Add Entry	💥 Remove Entry
Name 🔼	cmake-	gui			<u>?</u> ×	
Sp	pecify the	generator for this project				
	Visual Stud	lio 12 2013				
G	🖲 Use de	fault native compilers				
c	C Specify	native compilers				
c	C Specify	toolchain file for cross-compiling				
F C	C Specify	options for cross-compiling				files.
Configure						
			< Back	Finish	Cancel	

Wählen Sie hier **Ihre installierte Visual Studio Version** aus. Danach klicken Sie auf den "Finish" Button. CMake wird nun die nötigen Konfigurationen innerhalb des "build" Ordner vornehmen um ein Visual Studio Projekt erzeugen zu können.

Wenn in der Ausgabe im unteren Teil des Fensters der Text "Configuring done" erscheint, war die Konfiguration erfolgreich. Klicken Sie anschließend auf den "Generate" Button um das Visual Studio Projekt zu erzeugen.

A CMake 3.2.1 - Z:/cg_exercise_01/01_colors/build				
File Tools Options Help				
Where is the source code: Z:/cg_exercise_01/01_colors	Browse Source			
Where to build the binaries: Z:/cg_exercise_01/01_colors/build	Browse Build			
Search:	Grouped F Advanced 🔂 Add Entry			
Name	Value			
ANTTWEAKBAR_LIB ANTTWEAKBAR_LIB_DLL BUILD_SHARED_LIBS CMAKE_BUILD_TYPE CMAKE_CONFIGURATION_TYPES CMAKE_INSTALL_PREFIX GLEW_BUILD_INFO GLEW_LIBRARY GLEW_LIBRARY_SHARED GLEW_BUILD_DCS	Z:/cg_exercise_01/cglib/lib/AntTweakBar/lib/AntTweakBar.lib Z:/cg_exercise_01/cglib/lib/AntTweakBar/lib/AntTweakBar.dll Debug Debug;Release;MinSizeRel;RelWithDebInfo C:/Program Files (x86)/cg GLEW_INCLUDE_PATH-NOTFOUND GLEW_LIBRARY-NOTFOUND GLEW_LIBRARY_SHARED-NOTFOUND			
Press Configure to update and display new values in red, then press Generate to generate selected build files.				
Check for working CXX compiler using: Visual Studio 12 2013 Check for working CXX compiler using: Visual Studio 12 2013 works Detecting CXX compiler ABI info				
Detecting CXX compiler ABI info - done Detecting CXX compile features Detecting CXX compile features - done				
<pre>Found OpenGL: opengl32 could not find system wide GFLW install, using our own version of GLFW instead Looking for include file pthread.h Looking for include file pthread.h - not found Found Threads: TRUE Using Win32 for window creation Using WGL for context creation GLFW_LIBRARIES: opengl32;glfw Z:/cg_exercise_Ol/cglib/build/glew Configuring done Generating done</pre>				
Generating done				

CMake sollte nun in dem "build" Ordner eine Visual Studio Projektdatei erzeugt haben. Sie können an dieser Stelle CMake beenden. Navigieren Sie mit Ihrem Dateibrowser in den "build" Ordner und öffnen Sie die "cg.sln" Datei. Daraufhin sollte sich Visual Studio öffnen und das Projekt laden. Hier können Sie nun mit der Bearbeitung der Aufgaben beginnen.

Hinweis: Diese Prozedur müssen Sie nur einmal pro Übungsblatt durchführen. Danach können Sie immer die entsprechende Visual Studio Projektdatei öffnen um die Bearbeitung fortzusetzen.