Um VisualStudioCode und den GCC Compiler zu installieren befolgen Sie folgende Schritte:

1. [Projekt von TortoiseSVN herunterladen](#Schritt1)
2. [TDM\_GCC Compiler installieren](#Schritt2)
3. [CMake installieren](#Schritt3)
4. [Installation überprüfen](#Schritt4)
5. [VisualStudioCode installieren](#Schritt5)
6. [Projekt ausführen](#Schritt6)
7. [Eigenes Projekt erstellen](#Schritt8)
8. [Funktionsweise von CMake und GCC Compiler](#Schritt8)

1. Projekt von TortoiseSVN herunterladen

Voraussetzung: TortoiseSVN ist installiert. Rechtsklick auf dem Desktop und **SVN Checkout…** auswählen. Im Feld **URL of repository** folgenden Link eingeben: „svn://gs-tuebingen.dyndns.org/verteilSoftware“. Mit OK bestätigen.

2. TDM\_GCC Compiler installieren

Auf dem Desktop wird ein neuer Ordner mit dem Namen **verteilSoftware** angelegt. In diesem Ordner die Datei **GCC\_CMake\_Testprojekt\_und\_Doku.zip** extrahieren und auf dem Desktop speichern.

Im extrahierten Ordner **Install\_GCC\_CMake\_Testprojekt\_und\_Doku** befindet sichim Unterverzeichnis **Install\_GCC\_CMake\_und\_Doku** der Ordner **1\_TDM\_GCC**.  
Darin **tdm64-gcc-9.2.0.exe** als Administrator ausführen (Rechtsklick auf .exe und **Als Administrator ausführen** auswählen).  
Im ersten Schritt der Installation **Create** und anschließend **MinGW-w64/TDM64 (32-bit and 64-bit)** auswählen. Mit Next bestätigen und folgenden Pfad bei Installation Directory angeben: **C:\TDM-GCC-64**Wieder mitNext bestätigen und die Installation durchführen.

3. CMake installieren

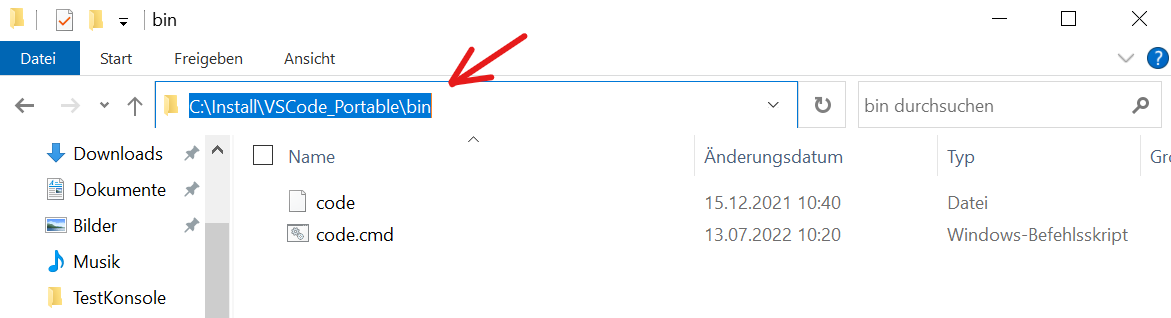
Im Ordner **Install\_GCC\_CMake\_Testprojekt\_und\_Doku** befindet sichim Unterverzeichnis **Install\_GCC\_CMake\_und\_Doku** der Ordner **2\_CMake**. Darin die Datei **cmake-3.19.4-win64-x64.msi** ausführen (keine Administratorrechte notwendig). Abschließend unbedingt in der Installation **Install CMake for all users** auswählen.

4. Installation überprüfen

Um die Installation zu überprüfen im Ordner **Install\_GCC\_CMake\_Testprojekt\_und\_Doku**  das Unterverzeichnis **Install\_GCC\_CMake\_und\_Doku** öffnen.   
Jetzt im Explorer an Stelle des Pfadnamens den Befehl **cmd** eingeben und mit Enter bestätigen.  
Im sich öffnenden Konsolenfenster folgenden Befehl eingeben: **path > path.txt** eingeben und mit Enter bestätigen. Dadurch wird eine neue Datei mit dem Namen **path.txt** angelegt. Im Explorer anschließend diese Datei öffnen. Wenn darin sowohl der Pfad **C:\TDM-GCC-64\bin;** als auch **C:\Install\VSCode\_Portable\bin;** vorhanden sind, hat die Installation funktioniert.

Um ganz sicher zu gehen, dass alles funktioniert hat, im Ordner **Install\_GCC\_CMake\_Testprojekt\_und\_Doku** in das Unterverzeichnis **TestKonsole** gehenund dortdie Datei **\_compileAndRun.cmd** mit einem Doppelklick ausführen. Der Aufforderung eine Taste zu drücken bitte folge leisten. Dadurch wurde die Datei **Welt.exe** erstellt. Diese mit einem Doppelklick ausführen. Die Anzeige „Hallo, du schoene Welt!“ ist der (fast) endgültige Beweis, dass die Installation geklappt hat ☺.

5. VisualStudioCode installieren

Um VisualStudioCode zu installieren im Ordner **VerteilSoftware** den Unterordner **VisualStudioCode\_Portable** öffnen**.** Darin die Datei **VSCode\_Portable.zip** extrahieren und auf dem Desktop speichern.   
Anschließend einen neuen Ordner **Install** in **c:\** erstellen.  
Den extrahierten Ordner **VSCode\_Portable** in den neu erstellten Ordner **Install** verschieben. Im Ordner **VSCode\_Portable** in den Unterordner **bin** gehen. Hier im Explorer den Pfadnamen von **bin** kopieren.

Die Windows-Taste drücken und nach **Umgebungsvariablen bearbeiten** suchen**.** Das Programm öffnen und im sich öffnenden Fenster unten auf **Umgebungsvariablen…** klicken**.** Im unteren Bereich des neu geöffneten Fensters unter **Systemvariablen** die Variable **Path** suchen und mit einem Doppelklick bearbeiten. Anschließend auf **Neu** klicken und bereits aus dem **bin**-Ordner kopierten Pfad (C:\Install\VSCode\_Portable\bin)eingeben. Mit OK bestätigen.

Um zu kontrollieren ob der Pfad richtig eingegeben wurde **Windows + R** drückenund **cmd** in dem neu geöffneten Fenster eingeben.Anschließend in der Konsole **code** eingeben und mit Enter bestätigen. Wenn sich Visual Studio Code öffnet, ist die Installation erfolgreich abgeschlossen ☺.

6. Projekt ausführen

Für jede folgende Methode muss das Projekt wie in Kapitel 7 beschrieben aufgebaut sein.

Das Projekt kann:

1. Manuell durch folgende Kommandos in der Konsole ausgeführt werden:
   1. Cmake . –G “MinGW Makefiles“
   2. Mingw32-make
2. Durch eine **Compile.cmd** Datei mit in Option 1 genannten Befehlen ausgeführt werden
3. In VisualStudioCode durch Drücken des **kleinen Pfeiles ᐅ** in der Statusleiste (blaue Leiste im unteren Bereich von VisualStudioCode).

!!! Als Compiler muss der **GCC Compiler** verwendet werden !!!

7. Eigenes Projekt erstellen

Um ein eigenes Projekt zu erstellen muss im Projektordner Folgendes enthalten sein:

1. **CMakeLists.txt** Datei mit Inhalt:

cmake\_minimum\_required(VERSION 3.16)

project(Welt)

file(GLOB\_RECURSE USER\_SOURCE\_FILES

src/\*.cpp

)

add\_executable(Welt

${USER\_SOURCE\_FILES}

)

target\_include\_directories(Welt PRIVATE

src

)

1. **Compile.cmd** Datei (Kommandos können auch manuell in Konsole eingegeben werden)

Cmake . –G “MinGW Makefiles“

Mingw32-make

1. Mingw32-make **src** Ordner, der mindestens Folgendes enthalten muss:
   1. **main.h** Datei
   2. **Welt.cpp** Datei
   3. **Welt.h** Datei

8. Erklärung:

Die c++ Datein sind im Ordner src, damit sie aufgeräumt sind und der Überblick gegeben ist.

1. **CMakeLists.txt** Datei:

cmake\_minimum\_required(VERSION 3.16) #Mindestens version3.16 von cmake für das Projekt Welt

project(Welt)

#Fügt alle c++ Dateien zusammen aus Ordner src zum Kompilieren. Hier können weitere Pfade angegeben werden.

file(GLOB\_RECURSE USER\_SOURCE\_FILES

src/\*.cpp

)

# Erstellt .exe datei aus USER\_SOURCE\_FILES. Welt ist der Name, der exe Datei. Kann umbenannt werden.

add\_executable(Welt

${USER\_SOURCE\_FILES}

)

# Alle Pfade von Bibliotheken werden in CMakeCache.txt zusammengefasst.

target\_include\_directories(Welt PRIVATE

src

)

1. **Compile.cmd** Datei:

# Generiert Makefiles zur Verwendung mit mingw32-make  
# MinGw(Minimalistische Grafikanwendung für windows)

Cmake . –G “MinGW Makefiles“

# Ausführung der von cmake erstellten Dateien.

Mingw32-make