**Installatie document ROC ter Aa**

**Project: WireWorld**

**Client: ROC Ter Aa**

**Project Number: 000005**

**Auteur: Giel Willemsen, Guylian Gilsing**

**Date: 13-11-2019**

**Version: 0.1**

# Installeren

|  |  |
| --- | --- |
| **Wat te installeeren** | **Notes** |
| Cmake | Cross platform build description builder |
| Git | Version control |
| Visual Studio 2019 | IDE |
| GLFW | OpenGL frame work |
| GLAD | opengl function loader |
| Dear IMGUI | C++ bloat free imgui library |
| GLM | OpenGl math library |
| VC++ | C++ from MS Visual Studio |

Als dit alles is geïnstalleerd, start dan Visual Studio 2019 op en maak een nieuw leeg CMake Project aan.

# Project mappen structuur

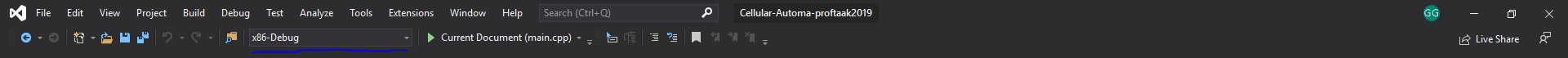
Dit is de standaard mappen structuur van het (nu nog) lege Cmake project.

* dependencies
  + GLAD
  + IMGUI
* src
  + main.cpp
* CMakeLists.txt

In het CMakeLists.txt bestand mag de de code gekopieerd worden die onderaan in dit document onder het kopje bijlages staat.

# Installatie instructies

|  |  |  |
| --- | --- | --- |
| **Waarbij hoort het** | **Versie nummer** | **Wat te doen** |
| Cmake | 3.16.0-rc3 | Express installatie |
| Git | version 2.24.0 | Express installatie |
| GLFW | glfw-3.3.bin.WIN32 | Download de prebuild windows library. Daarna maak een omgevingsvariabele aan die “GLFW\_INCLUDE\_DIR” heet, deze variabele wijst naar het mapje “include”.  Daarna maak een omgevingsvariabele aan die “GLFW\_LIBRARY” heet, deze variabele wijst naar het mapje “vc-2019”.  Voeg ook de waarde van “GLFW\_LIBRARY” toe aan je “Path variabele” |
| GLAD | - | Ga naar: https://glad.dav1d.de/ en installeer de core version van de GL 4.6 API. Zet daarna de inhoud in het mapje “dependencies/GLAD”. |
| GLM | glm-0.9.9.6 | Maak een omgevingsvariabele aan die “GLM\_INCLUDE\_DIR” heet, deze variabele wijst naar het mapje “glm” in de main GLM directory. |
| Dear IMGUI | v1.74 | Ga naar [https://github.com/ocornut/imgui/releases en download versie 1.47](https://github.com/ocornut/imgui/releases%20en%20download%20versie%201.47), zet daarna de .h en .cpp files, die je in de main directory vindt, in "dependencies/IMGUI" |
| Visual Studio 2019 | - | Als de installatie begonnen is check bij workloads de checkboxes bij “Desktop Development with C++”. |

Als je de applicatie wilt gaan compilen moet je in Visual Studio 2019 X86-Debug aan hebben staan als je de 32 bits variant van GLFW gebruikt. Anders kun je gewoon X64-Debug gebruiken.

# Bijlagen

## CMakeLists.txt basis layout.

# CMakeList.txt : CMake project for Cellular-automata-proftaak2019, include source and define  
# project specific logic here.  
#cmake\_minimum\_required (VERSION 3.8)

# Add source to this project's executable.  
#add\_executable (Cellular-automata-proftaak2019 "Cellular-automata-proftaak2019.cpp" "Cellular-automata-proftaak2019.h")

# TODO: Add tests and install targets if needed.  
cmake\_minimum\_required (VERSION 3.8)

project(Cellular-automata)

# link libraries directories  
include\_directories($ENV{GLFW\_INCLUDE\_DIR})  
link\_directories($ENV{GLFW\_LIBRARY})

include\_directories($ENV{GLM\_INCLUDE\_DIR})  
include\_directories(dependencies/glad/include)

#add\_subdirectory(dependencies/glad/src)  
include\_directories(dependencies/imgui)  
add\_subdirectory(dependencies/imgui)  
include\_directories(src/imgui)  
include\_directories(src)

# we need openGL  
#this is some CMake magic right here  
find\_package(OpenGL REQUIRED)

set (CPPFILES

"src/shader.cpp"

"src/main.cpp"

"src/imgui/imgui\_impl\_glfw.cpp"

"src/imgui/imgui\_impl\_opengl3.cpp"

)  
add\_executable(App ${CPPFILES} dependencies/GLAD/src/glad.c)

# using the dlls we need glew32.dll, glew32.lib (GLEW)  
# and glfw32.dll, glfw32dll.lib (GLFW)  
# and require the OpenGL32 lib  
# they are on path  
target\_link\_libraries(App OpenGL32 glfw3 imgui)