P2 CG

Gui Costa nº 61172

Objetivos:

Mostrar um símbolo colorido modelado a partir de triângulos usando qualquer ferramenta de modelação.

# Plataforma de desenvolvimento:

SO Windows com MINGW64, editor de texto é o visual code com extensão CMAKE\_TOOLS para a execução dos ficheiros CMakelist.txt. A linguagem usada é c++. As livrarias usadas são GLAD e GLFW 3.3.3. Github para a gestão de versões. A ferramenta usada para modelar foi geogebra.

# Resultados:

Build:

Devido ao uso do vscode e das extensões relativas ao cmake o processo é bastante simples, para cada pasta nova (neste caso o P2) abrimos a palete de comandos do vscode visíveis na imagem a baixo e corremos primeiro CMake: configure que executa o seguinte comando

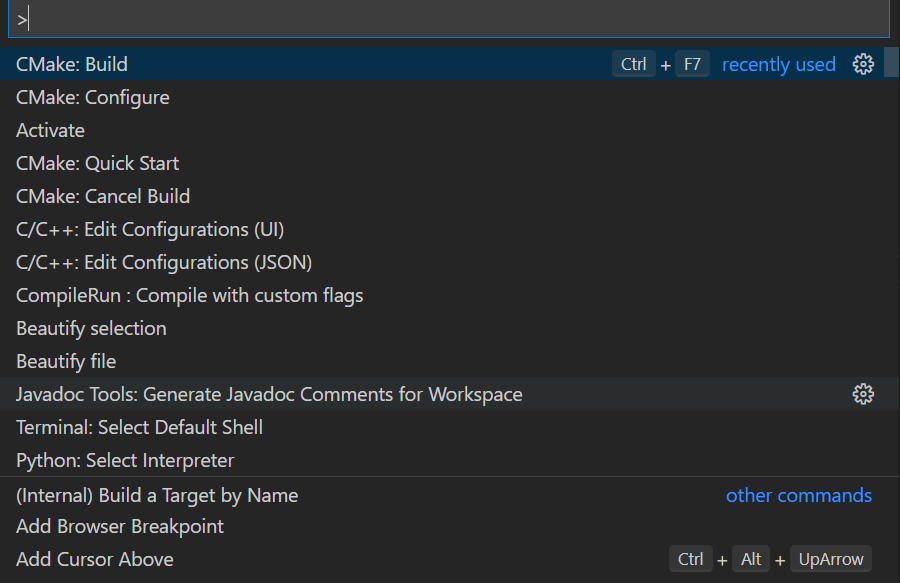
"C:\Program Files\CMake\bin\cmake.EXE" --no-warn-unused-cli -DCMAKE\_EXPORT\_COMPILE\_COMMANDS:BOOL=TRUE -DCMAKE\_BUILD\_TYPE:STRING=Debug "-DCMAKE\_C\_COMPILER:FILEPATH=C:\Program Files\mingw-w64\x86\_64-8.1.0-posix-seh-rt\_v6-rev0\mingw64\bin\gcc.exe" "-DCMAKE\_CXX\_COMPILER:FILEPATH=C:\Program Files\mingw-w64\x86\_64-8.1.0-posix-seh-rt\_v6-rev0\mingw64\bin\g++.exe" -Hc:/Users/UserName/Documents/GitHub/CG -Bc:/Users/Gui/Documents/GitHub/CG/build -G "MinGW Makefiles"

Figura 1 - Palete de comandos

Depois de realizar o projeto e querer testar corre-se o segundo comando da palete que é o CMake: Build que vai compilar o projeto e as dependências, este executa a seguinte linha

"C:\Program Files\CMake\bin\cmake.EXE" --build c:/Users/UserName/Documents/GitHub/CG/build --config Debug --target P2 -- -j 18

Run:

Finalmente para executar escreve-se "C:\Users\UserName\Documents\GitHub\CG\build\P2\P2.exe"

O programa executa com sucesso e gera uma janela de fundo cinzento-escuro que mostra a letra gamma maiúscula:

Imagens:

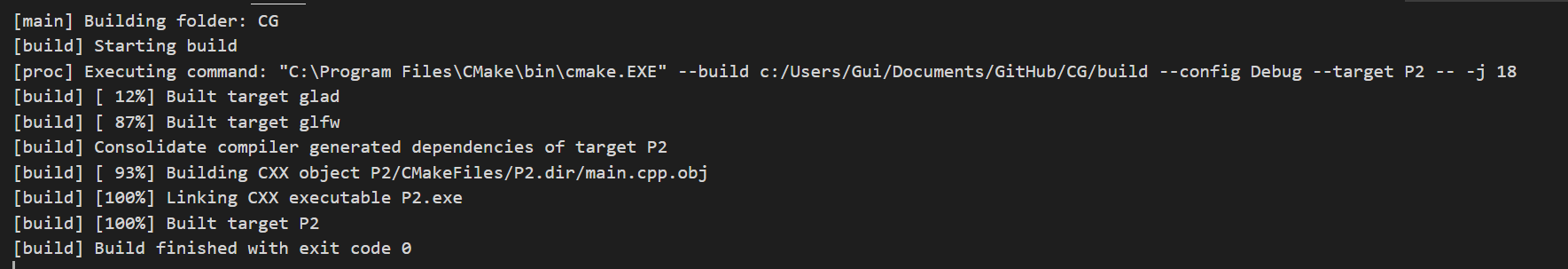


Figura 2 - resultado do Cmake build no vs code



Figura 3 - Resultado final