

DADOS DO RELATÓRIO

Matéria: Computação Gráfica Cod: CMP1170

Relatório Atividade 1

Aluno: Vitor de Almeida Silva

Matrícula: 20161003305497

Requisitos: Windows 7 ou 10, suporte para OpenGL 4.0 ou superior

ENUNCIADO

Elabore um relatório explicando pontualmente e resumidamente a funcionalidade de cada uma das bibliotecas abaixo:

1 - GLEW

2 - GLFW

3 - GLM

4 - SOIL2

Instale e configure as respectivas bibliotecas no seu ambiente de programação e acrescente no relatório o passo a passo realizado neste processo.

Teste o programa abaixo, descreva no relatório o comando realizado e também o passo a passo do respectivo código:

```
#include <GL/glew.h>
#include <GLFW/glfw3.h>
#include <iostream>
using namespace std;

void init(GLFWwindow* window){

void display(GLFWwindow* window, double currentTime){
    glClearColor(1.0, 0.0, 0.0, 1.0);
    glClear(GL_COLOR_BUFFER_BIT);
}

int main() {
    if(!glfwInit()) { exit(EXIT_FAILURE); }
    glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 4);
    glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 3);
    GLFWwindow* window = glfwCreateWindow(600, 600, "Programa Exemplo", NULL, NULL);
    glfwMakeContextCurrent(window);
    if(glewInit() != GLEW_OK) { exit(EXIT_FAILURE); }
    glfwSwapInterval(1);

    init(window);

    while(!glfwWindowShouldClose(window)) {
        display(window, glfwGetTime());
        glfwSwapBuffers(window);
        glfwPollEvents();
    }

    glfwDestroyWindow(window);
    glfwTerminate();
    exit(EXIT_SUCCESS);
}
```

1 DESENVOLVIMENTO

O desenvolvimento do relatório está dividido em seções que se referem a cada etapa necessária. Resumidamente, é preciso se iniciar com o download das bibliotecas (podem ser obtidas pelo site oficial do OpenGL e GLW.org). Posteriormente é preciso instalar os componentes necessários e as bibliotecas e, por fim, compilar o código.

É recomendável separar os arquivos em uma pasta específica de nome qualquer, “**openGLLibs**” por exemplo, somente a nível de organização e fácil acesso.

1.2 Instalando componentes necessários

Para que as bibliotecas sejam usáveis no programa, é preciso ligar as mesmas. Isso é feito através do software Cmake. Também, faz-se necessário, ter instalado e referenciado no caminho path os compiladores para os códigos, no presente caso será utilizado o gcc e o g++. Os passos para esses processos são mostrados a seguir.

1.2.1 Dlls necessárias

Verifique se os arquivos **opengl32.dll** e **glu32.dll** estão presentes na pasta System32 do Windows. (atualize o drive de vídeo caso não conste estes arquivos).

1.2.2 Baixar o MinGw

Este arquivo contém os compiladores e pode ser baixado do link a seguir:

<https://sourceforge.net/projects/mingww64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/8.1.0/threads-posix/sjlj/x86_64-8.1.0-release-posix-sjlj-rt_v6-rev0.7z/download> .

Após o download, os caminhos das pastas “**C:\mingw64\bin**” e “**C:\mingw64\x86_64-w64-mingw32\bin**”, devem ser copiados para o path de variáveis ambientes do sistema.

A ação pode ser validada através do comando “**g++ --version**” no terminal, se a versão do g++ for retornada ele está funcionando normalmente.

1.2.3 instalando o Cmake

Cmake é uma ferramenta criada para montar, tesar e ligar bibliotecas de softwares. Aqui, será utilizado para adicionar as bibliotecas que forem necessárias para o projeto. Cmake pode ser baixado pelo seguinte link:

< <https://cmake.org/download/>>.

Terminado o download, basta clicar sobre o executável, assinalar a opção “adicionar caminho as variáveis de ambiente”, caso contrário, é necessário adicionar o caminho “**C:\Program Files\CMake\bin**” ao path do sistema, e clicar em *next* (próximo) até terminar a ação.

Ao fim pode-se testar o processo através do comando “**cmake --version**” no terminal do sistema. Pode-se ainda, usar o comando “**Cmake --Help**” pra saber o uso do mesmo.

1.2.4 Instalar FreeGlut

Resumidamente FreeGlut é um conjunto de bibliotecas *open-source* alternativa ao GLUT. Este conjunto de bibliotecas, permite ao desenvolvedor criar e gerenciar janelas de conteúdos de contextos do OpenGL em várias plataformas. Este, por sua vez, também permite a leitura do mouse, teclado e funções de controles. FreeGlut pode ser baixado do link a seguir:

< <http://freeglut.sourceforge.net/>>.

O FreeGlut deve ser instalado utilizando o CMake. Para isso basta ir para a pasta do mesmo e buscar pelo arquivo “**CMakeList.txt**”, abrir o terminal neste caminho e executar os comandos a seguir:

- 1) **cmake -G "MinGW Makefiles" -S . -B . -DCMAKE_INSTALL_PREFIX=C:\mingw64\x86_64-w64-mingw32**
- 2) **mingw32-make all**
- 3) **mingw32-make install**

1.3 Instalando as bibliotecas

Existe um método padrão para que as bibliotecas sejam instaladas. Todas são feitas a partir do diretório onde se encontra o arquivo “**CMakeList.txt**”. Elas também são instaladas pelos mesmos comandos utilizados para instalar a FreeGlut.

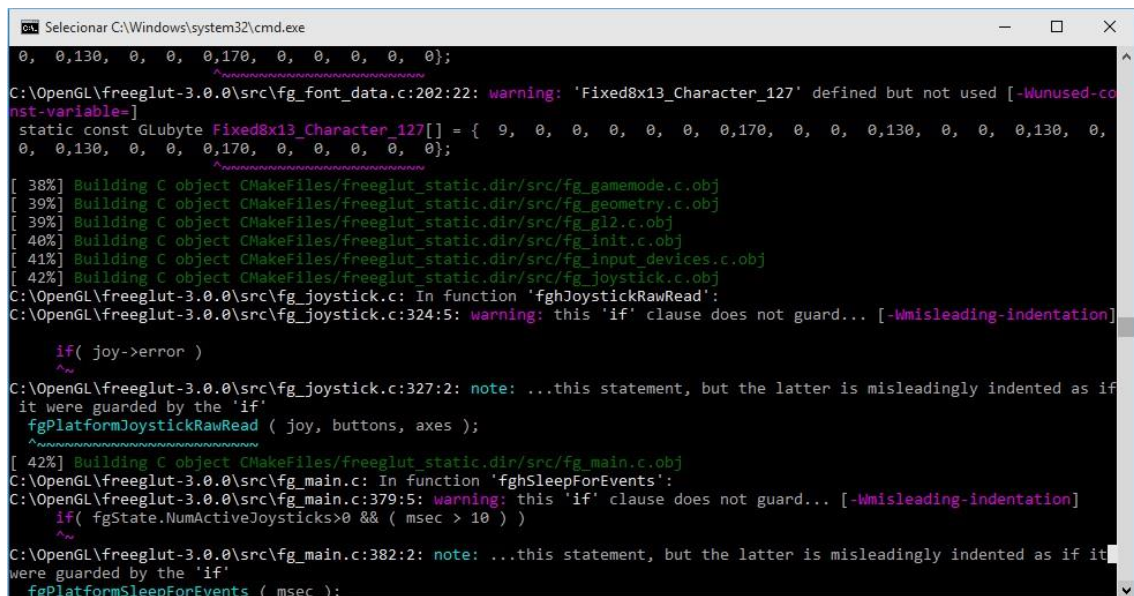
1.3.1 GLEW

OpenGL Extension Wrangler Library (GLEW), providência um mecanismo de tempo de execução eficiente para determinar qual extensão do OpenGL é suportada na plataforma alvo. GLEW tem sido testada em variados sistemas operacionais, incluindo Windows, Linux, Mac OS X, FreeBSD, IRIX e Solaris[2].

- 1) baixe o arquivo binário da biblioteca no link:
<<http://glew.sourceforge.net/index.html> > ;
- 2) Vá na pasta onde se encontra o arquivo “CMakeList.txt” e execute os três comandos a seguir:
 - a. **cmake -G "MinGW Makefiles" -S . -B . -DCMAKE_INSTALL_PREFIX=C:\mingw64\x86_64-w64-mingw32**
 - b. **mingw32-make all**
 - c. **mingw32-make install**

A Figura 1 mostra um exemplo do que se tem no terminal ao decorrer do processo de instalação da biblioteca.

Figura 1: Resultado da Saída final



```
0, 0,130, 0, 0, 0,170, 0, 0, 0, 0, 0};
C:\OpenGL\freeglut-3.0.0\src\fg_font_data.c:202:22: warning: 'Fixed8x13_Character_127' defined but not used [-Wunused-const-variable=]
static const GLubyte Fixed8x13_Character_127[] = { 9, 0, 0, 0, 0, 0, 0, 0,170, 0, 0, 0,130, 0, 0, 0,130, 0,
0, 0,130, 0, 0, 0,170, 0, 0, 0, 0, 0};
[ 38%] Building C object CMakeFiles/freeglut_static.dir/src/fg_gamemode.c.obj
[ 39%] Building C object CMakeFiles/freeglut_static.dir/src/fg_geometry.c.obj
[ 39%] Building C object CMakeFiles/freeglut_static.dir/src/fg_gl2.c.obj
[ 40%] Building C object CMakeFiles/freeglut_static.dir/src/fg_init.c.obj
[ 41%] Building C object CMakeFiles/freeglut_static.dir/src/fg_input_devices.c.obj
[ 42%] Building C object CMakeFiles/freeglut_static.dir/src/fg_joystick.c.obj
C:\OpenGL\freeglut-3.0.0\src\fg_joystick.c: In function 'fghJoystickRawRead':
C:\OpenGL\freeglut-3.0.0\src\fg_joystick.c:324:5: warning: this 'if' clause does not guard... [-Wmisleading-indentation]
    if( joy->error )
    ^
C:\OpenGL\freeglut-3.0.0\src\fg_joystick.c:327:2: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the 'if'
    fgPlatformJoystickRawRead ( joy, buttons, axes );
    ^
[ 42%] Building C object CMakeFiles/freeglut_static.dir/src/fg_main.c.obj
C:\OpenGL\freeglut-3.0.0\src\fg_main.c: In function 'fghSleepForEvents':
C:\OpenGL\freeglut-3.0.0\src\fg_main.c:379:5: warning: this 'if' clause does not guard... [-Wmisleading-indentation]
    if( fgState.NumActiveJoysticks>0 && ( msec > 10 ) )
    ^
C:\OpenGL\freeglut-3.0.0\src\fg_main.c:382:2: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the 'if'
    fgPlatformSleepForEvents ( msec );
    ^
```

Fonte: Autoral

1.3.2 GLFW

A **biblioteca Graphics Library Framework** (GLFW), permite que programas possam criar e gerenciar janelas e contextos OpenGL. Tal biblioteca, também torna possível o uso de múltiplos monitores e modos de vídeo [1].

- 1) Baixe o arquivo binário biblioteca no link (escolha 64bits ou 32bits, de acordo com seu sistema : < <https://www.glfw.org/download.html> >;
- 2) Vá na pasta onde se encontra o arquivo “CMakeList.txt” e execute os três comandos as seguir:
 - a. **cmake -G "MinGW Makefiles" -S . -B . -DCMAKE_INSTALL_PREFIX=C:\mingw64\x86_64-w64-mingw32**
 - b. **mingw32-make all**
 - c. **mingw32-make install**

1.3.3 GLM

OpenGL Mathematics (GLM) é um cabeçario somente para bibliotecas matemáticas em c++. É utilizada para softwares gráficos baseados nas especificações do OpenGL Shading Language (GLSL). Essa biblioteca pode ser usada para se realizar os cálculos de movimentos de objetos de forma automática.

- 1) Baixe o arquivo binário biblioteca no link (escolha 64bits ou 32bits, de acordo com seu sistema : < <https://glm.g-truc.net/0.9.9/index.html> >;
- 2) Vá na pasta onde se encontra o arquivo "CMakeList.txt" e execute os três comandos as seguir:

- a. **cmake -G "MinGW Makefiles" -S . -B . -DCMAKE_INSTALL_PREFIX=C:\mingw64\x86_64-w64-mingw32**
- b. **mingw32-make all**
- c. **mingw32-make install**

1.3.4 SOIL2

Simple OpenGL Image Library 2 (SOIL2) é uma biblioteca em c usada primariamente para subir as texturas dentro do OpenGL. Ela pode ser usada também para salvar e carregar imagens em uma variedade de formatos.

- 1) Baixe o arquivo binário em um dos links abaixo (escolha 64bits ou 32bits, de acordo com seu sistema):

(preferencial) < <https://github.com/Jacob-Tate/soil2-cmake>>

< <https://github.com/SpartanJ/SOIL2>>

< <http://www.lonesock.net/soil.html>>

< <https://bitbucket.org/SpartanJ/soil2/downloads/?tab=tags>>

- 2) Vá na pasta onde se encontra o arquivo "CMakeList.txt" e execute os três comandos as seguir:

- a. **cmake -G "MinGW Makefiles" -S . -B . -DCMAKE_INSTALL_PREFIX=C:\mingw64\x86_64-w64-mingw32**

- b. mingw32-make all
- c. mingw32-make install

A Figura 2 mostra um exemplo no cmd do Windows de como instalar a biblioteca SOIL2.

Figura 2: instalando biblioteca SOIL2

```
C:\OpenGL\soil2-cmake-master>
C:\OpenGL\soil2-cmake-master>cmake -G "MinGW Makefiles" -S . -B . -DCMAKE_INSTALL_PREFIX=C:/mingw64/x86_64-w64-mingw32
-- The CXX compiler identification is GNU 8.1.0
-- The C compiler identification is GNU 8.1.0
-- Check for working CXX compiler: C:/mingw64/bin/g++.exe
-- Check for working CXX compiler: C:/mingw64/bin/g++.exe - works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Check for working C compiler: C:/mingw64/bin/gcc.exe
-- Check for working C compiler: C:/mingw64/bin/gcc.exe - works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: C:/OpenGL/soil2-cmake-master

C:\OpenGL\soil2-cmake-master>mingw32-make all
Scanning dependencies of target soil2
[ 16%] Building C object CMakeFiles/soil2.dir/src/SOIL2/etc1_utils.c.obj
[ 33%] Building C object CMakeFiles/soil2.dir/src/SOIL2/image_DXT.c.obj
[ 50%] Building C object CMakeFiles/soil2.dir/src/SOIL2/image_helper.c.obj
[ 66%] Building C object CMakeFiles/soil2.dir/src/SOIL2/SOIL2.c.obj
[ 83%] Building CXX object CMakeFiles/soil2.dir/src/common/common.cpp.obj
[100%] Linking CXX static library libsoil2.a
[100%] Built target soil2

C:\OpenGL\soil2-cmake-master>mingw32-make install
[100%] Built target soil2
Install the project...
-- Install configuration: ""
-- Installing: C:/mingw64/x86_64-w64-mingw32/lib/libsoil2.a
C:\OpenGL\soil2-cmake-master>
```

Fonte: Autoral

1.4 Execução do Código

Após a etapa de download e instalação das bibliotecas, assim como, os softwares de vinculação necessários, pode-se partir para a execução do código.

Resumidamente, o que o código faz é seguir os seguintes passos:

- 1) iniciar a função da GLFW indicando a criação de uma janela (já verificando algum erro de inicialização);
- 2) configurar as dimensões assim como as cores criando uma variável de referência para a biblioteca **GLFWWindow* Window**, da mesma forma, já configurando as dimensões e cores;
- 3) iniciar a abertura da janela **Window**;

- 4) permanecer em um loop mostrando as janela em vermelho até o momento que ela tenha que ser fechada **while(!glfwWindowShouldClose(window));**
- 5) Destruir o objeto window, encerrar os processos da glfw e fechar o programa.

1.4.1 Execução

Para se executar o programa, basta ir até a pasta onde se encontra o arquivo .cpp e executar o terminal a partir deste caminho. Com o terminal aberto é necessário inserir o comando de compilação, tal comando é mostrado a seguir:

```
>> g++ -o nomedoprogram -Wall nomedoprogram.cpp -lglfw3 -lopengl32 -lglu32 -lgdi32
```

Repare que os locais onde se tem escrito “**nomedoprogram**” devem ser substituídos pelo nome do arquivo .cpp que contem o código fonte a ser executado. Dessa forma, a Figura 3 mostra o resultado retornado pela execução do programa sugerido na atividade.

Figura 3: Execução do programa da atividade



Fonte: Autoral

Referências

[1] WIKIPEDIA. **GLFW**. 2015. Disponível em: <<https://pt.wikipedia.org/wiki/GLFW>>. Acesso em: 18 fev. 2020.

[2] WIKIPEDIA. **GLFW**. 2015. Disponível em: <<http://glew.sourceforge.net/index.html>>. Acesso em: 18 fev. 2020