**CENTRO FEDERAL DE EDUACAÇÃO TECNOLÓGICA**

**CEFET-MG**

**TUTORIAL DE INSTALAÇÃO DO GLUT EM PARA COMPILAÇÃO EM TERMINAL DO WINDOWS (64bits)**

**PEDRO BRANDÃO BELISÁRIO**

**ENGENHARIA DE COMPUTAÇÃO - COMPUTAÇÃO GRÁFICA**

**AGOSTO DE 2015**

**OBJETIVO**

Esse tutorial tem como objetivo demonstrar como configurar um computador com o sistema operacional Windows (64bits), para a compilar, via terminal, um código desenvolvido em linguagem C com a biblioteca GLUT. Ao final, está disponibilizado um exemplo de código do TP0 para teste da configuração do GLUT.

**Antes de começar:**

Nesse tutorial parte-se do pressuposto que já se é possível, em sua máquina, compilar no terminal do Windows utilizando o compilador MinGW.

Ressalva-se o fato de que é para Windows 64bits, e que o código deve ser escrito em linguagem C e salvo como “.c”.

**Tutorial**

1. **Baixando o FreeGlut**
   1. Baixe o FreeGlut em:

http://www.transmissionzero.co.uk/software/freeglut-devel/

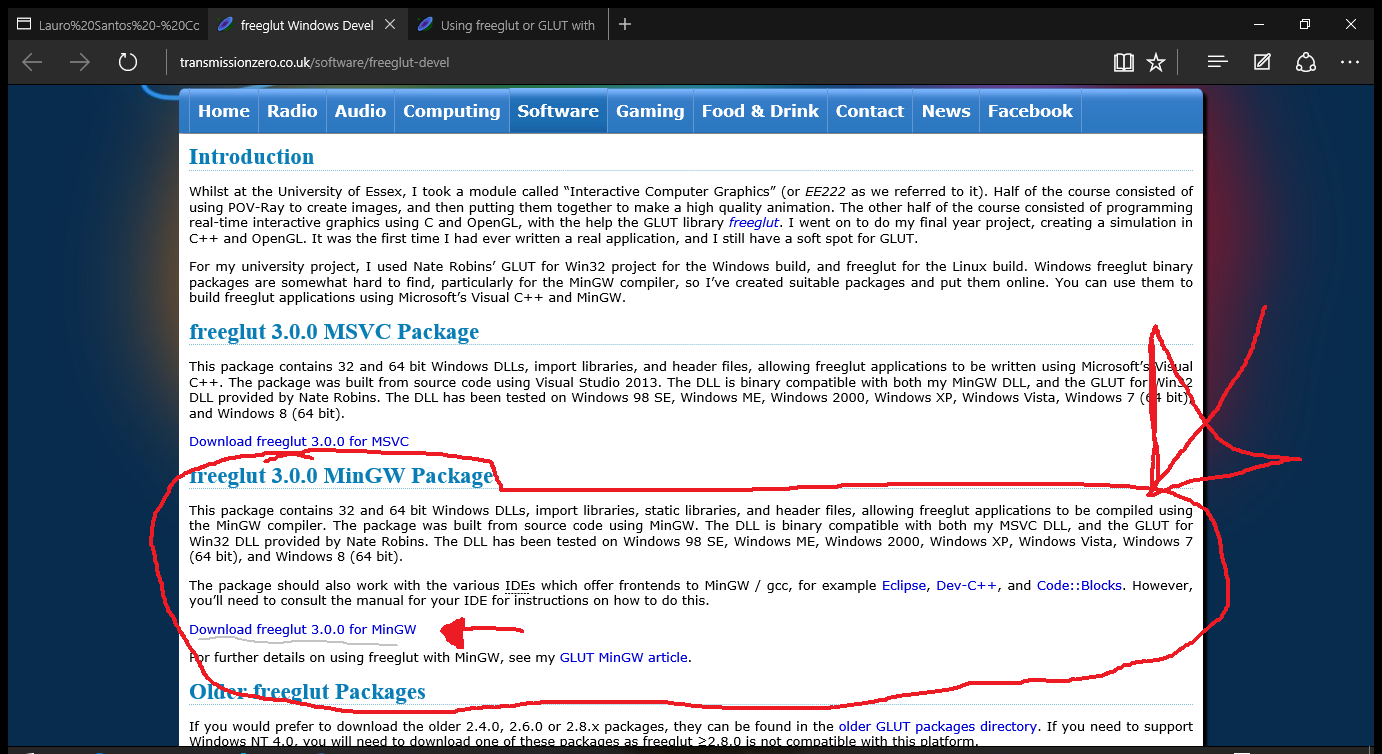
* 1. ****Deve-se baixar a versão “freeglut 3.0.0 MinGW Package"

Figura Ilustração de o que se deve baixar. Disponível em: http://puu.sh/jJezh/19e1394e47.png

* 1. Feito isso descompacte o arquivo baixado

1. **Configurando o Glut para o seu computador**
   1. Crie uma pasta com o nome “MinGW” disponível para todos os usuários de seu computador em “C:\Program Files (x86)\Common Files\”.
   2. Feito isso copie os arquivos “lib” e “include” do arquivo que foi baixado para essa pasta criada.
   3. Crie uma nova pasta com o nome “bin” dentro do MinGW e copie para ela o arquivo “freeglut.dll”. (Figura 2)
   4. Vá para a pasta “C:\Windows\SysWOW64” e também copie o arquivo “freeglut.dll” para essa pasta.

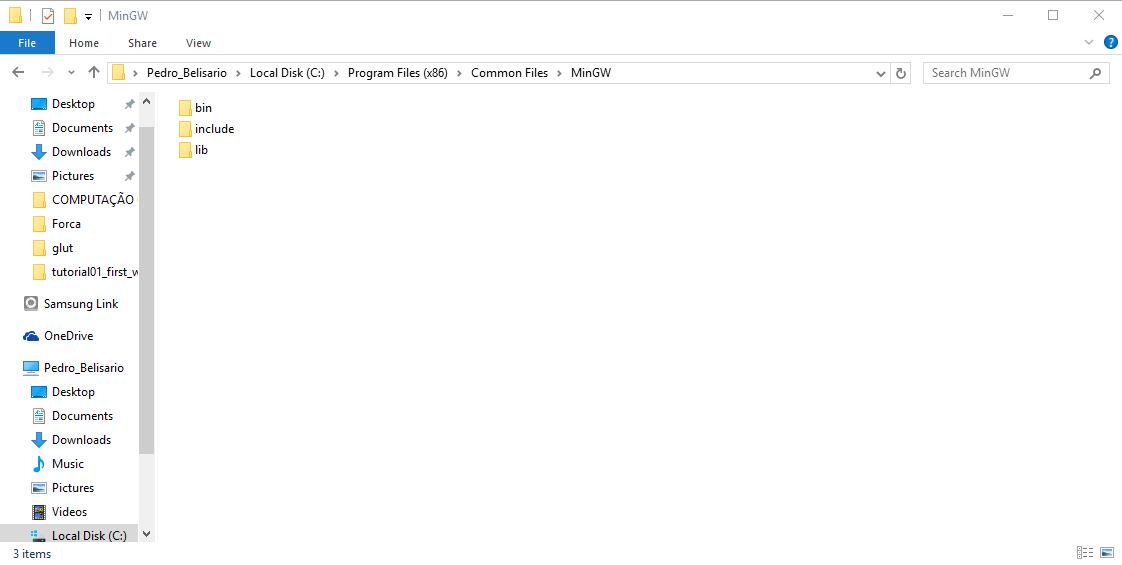
****

Figura Pasta MinGW ao final da configuração do free glut.

Disponível em: http://puu.sh/jJgbh/5c47862001.png

1. **Compilando aplicações freeglut com o MinGW**
   1. Para criar uma aplicação em glut deve-se incluir a biblioteca glut.h em seu código, na forma #include <GL/glut.h>.
   2. Para compilar o código, ao contrário da linguagem C, se faz necessário mais passos e comandos. Dessa forma recomenda-se salvar o esqueleto do comando necessário.
   3. A primeira linha de compilação deve ser essa:

gcc -c -o NOME\_DO\_CODIGO.o NOME\_DO\_CODIGO.c -I"C:\Program Files(x86)\Common Files\MinGW\GLUT\include"

E deve-se substituir o texto *NOME\_DO\_CODIGO* pelo nome dado ao arquivo com o código a ser compilado. (Figura 3)

* 1. A segunda linha de compilação deve ser essa:

gcc -o NOME\_DO\_CODIGO.exe NOME\_DO\_CODIGO.o -L"C:\Program Files(x86)\Common Files\MinGW\GLUT\lib" -lglut32 -lopengl32 -Wl,--subsystem,windows

E deve-se substituir o texto *NOME\_DO\_CODIGO* pelo nome dado ao arquivo com o código a ser compilado. (Figura 3)

* 1. Feito isso será criado de forma automática, na pasta em que está salvo o código, um arquivo de extensão “.o” e outro arquivo de extensão “.exe”. Para abrir a extensão “.exe” basta clicar nela, ou no próprio terminal escrever o seu nome sem sua extensão. (Figura 3)

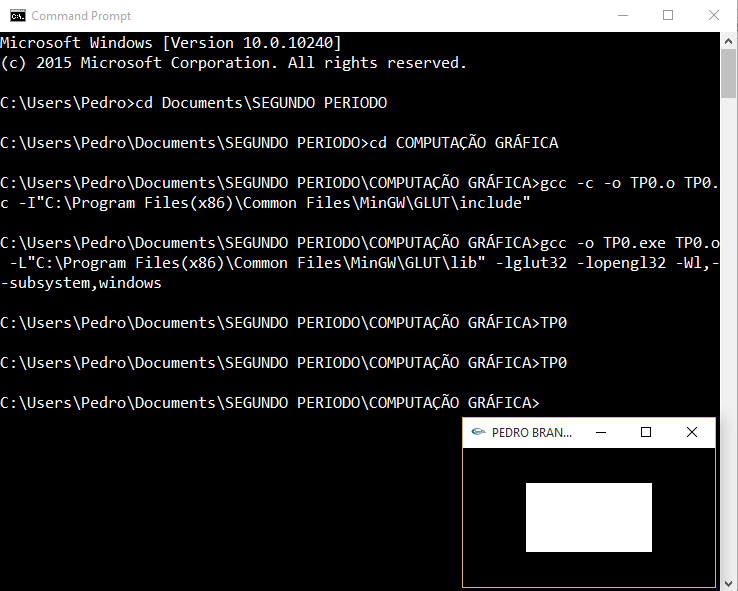
****

Figura Compilação do código e a janela a ser aberta. Disponível em: http://puu.sh/jJhpf/e6a165af9c.png

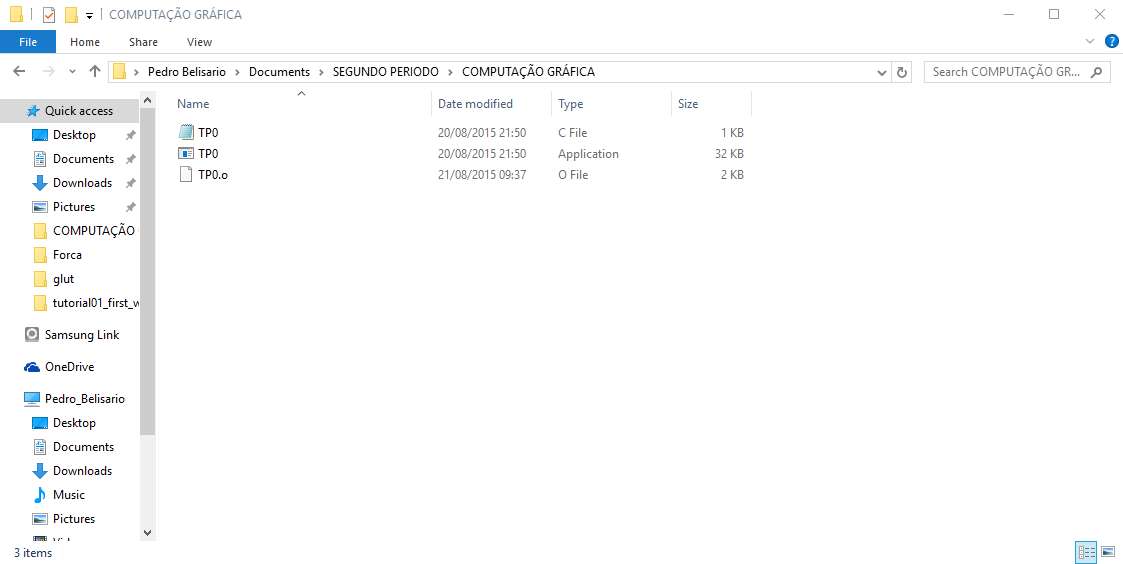
****

Figura Pasta em que está salvo o código após a sua compilação.

Disponível em: http://puu.sh/jJhen/ec315f2fdb.png

1. **Exemplo de código**

#include <stdlib.h>

#include <GL/glut.h>

void keyboard(unsigned char key, int x, int y);

void display(void);

int main(int argc, char\*\* argv){

glutInit(&argc, argv);

glutCreateWindow("PEDRO BRANDAO BELISARIO"); //Cria uma janela com nome a sua escolha

glutKeyboardFunc(&keyboard);

glutDisplayFunc(&display);

glutMainLoop();

return EXIT\_SUCCESS;

}

//Procedimento em que se configura o botao <ESC> para fechar a janela

void keyboard(unsigned char key, int x, int y){

switch (key) {

case '\x1B':

exit(EXIT\_SUCCESS);

break;

}

}

//Procedimento em que se cria a imagem a ser visualizada na janela criada

void display(){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0f, 1.0f, 1.0f); // definição percentual das cores RGB

glBegin(GL\_POLYGON); //posicionamento de um polígono no plano cartesiano

glVertex2f(-0.5f, -0.5f);

glVertex2f( 0.5f, -0.5f);

glVertex2f( 0.5f, 0.5f);

glVertex2f(-0.5f, 0.5f);

glEnd();

glFlush();

}