

Computação Gráfica

Tutorial OpenGL

Professora: Gilda Aparecida de Assis
gilda1@gmail.com

GLUT

- GLUT (OpenGL Utility Toolkit) é uma biblioteca de funcionalidades para criação e controle de janelas e também tratamento de eventos de entrada (mouse e teclado).
- A GLUT permite criar uma janela para gráficos OpenGL e manipular as imagens geradas.
- É muito útil e simples, e embora tenha sido descontinuada a sua manutenção, ainda serve para esse propósito.

Como alterar propriedades da janela glut

//Alterar o tamanho da janela antes de criá-la

```
GlutInitWindowSize(600, 400);
```

//Alterar a posição da janela antes de criá-la

```
glutInitWindowPosition(100, 0);
```

//Cria a janela

```
glutCreateWindow("Meu programa OpenGL – Gilda");
```

//Função que redimensiona a janela já criada

```
glutReshapeWindow(400, 600);
```

//Função que modifica a posição inicial da janela já criada

```
glutPositionWindow(500, 200);
```

OPENGL

- É uma biblioteca de rotinas gráficas de modelagem, manipulação de objetos e exibição tridimensional.
- Foi criada em 1992 pela Silicon Graphics para ser uma API gráfica independente de dispositivo de exibição.
- OpenGL tem se tornado um padrão na indústria de desenvolvimento de aplicações. Diversos jogos, aplicações científicas e comerciais tem utilizado OpenGL, principalmente com a adoção deste padrão por parte dos fabricantes de placas de vídeo.
- Todas as rotinas OpenGL são implementadas em C.

OpenGL como máquina de estados

- Um estado não é alterado, a menos que uma função seja chamada para isto.
- OpenGL mantém uma série de variáveis de estado, tais como estilo (ou padrão) de uma linha, posições e características das luzes, e propriedades do material dos objetos desenhados.

```
glEnable (GL_LIGHTING);
```

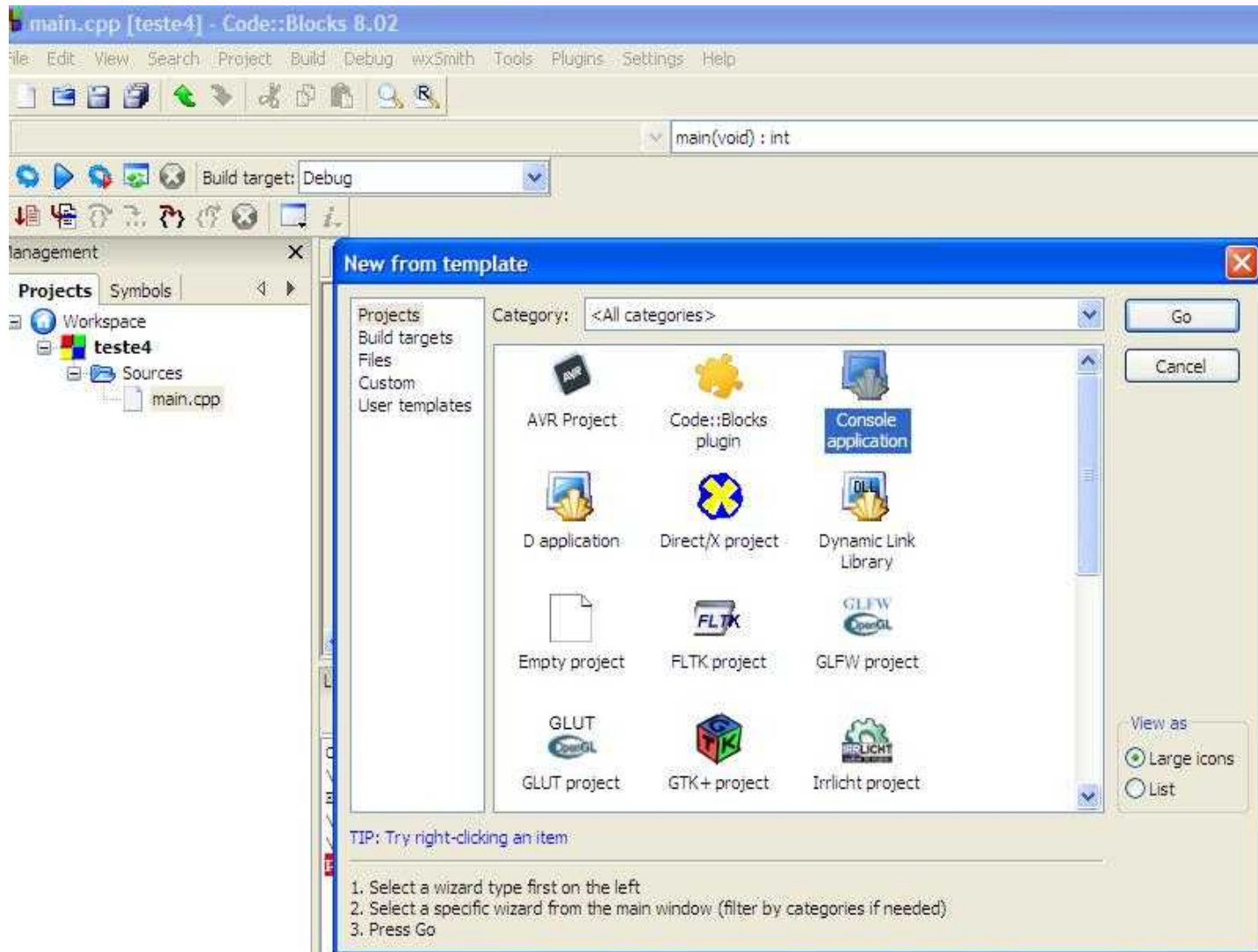
```
glDisable (GL_LIGHTING);
```

```
glEnable(GL_COLOR_MATERIAL);
```

```
glDisable(GL_COLOR_MATERIAL);
```

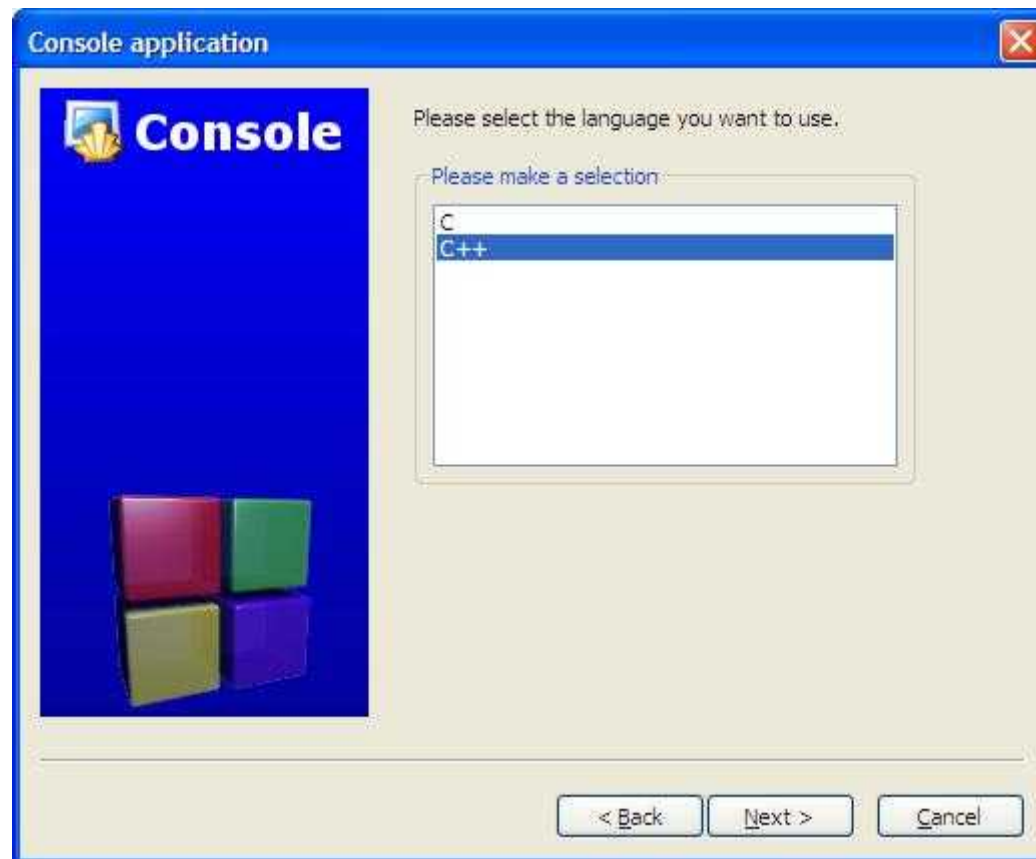
Criar novo projeto no CodeBlocks

- File → New → Project... → Console Application



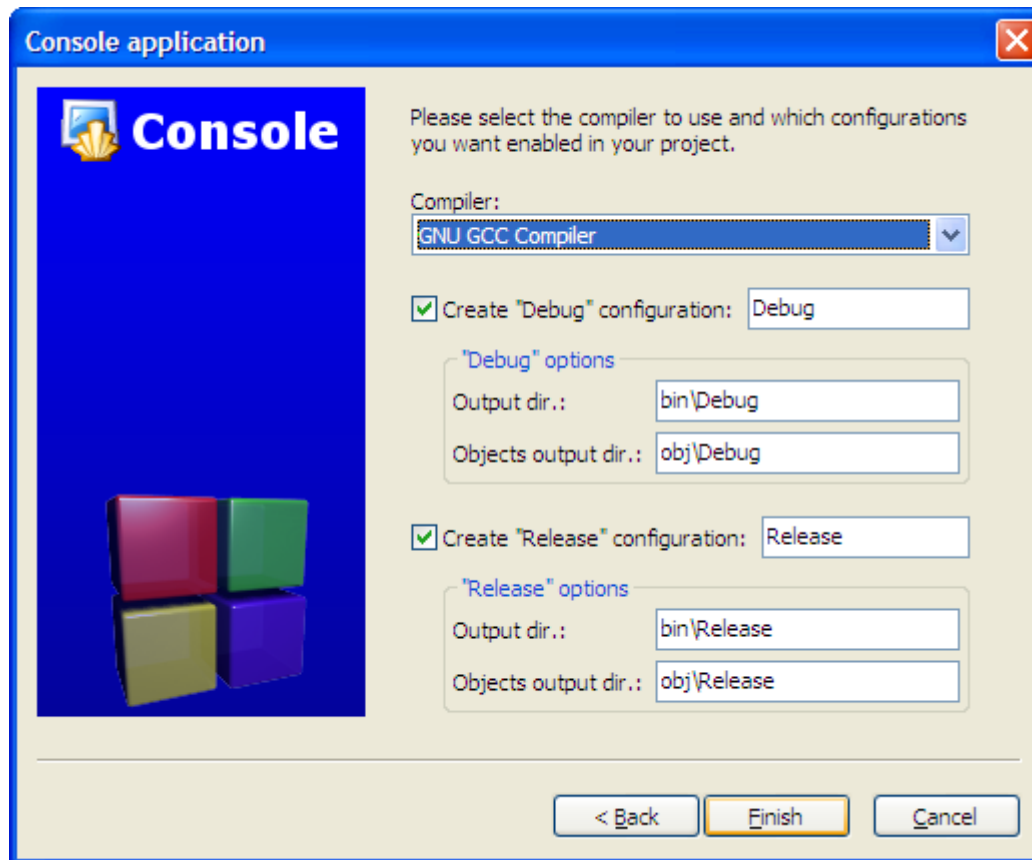
Criar novo projeto no CodeBlocks

- Selecionar a linguagem C++



Criar novo projeto no CodeBlocks

- Escreva o título do projeto e selecione uma pasta para armazenar o projeto
- A seguir, selecione o compilador GCC

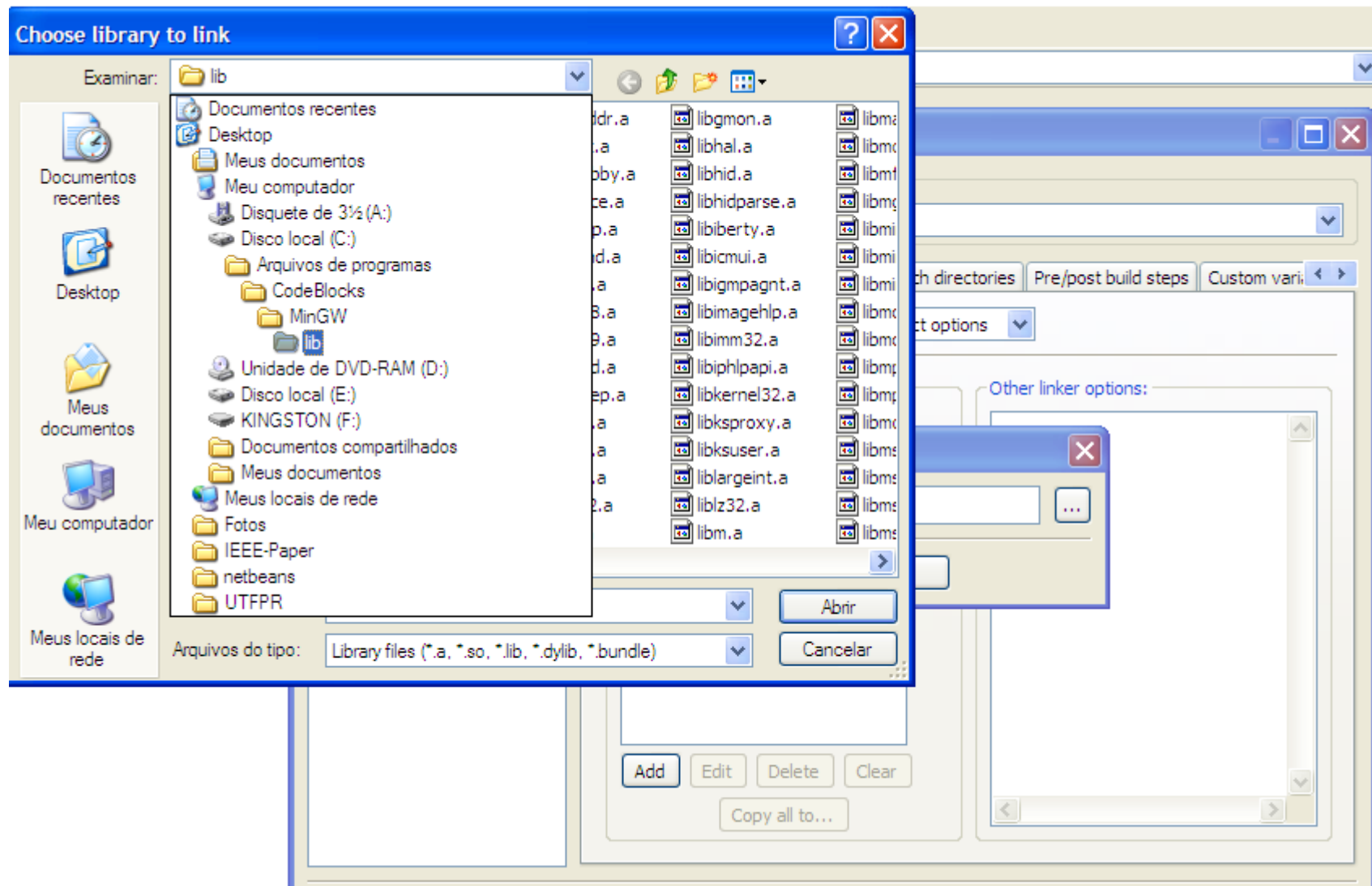


Criar novo projeto no CodeBlocks

- Copiar os arquivos para as seguintes pastas:
 - glut.h para ...\\CodeBlocks\\MinGW\\include\\GL
 - libopengl32.a para ...\\CodeBlocks\\MinGW\\lib
 - libglu32.a para ...\\CodeBlocks\\MinGW\\lib
 - glut32.lib para ...\\CodeBlocks\\MinGW\\lib
 - glut32.dll para C:\\Windows\\System32

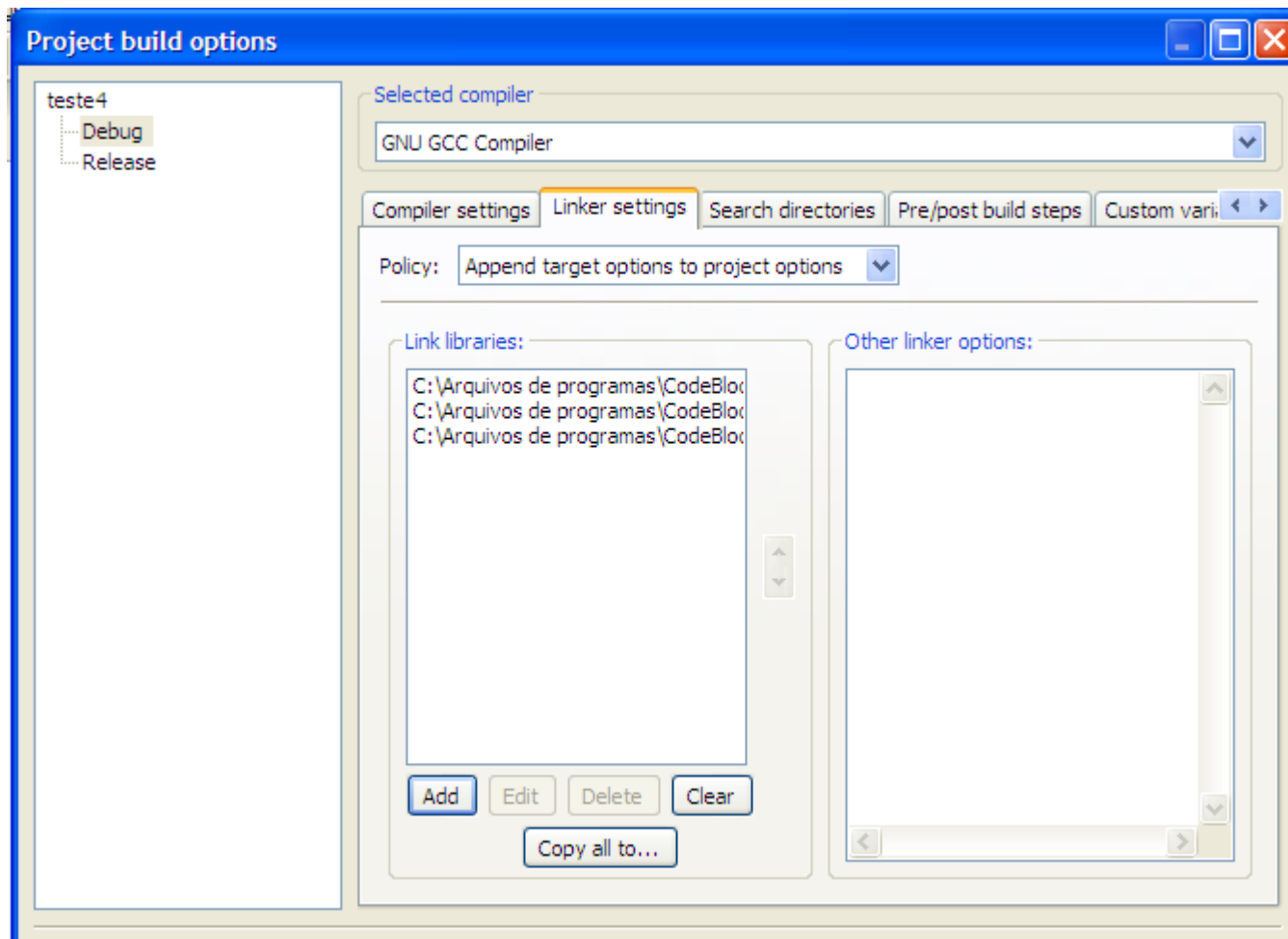
Criar novo projeto no CodeBlocks

- Project → Build Options... Linker settings (aba) → Add (botão) → ... (botão)



Criar novo projeto no CodeBlocks

- Adicionar: **glut32.lib, libopengl32.a, libglu32.a**



Exemplo de Código Opengl - 1

main.cpp

```
#include <windows.h>
#include <gl/glut.h>
#include <gl/gl.h>

// Função de callback de desenho
// Executada sempre que é necessario re-exibir a imagem
void RenderScene(void)
{
    // Limpa a janela com a cor especificada como cor de fundo
    glClear(GL_COLOR_BUFFER_BIT);

    // Indica a cor que todas as primitivas geométricas deste ponto em diante devem ter
    glColor3f (1.0f, 0.0f, 1.0f);

    glBegin (GL_TRIANGLES); // Chamadas de funções OpenGL para desenho
        glVertex2f (0,1);
        glVertex2f (0,-1);
        glVertex2f (1,0);
    glEnd();

    // Flush dos comandos de desenho que estejam no "pipeline" da OpenGL
    glFlush();
}
```

Exemplo de Código Opengl - 1

main.cpp (cont)

```
int main(void)
{
    // Cria uma janela com o titulo especificado
    glutCreateWindow("Meu primeiro programa OpenGL ");

    // Especifica para a OpenGL que funcao é chamada para geracao da imagem
    glutDisplayFunc(RenderScene);

    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // cor de fundo da janela

    // Dispara a "maquina de estados" de OpenGL
    glutMainLoop();
    return(1);
}
```

Exemplo de Código Opengl - 2

main.cpp - 2

```
#include <windows.h>
#include <gl/glut.h>
#include <gl/gl.h>

// Função de callback de desenho
// Executada sempre que é necessario re-exibir a imagem
void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (1.0f, 0.0f, 1.0f);
    glBegin (GL_TRIANGLES); // Chamadas de funções OpenGL para desenho
        glVertex2f (0,1);
        glVertex2f (0,-1);
        glVertex2f (1,0);
        glColor3f (0.0f, 1.0f, 0.0f); glVertex2f (0,1);
        glColor3f (1.0f, 0.0f, 0.0f); glVertex2f (-1,0);
        glColor3f (0.0f, 0.0f, 1.0f); glVertex2f (0,-1);
    glEnd();

    glFlush();
}
```

Exemplo de Código Opengl - 2

main.cpp (cont) - 2

```
int main(void)
{
    // Cria uma janela com o titulo especificado
    glutCreateWindow("Meu primeiro programa OpenGL ");

    // Especifica para a OpenGL que funcao é chamada para geracao da imagem
    glutDisplayFunc(RenderScene);

    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // cor de fundo da janela

    // Dispara a "maquina de estados" de OpenGL
    glutMainLoop();
    return(1);
}
```

Exemplo de Código Opengl - 3

main.cpp - 3

```
#include <windows.h>
#include <gl/glut.h>
#include <gl/gl.h>
int oculta = 0; //variável global, alterada em teclado e usada em RenderScene

void teclado(unsigned char tecla, int x, int y); //função de manipulação de eventos
                                                //do teclado

void RenderScene(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (1.0f, 0.0f, 1.0f);
    glBegin (GL_TRIANGLES); // Chamadas de funções OpenGL para desenho
        glVertex2f (0,1);
        glVertex2f (0,-1);
        glVertex2f (1,0);
        if (!oculta) { // Oculta é uma variável static inicializada com 0
                        // e atualizada para 1 quando uma tecla é pressionada ('-')
            glColor3f (0.0f, 1.0f, 0.0f); glVertex2f (0,1);
            glColor3f (1.0f, 0.0f, 0.0f); glVertex2f (-1,0);
            glColor3f (0.0f, 0.0f, 1.0f); glVertex2f (0,-1);
        }
    glEnd();

    glFlush();
}
```


Exemplo de Código Opengl - 3

main.cpp (cont) - 3

```
void teclado(unsigned char tecla, int x , int y) {  
    if (tecla == '-') { oculta = 1;}  
    if (tecla == '+') { oculta = 0;}  
    glutPostRedisplay();  
}  
  
int main(void)  
{  
    // Cria uma janela com o titulo especificado  
    glutCreateWindow("Meu primeiro programa OpenGL ");  
  
    // Especifica para a OpenGL que funcao é chamada para geracao da imagem  
    glutDisplayFunc(RenderScene);  
    glutKeyboardFunc(teclado);  
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // cor de fundo da janela  
  
    // Dispara a "maquina de estados" de OpenGL  
    glutMainLoop();  
    return(1);  
}
```

Exemplo de Código Opengl - 4

main.cpp (testando outras primitivas)

```
glBegin (GL_QUADS); // Chamadas de funções OpenGL para desenho
glVertex2f (0,1);
glVertex2f (0,-1);
glVertex2f (1,0);
glVertex2f (-1,0);
glEnd();
```

```
glBegin (GL_QUADS); // Chamadas de funções OpenGL para desenho
glVertex2f (0,0);
glVertex2f (0.5, 0);
glVertex2f (0.5,0.5);
glVertex2f (0,0.5);
glEnd();
```

Outras primitivas OpenGL

GL_LINES

GL_LINE_STRIP

GL_LINE_LOOP

GL_POLYGON

GL_TRIANGLES

GL_TRIANGLE_STRIP

GL_TRIANGLE_FAN

GL_QUADS

GL_QUAD_STRIP

Interação com OpenGL

São associadas ações aos eventos do usuário

- Eventos de mouse

- Eventos de teclado

- Eventos de joystick

As ações podem ser:

- Mover ([glTranslatef](#))

- Girar ([glRotatef](#))

- Mudar a escala ([glScalef](#))

- Mudar a cor ou material ([glMaterial](#))

- Mover a câmera ([gluLookAt](#))

Etc...

Funções de callback para interação

`glutMouseFunc` (Nome_da_função_gerencia_mouse)

`glutKeyboardFunc` (Nome_função_gerencia_teclado)

As funções que gerenciam os eventos do teclado e mouse devem alterar variáveis globais que são usadas na função de Desenho e não redesenhar o objeto .

Nas funções de teclado e mouse deve ser incluído o comando `glutPostRedisplay();` para redesenhar a janela.

Exemplo de função callback

```
void teclado(unsigned char tecla, int x, int y)
{
    switch (tecla) {
        case '-': dx = dx - 0.1; glutPostRedisplay(); break;
        case '+': dx = dx + 0.1; glutPostRedisplay(); break;
    }
}
```

Para chamar a função `teclado` incluir na `main()`, após a criação da janela e antes do comando `glDisplayFunc(...)`:

```
glutKeyboardFunc(teclado);
```

Na função de desenho: `glTranslatef(dx, 0, 0);`

Exemplo de Código Opengl - 5

Exemplo de desenho de formas 3D pré-definidas em OpenGL

```
#include <windows.h>
#include <gl/glut.h>
#include <gl/gl.h>

void RenderScene(void)
{
    glClear (GL_COLOR_BUFFER_BIT);

    glMatrixMode (GL_PROJECTION );
    gluPerspective(45,1,0.1,500);
    glMatrixMode(GL_MODELVIEW);
    gluLookAt(0,80,200, 0,0,0, 0,1,0);

    GLfloat light_ambient[ ] = { 0.8, 0.8, 0.8, 1.0 };
    GLfloat light_diffuse[ ] = { 1.0, 0.0, 0.0, 1.0 };
    GLfloat light_position[ ] = { 1.0, 1.0, 1.0, 0.0 };
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glLightfv (GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv (GL_LIGHT0, GL_POSITION, light_position);
    glEnable (GL_LIGHTING);
    glShadeModel (GL_FLAT);
    glEnable (GL_LIGHT0);

    glutSolidTeapot (50);
    glFlush();
}
```

Exemplo de Código Opengl - 5

Exemplo de desenho de formas 3D pré-definidas em OpenGL (cont)

```
int main(void)
{
    // Cria uma janela com o titulo especificado
    glutCreateWindow("Meu primeiro programa OpenGL ");

    // Especifica para a OpenGL que funcao é chamada para geracao da imagem
    glutDisplayFunc(RenderScene);

    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // cor de fundo da janela

    // Dispara a "maquina de estados" de OpenGL
    glutMainLoop();
    return(1);
}
```


Tarefa no Moodle

Escreva um programa que utiliza a API OpenGL para desenhar um objeto 2D e que utiliza teclas do teclado alfanumérico ou teclas de direção para mover o objeto para direita, esquerda, cima e baixo.

- Faça o upload do código na tarefa prática 1 disponível no Moodle