

Introdução a OpenGL



Renato Rodrigues Oliveira da Silva

Danilo Medeiros Eler

Adaptado do material de

Marcela X. Ribeiro

Maria Cristina F. de Oliveira

Rosane Minghim

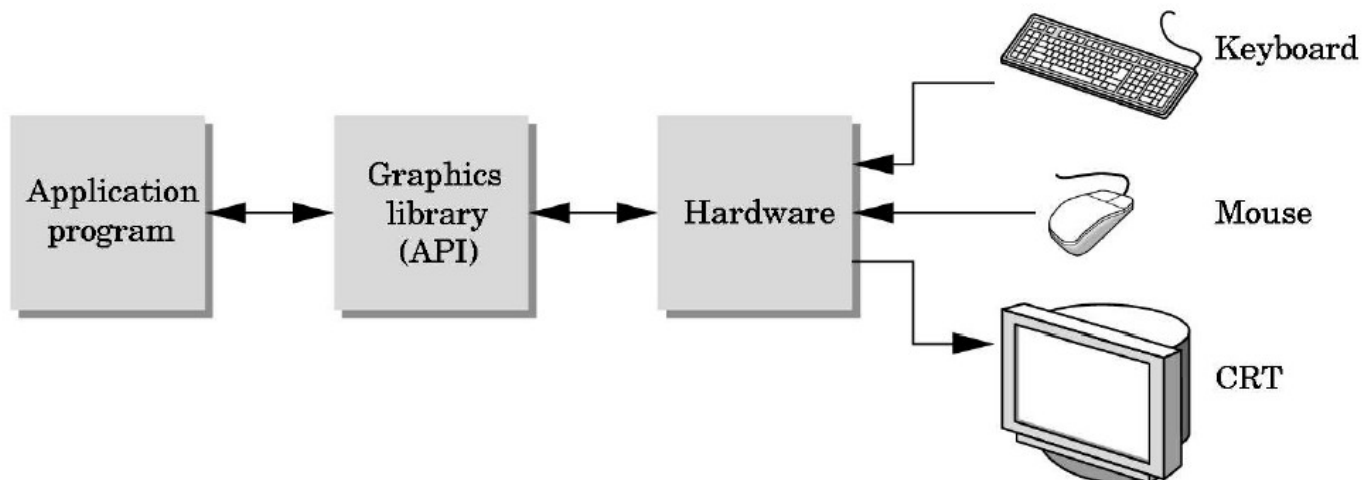
Sumário

- # Introdução
- # Núcleo OpenGL, GLU, GLUT
- # Ambiente 2D
- # Ambiente 3D
- # Tratamento de Eventos

Introdução – O que é OpenGL?

OpenGL – *Open Graphics Library*

- É uma API (*Application Program Interface*) para o aplicações gráficas
- Abstrai a complexidade do hardware

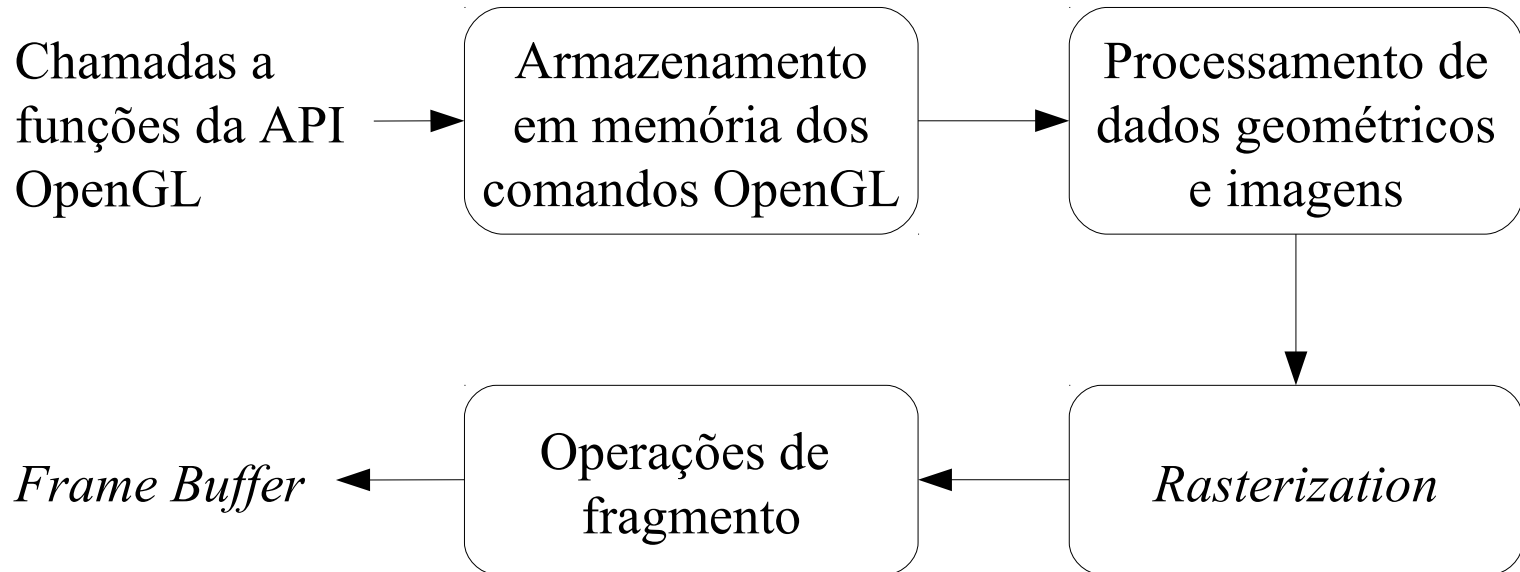


Introdução - Histórico

- # Especificação gerenciada por um consórcio independente formado em 1992
 - Constituído por empresas líderes na área: NVIDIA, 3Dlabs, Apple Computer...
 - Responsável pela aprovação de novas funcionalidades, versões e extensões da OpenGL
 - <http://www.opengl.org>

Núcleo OpenGL

Pipeline



Exemplo – Primitivas Geométricas

Triângulos

```
glBegin (GL_TRIANGLES) ;  
    glVertex2f (x1, y1, z1) ;  
    glVertex3f (x2, y2, z2) ;  
    glVertex3f (x3, y3, z3) ;  
glEnd () ;
```

Exemplo – Primitivas Geométricas

Quadriláteros

```
glBegin (GL_QUADS) ;  
    glVertex3f (x1, y1, z1) ;  
    glVertex3f (x2, y2, z2) ;  
    glVertex3f (x3, y3, z3) ;  
    glVertex3f (x4, y4, z4) ;  
glEnd () ;
```

Exemplo – Primitivas Geométricas

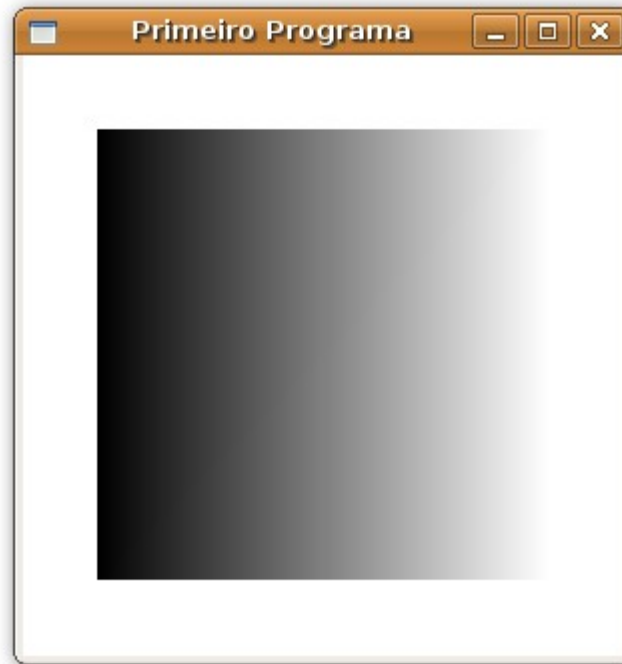
exemploPrimitivasGeometricas.c

OpenGL: Máquina de Estados

- # OpenGL rastreia diversas variáveis de estado
 - Tamanho atual de um ponto, cor de fundo da janela, cor do desenho
 - O valor corrente permanece ativo até que seja alterado
 - Tamanho de ponto: `glPointSize(3.0)`
 - Cor de traçado: `glColor3f(red, green, blue)`
 - Cor de fundo: `glClearColor(r, g, b, alpha)`
 - ...

Exemplo OpenGL

exemploEstadosOpenGL.c



Padronização dos nomes das funções

Convenção:

<Prefixo Biblioteca> <Comando Raiz>
<Contador de Argumentos> <Tipo Argumentos>

Visa padronizar e facilitar a utilização

Possibilita identificar a qual biblioteca a função pertence, quantos argumentos possui e quais são os tipos dos argumentos.

Padronização dos nomes das funções

Exemplo

```
void glColor3f(GLfloat red, GLfloat green,  
              GLfloat blue)
```

//gl - É o prefixo da biblioteca GL

//Color - comando objetivo da função

//3 - contador para o número de argumentos

//f - indica o tipo dos argumentos

Tipos de Dados

- # Tipos de dados próprios para OpenGL
- # Tornam o programa-fonte portátil

sufixo	Tipo	tipo C	nome
b	inteiro 8 bits	signed char	GLbyte
s	inteiro 16 bits	short	GLshort
i	inteiro 32 bits	int/long	GLint
f	float 32 bits	float	GLfloat
...

Tipos de Dados

//Perigo!!: sistema passa int...

```
void drawDot(int x, int y) {  
    glBegin(GL_POINTS);  
    glVertex2i(x, y); // função “espera” inteiro 32 bits  
    glEnd();  
}
```

// código seguro.

```
void drawDot(GLint x, GLint y) {  
    glBegin(GL_POINTS);  
    glVertex2i(x, y);  
    glEnd();  
}
```

GLU – OpenGL Utility Library

- # Instalada junto com a OpenGL
- # Contém uma série de funções que encapsulam comandos OpenGL de mais baixo nível
 - # Definição de matrizes para projeção
 - # Desenho de superfícies quádricas
 - # Curvas e superfícies NURBS

GLUT – OpenGL Utility Toolkit

- # Por ser portátil, a OpenGL não possui funções para gerenciamento de janelas, tratamento de eventos ou manipulação de arquivos
 - Podem ser utilizadas as funções específicas de cada plataforma
 - Ou usar uma biblioteca independente (GLUT, FLTK, wxWidgets, por exemplo)

GLUT – OpenGL Utility Toolkit

- # Biblioteca que inclui alguns elementos de interface gráfica com o usuário
 - Criação de janelas e menus pop-up
 - Gerenciamento de eventos de mouse e teclado
- # Independente de plataforma

GLUT - Inicialização

- # void glutInitDisplayMode(parâmetros)
 - GLUT_DOUBLE: Define que a GLUT usará dois buffers de cor
 - GLUT_SINGLE: Define apenas um buffer de cor
 - GLUT_DEPTH: Define o uso de um buffer de profundidade (z-buffer), para remoção de superfícies escondidas
 - GLUT_RGB: Define que as cores são especificadas por componentes RGB

GLUT - Inicialização

- # **void glutInitWindowPosition(int x, int y)**
 - ▣ Define a posição inicial da janela. Os parâmetros representam o canto superior esquerdo
- # **void glutInitWindowSize(int width, int height)**
 - ▣ Define a largura e altura da janela
- # **int glutCreateWindow(char *string)**
 - ▣ Cria a janela e define o seu título

GLUT - Inicialização

Exemplo

```
int main(int argc, char *argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE |
                        GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutCreateWindow("Primeiro
                    Programa");
    glutMainLoop();
}
```

GLUT - Exemplo

```
# exemploEstadosOpenGL.c
```



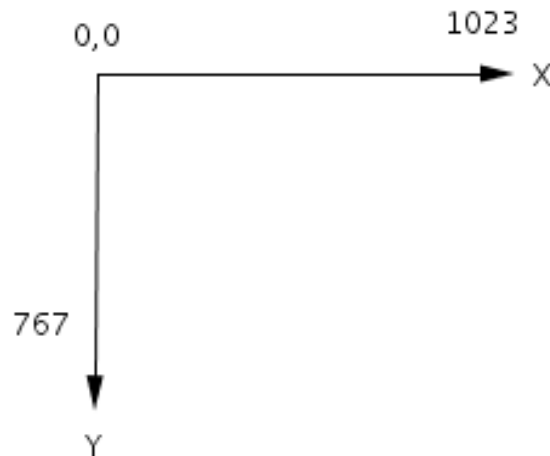
Ambiente de Trabalho

Ambiente de Trabalho

- # Em OpenGL os objetos são representados no Sistema de Referência do Universo (SRU)
 - Plano cartesiano bi ou tridimensional, com eixos se interceptando na origem
- # Todos os comandos e modelos são definidos em relação a este sistema de referência

Ambiente de Trabalho

- # No monitor do computador é adotado o SRT (Sistema de Referência da Tela)
 - No SRT a origem fica no canto superior esquerdo do monitor



Ambiente de Trabalho - 2D

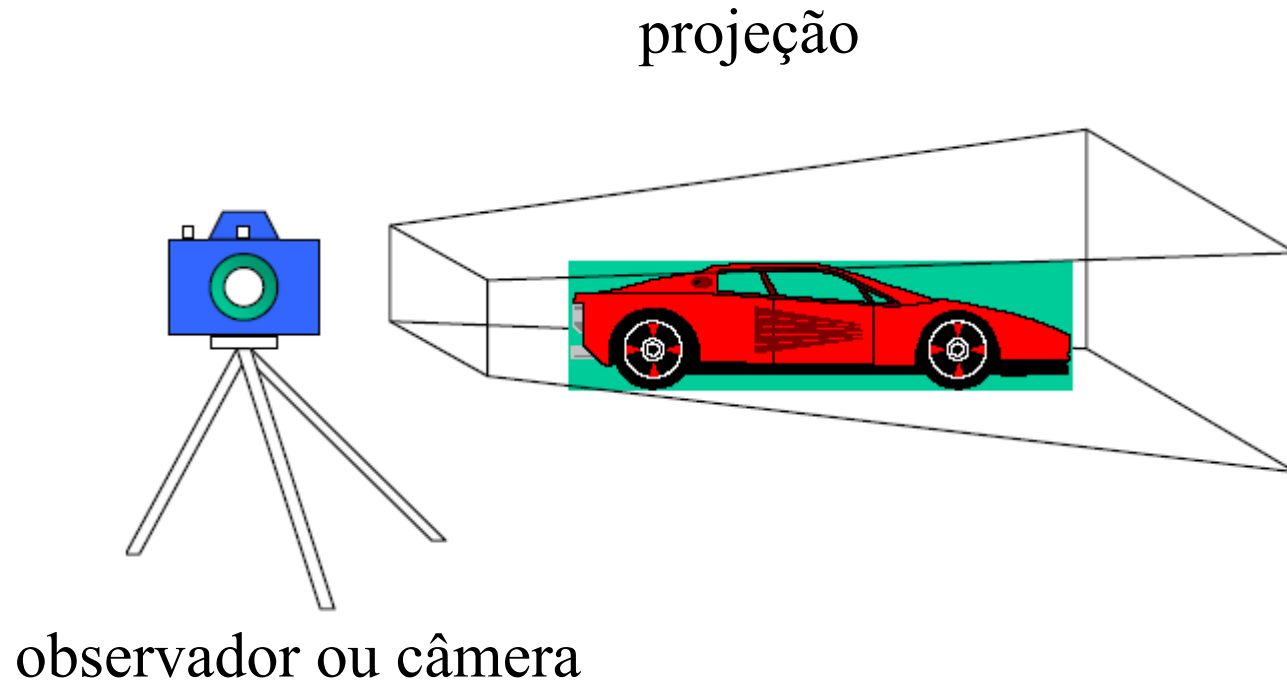
- # No caso 2D, é necessário definir a porção o universo que desejamos mapear na tela.
 - Essa área é chamada de janela de seleção, ou *window*

```
void gluOrtho2D(GLdouble left, GLdouble right,  
                GLdouble bottom, GLdouble top)
```

Ambiente de Trabalho - 2D

exemploPrimitivasGeometricas.c

Ambiente de Trabalho - 3D



Ambiente de Trabalho - 3D

A GLU oferece a seguinte função para posicionar e orientar a câmera

```
void gluLookAt(GLdouble obsx, obsy, obsz,  
               alvox, alvoy, alvoz,  
               upx, upy, upz)
```

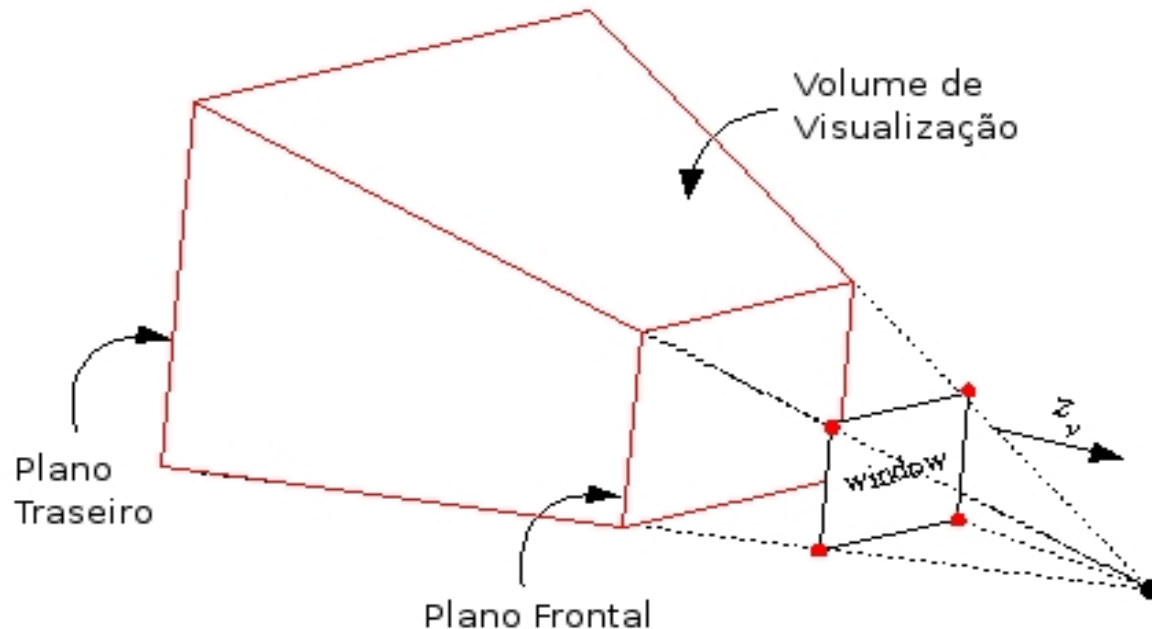
//obs*: define a posição da câmera

//alvo*: ponto para onde o observador está olhando

//up*: vetor que indica a 'vertical' da câmera

Projeção Perspectiva

- # As projetantes emanam de um único ponto, a uma distância finita do plano de projeção



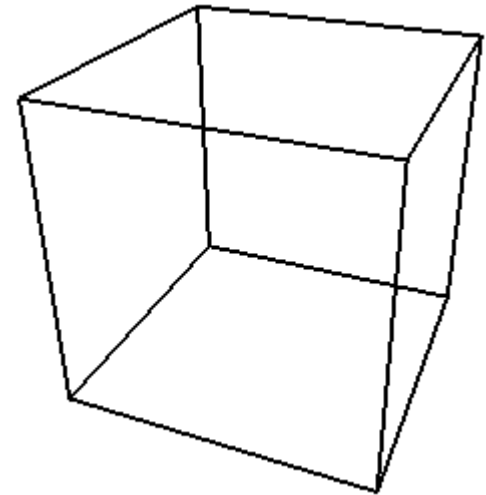
Projeção Perspectiva

```
void gluPerspective(GLdouble fovy, aspect,  
                  zNear, zFar)  
  
//fovy: ângulo de abertura da câmera em y  
//aspect: área de visualização em x  
//zNear: distância do observador ao plano  
         de corte frontal  
//zFar: distância do observador ao plano  
        de corte traseiro
```

Projeção Perspectiva

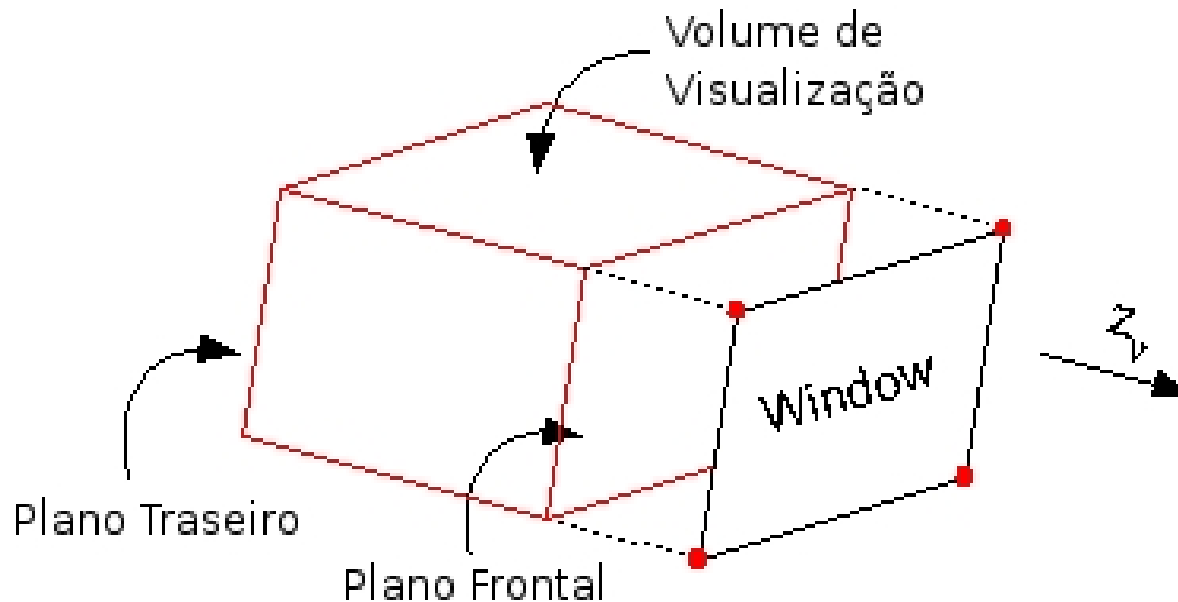
Exemplo

```
void Desenha(){
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60, 1.0f, 0.5, 500);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(40, 60, 100, 0, 0, 0, 0, 1, 0);
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0f, 0.0f, 1.0f);
    glutWireCube(50);
    glFlush();
}
```



Projeção Paralela

- # As projetantes são paralelas entre si
- # Não há alteração nas medidas do objeto



Projeção Paralela

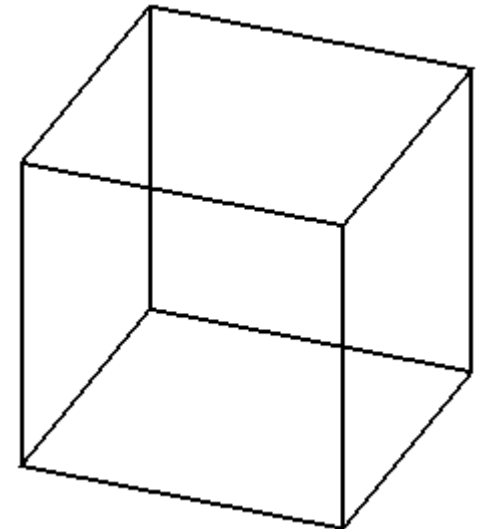
```
void gluOrtho(GLdouble left, right,  
              bottom, top, near, far)
```

```
/*Os parâmetros definem os limites mínimo e  
máximo da janela de projeção em x, y e z.*/
```

Projeção Paralela

Exemplo

```
void Desenha(){
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-65.0, 65.0, -65.0, 65.0, -400.0, 400.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(40,60,100, 0,0,0, 0,1,0);
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0f, 0.0f, 0.0f);
    glutWireCube(50);
    glFlush();
}
```





GLUT

Tratamento de Eventos

GLUT – Tratamento de eventos

- # Gerenciamento de eventos por meio de funções “callback”
- # Quem chama a função para o tratamento de um evento não é o programador e sim a GLUT
- # O programador define apenas a função a ser chamada, respeitando os parâmetros definidos pela GLUT

GLUT – Tratamento de eventos

- # void glutDisplayFunc(Desenha)
 - ▣ Define que a função Desenha será a responsável por redesenhar a janela, sempre que necessário
 - ▣ A função deve ter o seguinte protótipo
 - ▣ void <nomeFunção> (void)

GLUT – Tratamento de eventos

- # **void glutReshapeFunc (AlteraTamanhoJanela)**
 - ▣ Define a função que será responsável por tratar o evento referente ao redimensionamento da janela
 - ▣ Deve ter o protótipo
 - ▣ void <nomeFunção> (int largura, int altura)

GLUT – Tratamento de eventos

```
int main(int argc, char *argv[]) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE |
                        GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutCreateWindow("Primeiro Programa");
    glutDisplayFunc(Desenha);
    glutReshapeFunc(AlteraTamanhoJanela);
    ...
}
```

GLUT – Interação com o teclado

void glutKeyboardFunc(Teclado)

- Trata o pressionamento de teclas comuns
 - void <nomeFunção> (unsigned char tecla, int x, int y)
 - Tecla representa o código ASCII da tecla pressionada
 - Os parâmetros x e y representam as coordenadas do mouse quando o evento ocorreu

GLUT – Interação com o teclado

Exemplo

exemploTecladoOpenGL.c

GLUT – Interação com o mouse

- # void glutMouseFunc(GerenciaMouse)
 - ▣ Trata eventos de pressionamento e liberação de botões do mouse
 - ▣ void <nomeFunção> (int botao, int estado, int x, int y)
 - ▣ botão: GLUT_LEFT_BUTTON, GLUT_MIDDLE_BUTTON, GLUT_RIGHT_BUTTON
 - ▣ estado: GLUT_UP, GLUT_DOWN

GLUT – Interação com o mouse

Ex. posicionar pontos com o mouse

exemploMouseOpenGL.c

Bibliografia



- # Cohen, M. , Manssour, I. H, OpenGL – Uma Abordagem Prática e Objetiva