OpenGL
Primitivas de Desenho

Chessman Kennedy

OpenGL

- Open Graphic Library.
- É uma biblioteca para a síntese de gráficos 2D e 3D e para modelagem.
- É rápida e tem portabilidade para diferentes sistemas operacionais.
- Site do OpenGL: **www.opengl.org**

Semântica dos Nomes das Funções do OpenGL

- O nome de uma função do OpenGL posui 4 partes:

<prefixo bib><comando raiz><qtd argumentos><tipo dos argumentos>

- Exemlo:
  - **void glColor3f (GLfloat red, Glfloat green, Glfloat blue)**
  - gl = prefixo que representa a bibliotca gl.
  - Color = indica a finalidade de função.
  - 3 = quantidade de parâmetros da função
  - f = informa que os parâmetros são do tipo float.

Limpando a janela com uma cor padrão

- Para definir a cor para a limpar a janela, use a função **glClearColor**.
- Para limpar a janela, user a função **glClear**.
- Observe o exemplo a seguir.

Limpando a janela com uma cor padrão

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

// Inicializa parâmetros de renderização
void inicializar (void)
{
    // Define a cor de fundo da janela de visualização como preta
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
}

// Função callback chamada para fazer o desenho
void desenhar(void)
{
    //Limpa a janela de visualização com a cor de fundo especificada
    glClear(GL_COLOR_BUFFER_BIT);

    //Executa os comandos OpenGL
    glFlush();
}
```

Limpando a janela com uma cor padrão

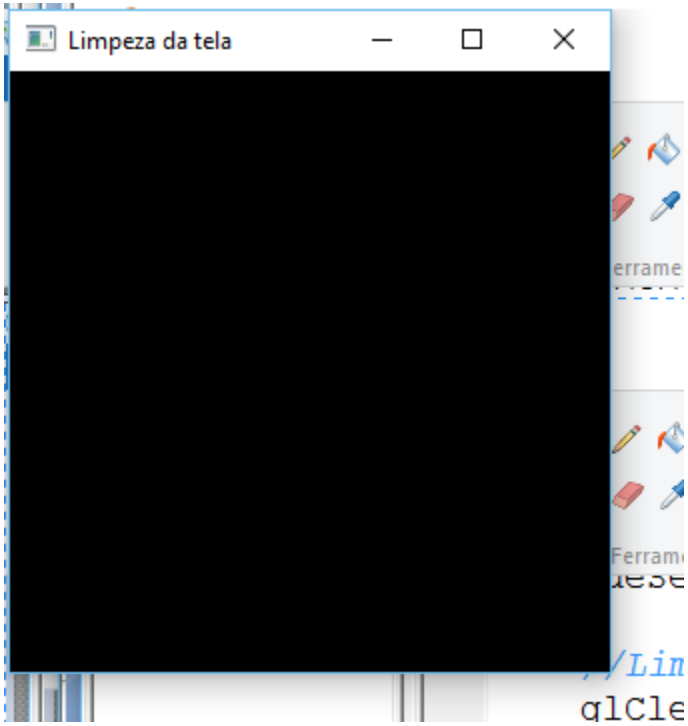
```
int main(void)
{
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow("Limpeza da tela");
    glutDisplayFunc(desenhar);
    inicializar();
    glutMainLoop();
}
```

Limpando a janela com uma cor padrão

Como está sendo gerada uma imagem, está sendo usado um buffer simples

```
int main(void)
{
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow("Limpeza da tela");
    glutDisplayFunc(desenhar);
    inicializar();
    glutMainLoop();
}
```

Limpando a janela com uma cor padrão



Desenhando

- Para fazer qualquer desenho, é necessário chamar as funções glBegin e glEnd:

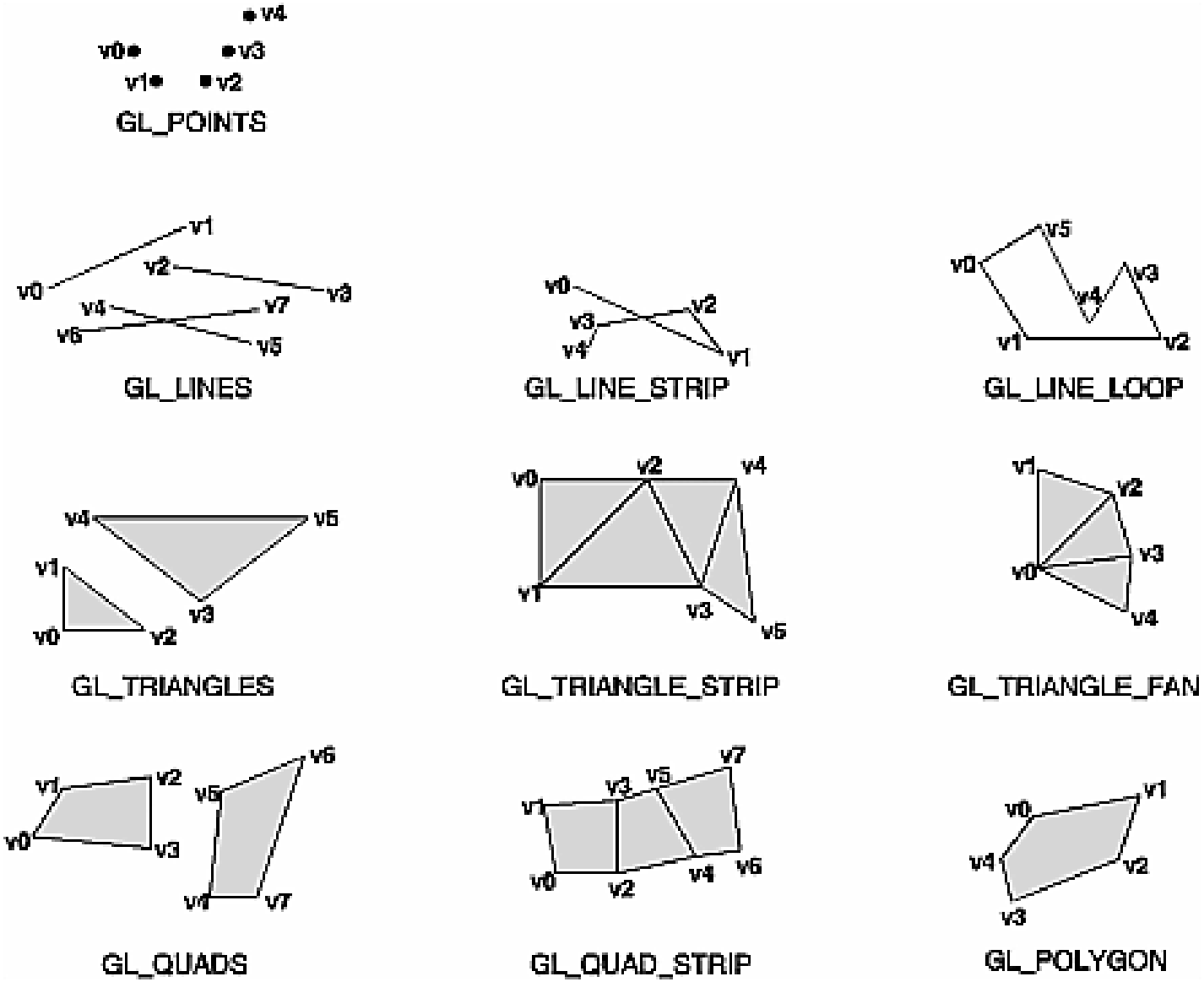
```
glBegin(PRIMITIVA);

glEnd();
```

PIMITIVA informa o objeto a ser desenhado.

Primitivas

Valor	Descrição	Qtd. Vértices
GL_POINTS	Desenho de pontos	1
GL_LINES	Desenho de linhas	2
GL_LINE_STRIP	Desenho de linhas conectadas	2
GL_LINE_LOOP	Desenho de linhas conectadas. O último vértice é conectado ao primeiro.	2
GL_POLYGON	Polígono	3
GL_TRIANGLES	Triângulos	3
GL_TRIANGLE_STRIP	Triângulos conectados.	3
GL_TRIANGLE_FAN	Triângulos criados a partir de um ponto central.	3
GL_QUADS	Quadriláteros.	4
GL_QUAD_STRIP	Quadriláteros conectados	4

Primitivas


v4
v0
v3
v1
v2
GL_POINTS
v1
v0
v2
v3
v4
v7
v6
v5
GL_LINES
v0
v2
v3
v4
v1
GL_LINE_STRIP
v5
v0
v3
v4
v1
v2
GL_LINE_LOOP
v4
v5
v1
v3
v0
v2
GL_TRIANGLES
v0
v2
v4
v1
v3
v5
GL_TRIANGLE_STRIP
v1
v2
v3
v0
v4
GL_TRIANGLE_FAN
v1
v2
v0
v3
v5
v6
v4
v7
GL_QUADS
v7
v3
v5
v1
v0
v2
v4
v6
GL_QUAD_STRIP
v1
v0
v4
v2
v3
GL_POLYGON

Definindo as Coordenadas

- Use a função **glVertex2i (int x,int y)** ou **glVertex2f (float x, float y)** para definir as coordenadas de um vértice (ou de um ponto).

Desenho de Pontos

- Passe a primitiva GL_POINTS para glBegin para desenhar pontos.
- Use uma das funções glVertex2 para desenhar cada ponto.
- Exemplo (slide a seguir):

Desenho de Pontos

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

// Inicializa parâmetros de rendering
void inicializar (void)
{
    // Define a cor de fundo da janela de visualização como preta
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    // Determina a região onde a imagem pode ser exibida.
    // O desenho não aparece se esta função não for usada
    gluOrtho2D (0.0f, 250.0f, 0.0f, 250.0f);
}
```

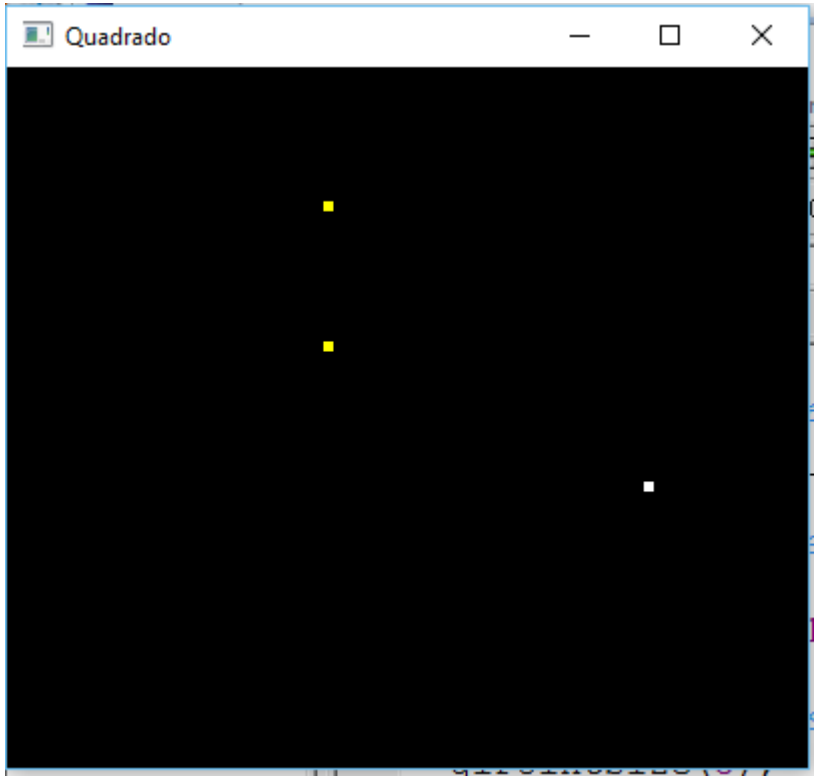
Desenho de Pontos

```
// Função callback chamada para fazer o desenho
void desenhar(void)
{
  glClear(GL_COLOR_BUFFER_BIT);

  // Especifica que a cor corrente é vermelha
  //          R     G     B
  glColor3f(1.0f, 1.0f, 0.0f);

  // Desenha pontos
  glPointSize(5);
  glBegin(GL_POINTS);
    glVertex2i(100,150);
    glVertex2i(100,200);
    glColor3f(1.0f, 1.0f, 1.0f);
    glVertex2i(200,100);
  glEnd();

  // Executa os comandos OpenGL
  glFlush();
}
```

Desenho de Pontos

```
int main(void)
{
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
  glutInitWindowSize(400,350);
  glutInitWindowPosition(10,10);
  glutCreateWindow("Quadrado");
  glutDisplayFunc(desenhar);
  inicializar();

  glutMainLoop();
}
```

Desenho de Pontos



Desenho de Pontos

- Passe a primitiva GL_LINES para glBegin para desenhar pontos.
- Use uma das funções glVertex2 para desenhar cada ponto.
- Exemplo (slide a seguir):

Desenho de Linhas

```
// Função callback chamada para fazer o desenho
void desenhar(void)
{
  glClear(GL_COLOR_BUFFER_BIT);

  // Desenha linhas
  glPointSize(5);
  glBegin(GL_LINES);
    glColor3f(0.20, 1.0, 1.0);
    glVertex2i(100,150);
    glVertex2i(100,200);
    glColor3f(0.20, 0.40, 1.0);
    glVertex2i(10,10);
    glVertex2i(200,200);
  glEnd();
```

Usando Vetores

- O OpenGL tem versões das funções que recebem vetores como entrada.
- São as mesmas funções, com o sufixo v.
- Veja o exemplo a seguir.

Usando Vetores

```
void desenhar(void)
{
  int vertice1[2] = {100,150};
  int vertice2[2] = {100,200};
  int vertice3[2] = {10,10};
  int vertice4[2] = {200,200};

  glClear(GL_COLOR_BUFFER_BIT);
  glPointSize(5);
  glBegin(GL_LINES);
    glColor3f(0.20, 1.0, 1.0);
    glVertex2iv(vertice1);
    glVertex2iv(vertice2);
    glColor3f(0.20, 0.40, 1.0);
    glVertex2iv(vertice3);
    glVertex2iv(vertice4);
  glEnd();
  glFlush();
}
```

Exercícios

- Experimente as outras primitivas.
- Use **glVertex2f (x,y)** para definir as coordenadas de cada vértice de figura;

Desenho de Texto

- O OpenGL não tem suporte para a escrita de texto.
- Para escrever texto, é necessário uma das seguintes tipos de fontes do glut:
  - Tipo Stroke:
    - Baseadas em segmentos de retas.
    - Podem ser redimensionadas e rotacionadas.
    - São mais lentas de serem geradas.
  - Tipo Bitmap
    - Baseadas em imagens.
    - São mais rápidas de serem geradas.

Desenho de Texto Bitmap

- Use glutBitmapCharactere (fonte, string).
- Valores para fonte:

VALOR	DESCRIÇÃO
GLUT_BITMAP_8_BY_13	Fonte padrão 8x13
GLUT_BITMAP_9_BY_15	Fonte padrão 9x15
GLUT_BITMAP_TIMES_ROMAN_10	Fonte times roman tamanho 10
GLUT_BITMAP_TIMES_ROMAN_24	Fonte times roman tamanho 24
GLUT_BITMAP_HELVETICA_10	Fonte helvetica tamanho 10
GLUT_BITMAP_HELVETICA_12	Fonte helvetica tamanho 12
GLUT_BITMAP_HELVETICA_18	Fonte helvetica tamanho 18

Desenho de Texto Stroke

- Use a função **glutStrokeCharacter(void *font, int character)**
- Valores de fonte:
  - GLUT_STROKE_ROMAN
  - GLUT_STROKE_MONO_ROMAN

Exemplo de Desenho de Texto Bitmap

```
void desenhar(void)
{

  //Fonte bitmap
  glRasterPos2f(10,230);
  char *string = "texto bitmap";

  while(*string){
      glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, *string++);
  }

  //Fonte stroke
  glRasterPos2f(10,220);
  char *string2="texto stroke";
  while(*string2){
    glutStrokeCharacter( GLUT_STROKE_MONO_ROMAN,*string2++);
  }

  glFlush();
}
```

Exercício

- Pesquise como redimensionar e rotacionar texto do tipo stroke.