Computação Gráfica

Prof. Rodrigo Martins
rodrigo.martins@francomontoro.com.br

Esse material foi cedido pelo Prof. Jorge Cavalcanti da UNIVASF
(UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO)

Primitivas Gráficas

- São elementos básicos que compõem um desenho;
- Em OpenGL são definidas em um sistema de coordenadas bidimensionais por meio de vértices;
- A partir de primitivas simples é possível criar estruturas mais complexas.
- Objetos e cenas criados usando OpenGl consistem em um conjunto de primitivas gráficas simples que são combinadas para formar os modelos.
- OpenGL fornece ferramentas para desenhar pontos, linhas, "polilinhas" e polígonos, que são formados por um ou mais vértices.

Primitivas Gráficas

- OpenGL possui 10 primitivas geométricas
  - um tipo de ponto
  - três tipos de linhas
  - seis tipos de polígonos.
- Os vértices são definidos pelo comando Vertex

  glVertex2f(float x, float y); //vértice para um eixo 2D

  glVertex3d(double x,double y, double z); //vértice para um eixo 3D
- As primitivas precisam ser delimitadas através de Begin ... End conforme abaixo:

```
Begin (nome da primitiva);
... // aqui serão colocados comandos Vertex.
End ();
```

Primitivas Gráficas

- Ponto
  - a primitiva responsável em desenhar pontos na tela é **GL_POINTS**
  - O código no exemplo abaixo desenha 3 pontos na tela. Cada vértice torna-se um ponto

```
glBegin( GL_POINTS );
glVertex2f( xf, yf);
glVertex2f( xf, yf);
glVertex2f( xf, yf);
 glEnd();
```

  - O tamanho do ponto pode ser modificado através do comando glPointSize (GLint tamanho), bastando passar como parâmetro o tamanho do ponto.

Primitivas Gráficas

- Exemplo para desenhar três pontos pretos na tela:

```
glBegin(GL_POINTS);
        glColor3f(0.0f, 0.0f, 0.0f);
        glVertex2i(100, 50);
        glVertex2i(100, 130);
        glVertex2i(150, 130);
glEnd()
```

Primitivas Gráficas

- Linhas
  - **GL_LINES :** O terceiro ponto é ignorado pois a linha é formada por dois vértices.
    - Se houvesse um quarto vértice, uma nova linha entre o terceiro e quarto vértice seria exibida
  - **GL_LINE_STRIP:** cria linhas consecutivas, ligando o primeiro vértice com o segundo, o segundo com o terceiro e assim por diante
  - **GL_LINE_LOOP :** Funciona de maneira semelhante ao anterior, porém o último vértice é ligado a primeira, devido a isso o LOOP no seu nome.
  - O espessura de uma linha pode ser modificada através do comando glLineWidth (GLint espessura), bastando passar como parâmetro a espessura da linha

Primitivas Gráficas

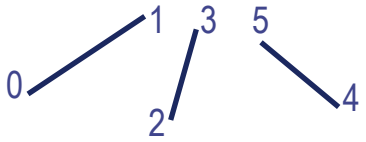
- Polígonos
  - Áreas formadas por várias linhas conectadas
  - Arestas do polígono não podem se cruzar
  - Devem ser áreas convexas
  - O número de segmentos do polígono não é restrito
  - OpenGL assume que todos os polígonos são simples
- Problema da superfície construída a partir de quadriláteros
  - Quadriláteros são polígonos não planares
  - Caso seja feita alguma transformação, podem deixar de ser polígonos simples
  - Para evitar que isto aconteça, é sempre bom utilizar triângulos para compor as superfícies, pois triângulos são sempre co-planares

Primitivas Gráficas

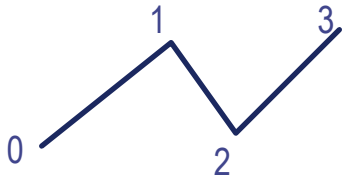
- Polígonos
  - **GL_TRIANGLES:** Desenha triângulos a cada 3 vértices fornecidos;
  - **GL_TRIANGLE_STRIP:** Uma série de triângulos conectados. Após o desenho do primeiro triângulo, cada vértice adicional forma um novo triângulo com dois últimos pontos fornecidos;
  - **GL_TRIANGLE_FAN:** Uma série de triângulos com um único vértice em comum. O vértice comum é o primeiro vértice fornecido;
  - **GL_QUADS:** Desenha um quadrilátero a cada 4 vértices fornecidos;
  - **GL_QUAD_STRIP:** Desenha uma série de quadriláteros. Após o primeiro, apenas mais 2 vértices precisam ser fornecidos para o desenho do segundo quadrilátero;
  - **GL_POLYGON:** Desenha polígonos **convexos** simples com um número arbitrário de vértices

Primitivas Gráficas

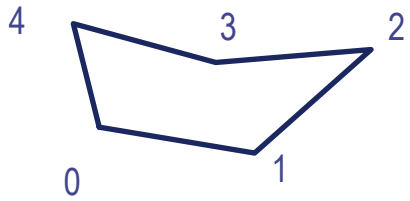
GL_POINTS



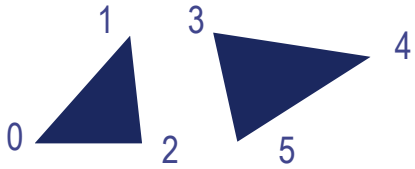
GL_LINES



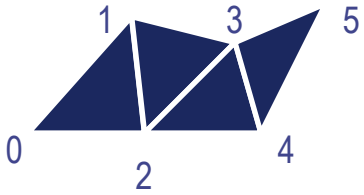
GL_LINE_STRIP



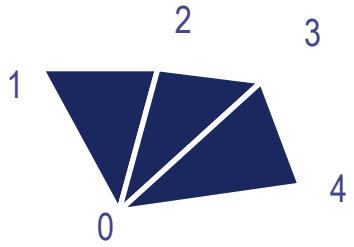
GL_LINE_LOOP



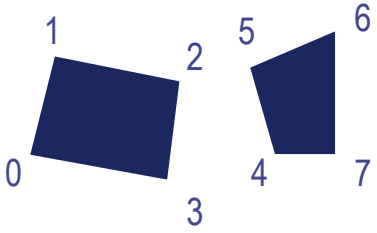
GL_TRIANGLES



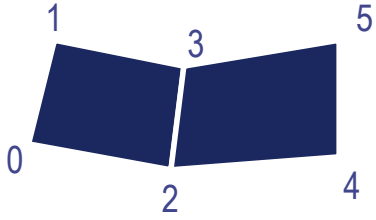
GL_TRIANGLE_STRIP



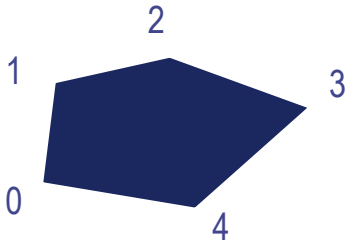
GL_TRIANGLE_FAN



GL_QUADS



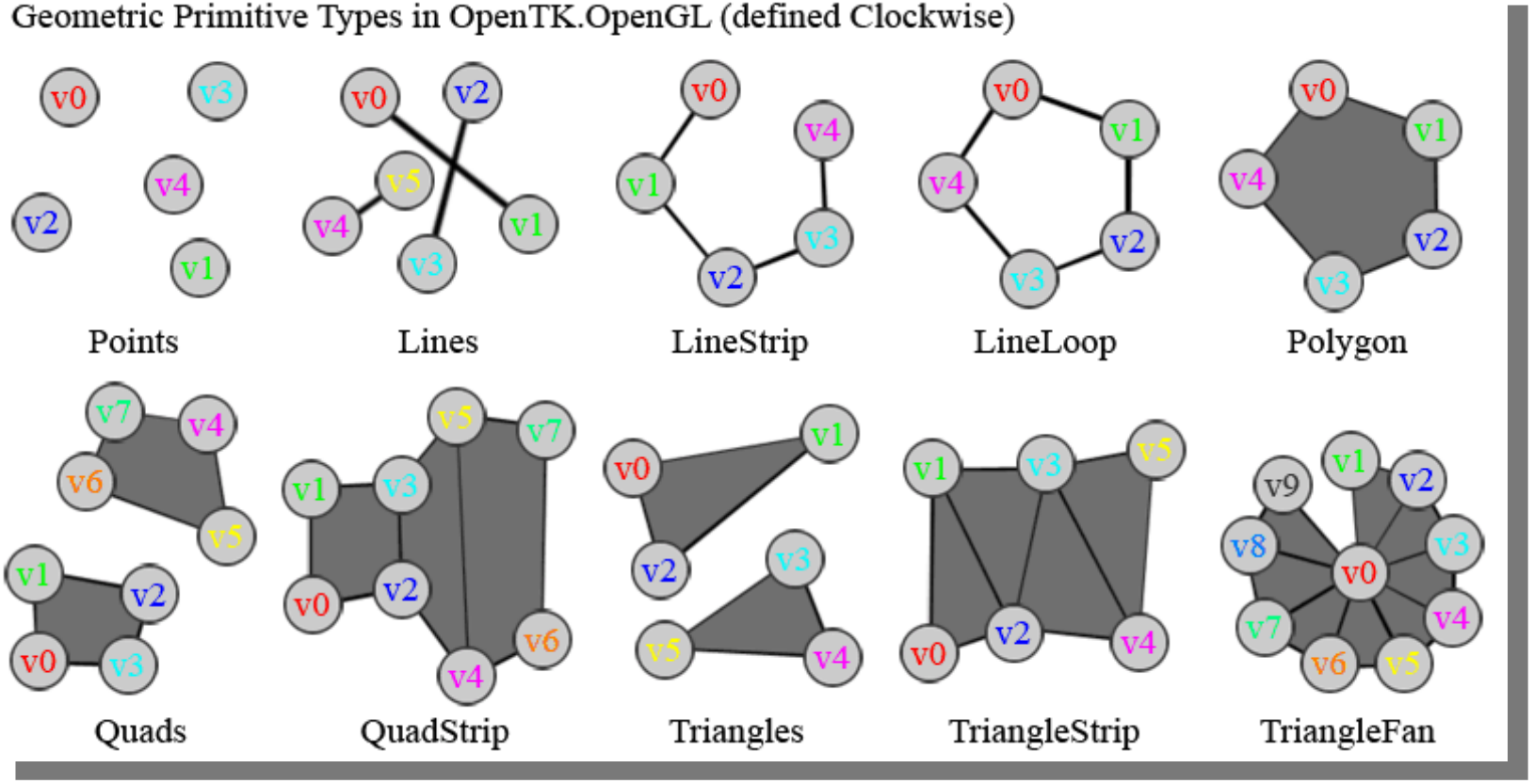
GL_QUAD_STRIP



GL_POLYGON

(convexo)

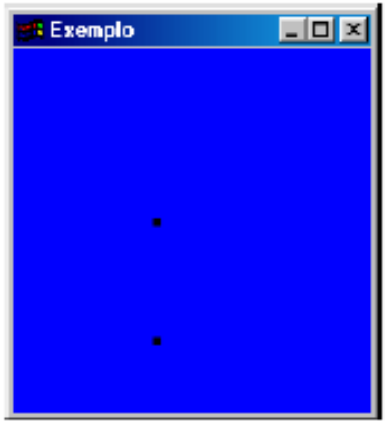
Primitivas Gráficas



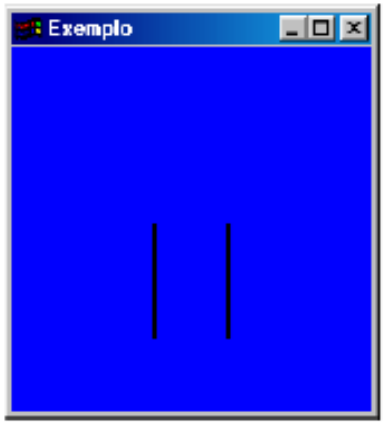
Primitivas Gráficas

- Exemplo:

```
glBegin(GL_POINTS);
        glColor3f(0.0f, 0.0f, 0.0f);
        glVertex2i(100, 50);
        glVertex2i(100, 130);
glEnd();
```



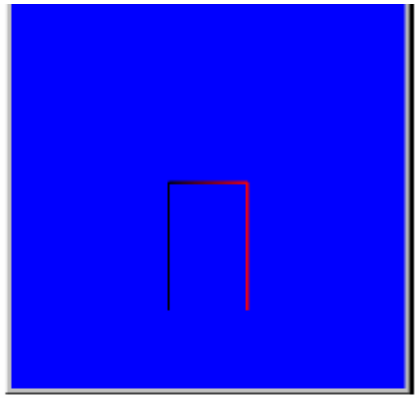

```
glBegin(GL_LINES);
        glColor3f(0.0f, 0.0f, 0.0f);
        glVertex2i(100, 50);
        glVertex2i(100, 130);
        glVertex2i(150, 130);
        glVertex2i(150, 50);
glEnd();
```

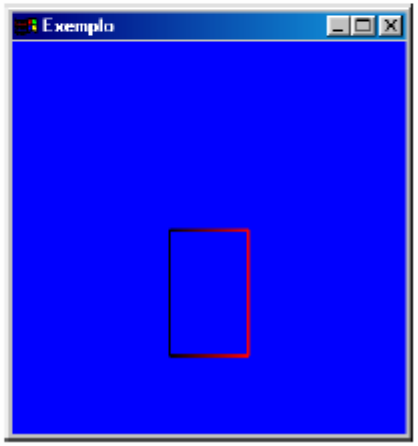


Primitivas Gráficas

- Exemplo:

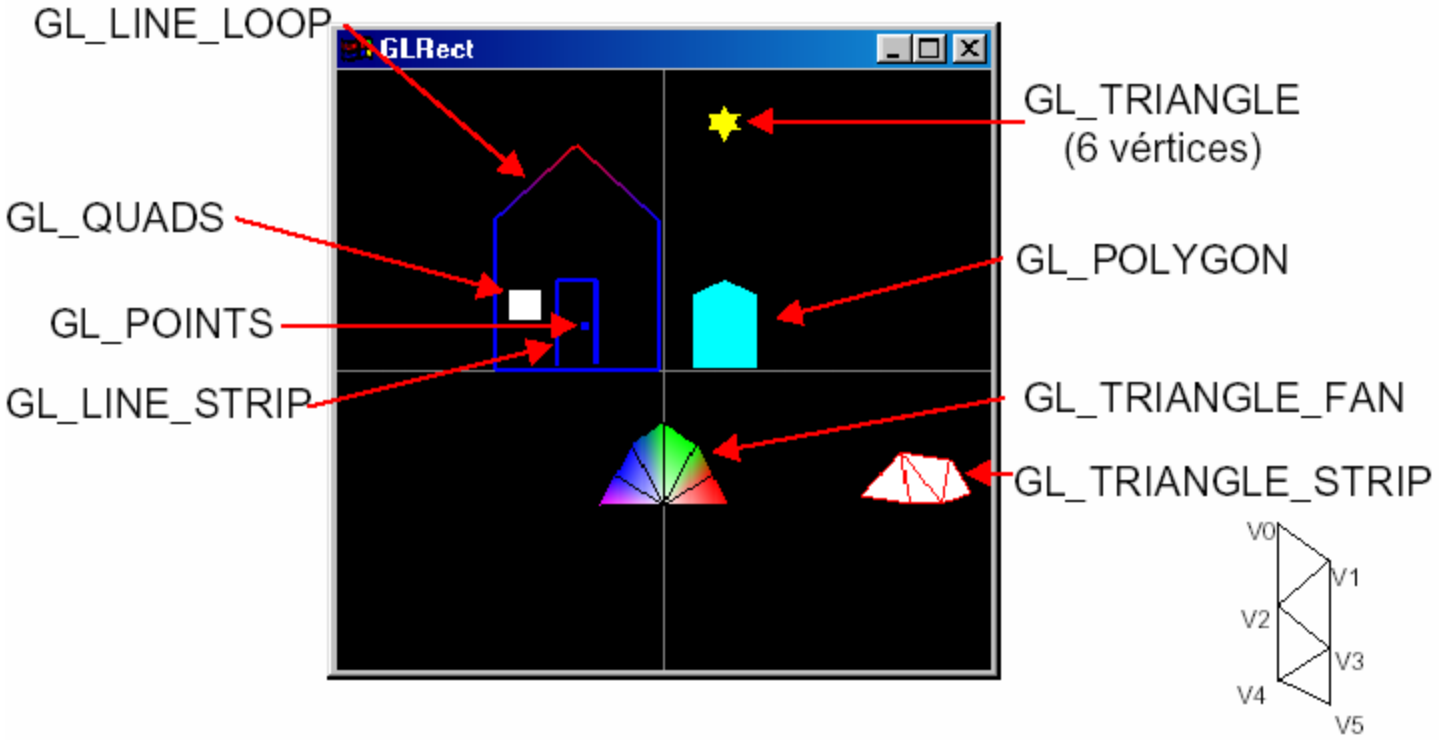
```
glBegin(GL_LINE_STRIP);
        glColor3f(0.0f, 0.0f, 0.0f);
        glVertex2i(100, 50);
        glVertex2i(100, 130);
        glColor3f(1.0f, 0.0f, 0.0f);
        glVertex2i(150, 130);
        glVertex2i(150, 50);
glEnd();
glBegin(GL_LINE_LOOP);
        glColor3f(0.0f, 0.0f, 0.0f);
        glVertex2i(100, 50);
        glVertex2i(100, 130);
        glColor3f(1.0f, 0.0f, 0.0f);
        glVertex2i(150, 130);
        glVertex2i(150, 50);
glEnd();
```





Primitivas Gráficas

- Outros exemplos



Programa 3

*main.cpp

```
1
2   #include <windows.h>
3   #include <gl/glut.h>
4
5   // Função callback chamada para fazer o desenho
6   void Desenha(void)
7   {
8       //glMatrixMode(GL_MODELVIEW);
9       //avisa a OpenGL que todas as futuras alterações, tais como operações de escala,
10      //rotação e translação, irão afetar os modelos da cena, ou em outras palavras, o que é desenhado.
11
12      glMatrixMode(GL_MODELVIEW);
13      //A função glLoadIdentity();
14      //faz com que a matriz corrente seja inicializada com a matriz identidade (nenhuma transaformação é acumulada)
15      glLoadIdentity();
16
17      // Limpa a janela de visualização com a cor de fundo especificada
18      glClear(GL_COLOR_BUFFER_BIT);
19
20      // Especifica que a cor corrente é vermelha
21      //          R     G     B
22      glColor3f(1.0f, 0.0f, 0.0f);
23
24     // antes da função glBegin(GL_QUADS) foram chamadas as funções glPointSize(6) e glLineWidth(6);
25     glPointSize(6); // aumenta a espessura do ponto
26     glLineWidth(1); // aumenta a espessura da linha
27
```

Programa 3

```
28      glBegin(GL_POINTS); // trocar por GL_LINES / GL_LINE_STRIP / GL_LINE_LOOP
29          glVertex2i(100,150);
30          glVertex2i(100,100);
31          // Especifica que a cor corrente é azul
32          glColor3f(0.0f, 0.0f, 1.0f);
33          glVertex2i(150,100);
34          glVertex2i(150,150);
35       glEnd();
36
37       // Executa os comandos OpenGL
38       glFlush();
39  }
40
41  // Inicializa parâmetros de rendering
42  void Inicializa (void)
43  {
44      // Define a cor de fundo da janela de visualização como preta
45      glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
46  }
47
```

Programa 3

```
48  // Função callback chamada quando o tamanho da janela é alterado
49  void AlteraTamanhoJanela(GLsizei w, GLsizei h)
50  {
51      // Evita a divisao por zero
52      if(h == 0){
53        h = 1;
54      }
55
56      // Especifica as dimensões da Viewport
57      glViewport(0, 0, w, h);
58
59      // Inicializa o sistema de coordenadas
60      glMatrixMode(GL_PROJECTION);
61      glLoadIdentity();
62
63      // Estabelece a janela de seleção (left, right, bottom, top)
64      if (w <= h){
65         gluOrtho2D (0.0f, 250.0f, 0.0f, 250.0f*h/w);
66      }
67      else{
68          gluOrtho2D (0.0f, 250.0f*w/h, 0.0f, 250.0f);
69      }
70  }
71
```

Programa 3

```
72  // Programa Principal
73  int main(int argc, char *argv[])
74  {
75      glutInit(&argc, argv);
76       glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
77       //inicialmente, os parâmetros passados para a função glutInitWindowSize foram alterados para (200,200)
78       //para diminuir o tamanho da janela GLUT;
79       glutInitWindowSize(200,200);
80       glutInitWindowPosition(10,10);
81       glutCreateWindow("Quadrado");
82       glutDisplayFunc(Desenha);
83       glutReshapeFunc(AlteraTamanhoJanela);
84       Inicializa();
85       glutMainLoop();
86  }
87
```

Primitivas Gráficas – Estilos de linha

Programa 4

main.cpp

```
1
2   #include <stdlib.h>
3   #include <GL/glut.h>
4
5   // Função callback de redesenho da janela de visualização
6   void Desenha(void)
7   {
8       // Limpa a janela de visualização com a cor
9       // de fundo definida previamente
10      glClear(GL_COLOR_BUFFER_BIT);
11
12      // Habilita a alteração do estilo das linhas
13      glEnable(GL_LINE_STIPPLE);
14
15      // Altera a cor do desenho para vermelho
16      glColor3f(1.0f, 0.0f, 0.0f);
17
18      // Altera a espessura da linha
19      glLineWidth(0.8);
20
21      // Altera o estilo da linha
22      glLineStipple(1, 0xFFFF);
23      glBegin(GL_LINES);
24          glVertex2f(-40.0f, 40.0f);
25          glVertex2f(40.0f, 40.0f);
26      glEnd();
27
```

Programa 4

```
28      // Altera a cor do desenho para verde
29      glColor3f(0.0f, 1.0f, 0.0f);
30
31      // Altera a espessura da linha
32      glLineWidth(1.6);
33
34      // Altera o estilo da linha
35      glLineStipple(4, 0x1F2F);
36      glBegin(GL_LINES);
37          glVertex2f(-40.0f, 30.0f);
38          glVertex2f(40.0f, 30.0f);
39      glEnd();
40
41      // Altera a cor do desenho para azul
42      glColor3f(0.0f, 0.0f, 1.0f);
43
44      // Altera a espessura da linha
45      glLineWidth(2.4);
46
47      // Altera o estilo da linha
48      glLineStipple(1, 0x01010);
49      glBegin(GL_LINES);
50          glVertex2f(-40.0f, 20.0f);
51          glVertex2f(40.0f, 20.0f);
52      glEnd();
53
```

Programa 4

```
54      // Altera a cor do desenho para preto
55      glColor3f(0.0f, 0.0f, 0.0f);
56
57      // Altera a espessura da linha
58      glLineWidth(3.2);
59
60      // Altera o estilo da linha
61      glLineStipple(2, 0xF020);
62      glBegin(GL_LINES);
63          glVertex2f(-40.0f, 10.0f);
64          glVertex2f(40.0f, 10.0f);
65      glEnd();
66
67      // Altera a cor do desenho para magenta
68      glColor3f(1.0f, 0.0f, 1.0f);
69
70      // Altera a espessura da linha
71      glLineWidth(4.0);
72
73      // Altera o estilo da linha
74      glLineStipple(3, 0x00FF);
75      glBegin(GL_LINES);
76          glVertex2f(-40.0f, 0.0f);
77          glVertex2f(40.0f, 0.0f);
78      glEnd();
79
```

Programa 4

```
79
80      // Altera a cor do desenho para preto
81      glColor3f(0.0f, 0.0f, 0.0f);
82
83      // Altera a espessura da linha
84      glLineWidth(4.8);
85
86      // Altera o estilo da linha
87      glLineStipple(3, 0x0F0F);
88      glBegin(GL_LINES);
89          glVertex2f(-40.0f, -10.0f);
90          glVertex2f(40.0f, -10.0f);
91      glEnd();
92
93      // Altera a cor do desenho para azul
94      glColor3f(0.0f, 0.0f, 1.0f);
95
96      // Altera a espessura da linha
97      glLineWidth(5.6);
98
99      // Altera o estilo da linha
100     glLineStipple(5, 0x5555);
101     glBegin(GL_LINES);
102         glVertex2f(-40.0f, -20.0f);
103         glVertex2f(40.0f, -20.0f);
104     glEnd();
105
```

Programa 4

```
106     // Altera a cor do desenho para verde
107     glColor3f(0.0f, 1.0f, 0.0f);
108
109     // Altera a espessura da linha
110     glLineWidth(6.4);
111
112     // Altera o estilo da linha
113     glLineStipple(1, 0x3F07);
114     glBegin(GL_LINES);
115         glVertex2f(-40.0f, -30.0f);
116         glVertex2f(40.0f, -30.0f);
117     glEnd();
118
119     // Altera a cor do desenho para vermelho
120     glColor3f(1.0f, 0.0f, 0.0f);
121
122     // Altera a espessura da linha
123     glLineWidth(7.2);
124
```

Programa 4

```
125         // Altera o estilo da linha
126         glLineStipple(4, 0x3F07);
127         glBegin(GL_LINES);
128             glVertex2f(-40.0f, -40.0f);
129             glVertex2f(40.0f, -40.0f);
130         glEnd();
131
132         // Executa os comandos OpenGL
133         glFlush();
134     }
135
136     // Função callback chamada quando o tamanho da janela é alterado
137     void AlteraTamanhoJanela(GLsizei w, GLsizei h)
138     {
139         // Evita a divisao por zero
140         if (h == 0)
141             h = 1;
142
143         // Especifica as dimensões da Viewport
144         glViewport(0, 0, w, h);
145
146         // Inicializa o sistema de coordenadas
147         glMatrixMode(GL_PROJECTION);
148         glLoadIdentity();
149
```

Programa 4

```
150         // Estabelece a janela de seleção (esquerda, direita, inferior,
151         // superior) mantendo a proporção com a janela de visualização
152         if (w <= h)
153             gluOrtho2D(-50.0f, 50.0f, -50.0f * h / w, 50.0f * h / w);
154         else
155             gluOrtho2D(-50.0f * w / h, 50.0f * w / h, -50.0f, 50.0f);
156     }
157
158     // Função callback chamada para gerenciar eventos de teclas
159     void Teclado (unsigned char key, int x, int y)
160     {
161         if (key == 27)
162             exit(0);
163     }
164
165     // Função responsável por inicializar parâmetros e variáveis
166     void Inicializa(void)
167     {
168         // Define a cor de fundo da janela de visualização como branca
169         glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
170     }
171
```

Programa 4

```
172  // Programa Principal
173  int main(int argc, char *argv[])
174  {
175      glutInit(&argc, argv);
176      // Define do modo de operação da GLUT
177      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
178      // Especifica a posição inicial da janela GLUT
179      glutInitWindowPosition(5, 5);
180      // Especifica o tamanho inicial em pixels da janela GLUT
181      glutInitWindowSize(450, 450);
182      // Cria a janela passando como argumento o título da mesma
183      glutCreateWindow("Desenho de Linhas");
184      // Registra a função callback de redesenho da janela de visualização
185      glutDisplayFunc(Desenha);
186      // Registra a função callback de redimensionamento da janela de visualização
187      glutReshapeFunc(AlteraTamanhoJanela);
188
189      // Registra a função callback para tratamento das teclas ASCII
190      glutKeyboardFunc (Teclado);
191
192      // Chama a função responsável por fazer as inicializações
193      Inicializa();
194
195      // Inicia o processamento e aguarda interações do usuário
196      glutMainLoop();
197
198      return 0;
199  }
```

Programa 5

*main.cpp

```
1
2
3   #include <stdlib.h>
4   #include <GL/glut.h>
5
6   float win, aspecto;
7   int largura, altura;
8
9   // Função que faz o desenho de uma casa composta de um quadrado e um triângulo
10  void DesenhaCasa()
11  {
12      // Altera a cor do desenho para azul
13      glColor3f(0.0f, 0.0f, 1.0f);
14      // Desenha a casa
15      glBegin(GL_QUADS);
16          glVertex2f(-15.0f,-15.0f);
17          glVertex2f(-15.0f,  5.0f);
18          glVertex2f( 15.0f,  5.0f);
19          glVertex2f( 15.0f,-15.0f);
20      glEnd();
21
```

Programa 5

```
22      // Altera a cor do desenho para branco
23      glColor3f(1.0f, 1.0f, 1.0f);
24      // Desenha a porta e a janela
25      glBegin(GL_QUADS);
26          glVertex2f(-4.0f,-14.5f);
27          glVertex2f(-4.0f,  0.0f);
28          glVertex2f( 4.0f,  0.0f);
29          glVertex2f( 4.0f,-14.5f);
30          glVertex2f( 7.0f,-5.0f);
31          glVertex2f( 7.0f,-1.0f);
32          glVertex2f(13.0f,-1.0f);
33          glVertex2f(13.0f,-5.0f);
34      glEnd();
35
36      // Altera a cor do desenho para azul
37      glColor3f(0.0f, 0.0f, 1.0f);
38      // Desenha as "linhas" da janela
39      glBegin(GL_LINES);
40          glVertex2f( 7.0f,-3.0f);
41          glVertex2f(13.0f,-3.0f);
42          glVertex2f(10.0f,-1.0f);
43          glVertex2f(10.0f,-5.0f);
44      glEnd();
45
```

Programa 5

```
46      // Altera a cor do desenho para vermelho
47      glColor3f(1.0f, 0.0f, 0.0f);
48      // Desenha o telhado
49      glBegin(GL_TRIANGLES);
50          glVertex2f(-15.0f, 5.0f);
51          glVertex2f(  0.0f,17.0f);
52          glVertex2f( 15.0f, 5.0f);
53      glEnd();
54  }
55
56  // Função que coloca uma linha ao redor da área da window
57  void FazMoldura()
58  {
59      glLineWidth(3);
60      glBegin(GL_LINE_LOOP);
61          glVertex2f(-win*aspecto,-win);
62          glVertex2f(-win*aspecto, win);
63          glVertex2f( win*aspecto, win);
64          glVertex2f( win*aspecto,-win);
65      glEnd();
66      glLineWidth(1);
67  }
68
```

Programa 5

```
69  // Função callback de redesenho da janela de visualização
70  void Desenha(void)
71  {
72      // Limpa a janela de visualização com a cor
73      // de fundo definida previamente
74      glClear(GL_COLOR_BUFFER_BIT);
75
76      // Define a Viewport 1 na metade esquerda da janela
77      glViewport(0, 0, largura, altura);
78      // Desenha a casa na Viewport 1
79      DesenhaCasa();
80      FazMoldura();
81
82      // Define a Viewport 2 na metade direita da janela
83      glViewport(largura, 0, largura, altura);
84      // Desenha a casa na Viewport 2
85      DesenhaCasa();
86      FazMoldura();
87
88      // Executa os comandos OpenGL
89      glFlush();
90  }
91
```

Programa 5

```
92  // Função callback chamada quando o tamanho da janela é alterado
93  void AlteraTamanhoJanela(GLsizei w, GLsizei h)
94  {
95      // Evita a divisao por zero
96      if(h == 0) h = 1;
97
98      // Atualiza as variáveis
99      largura = w/2;
100     altura = h;
101
102     aspecto = (float) largura/altura;
103     // Inicializa o sistema de coordenadas
104     glMatrixMode(GL_PROJECTION);
105     glLoadIdentity();
106
107     // Estabelece a janela de seleção (esquerda, direita, inferior,
108     // superior) mantendo a proporção com a janela de visualização
109     gluOrtho2D (-win*aspecto, win*aspecto, -win, win);
110 }
111
112 // Função callback chamada para gerenciar eventos de teclas
113 void Teclado (unsigned char key, int x, int y)
114 {
115     if (key == 27)
116         exit(0);
117 }
118
```

Programa 5

```
119  // Função responsável por inicializar parâmetros e variáveis
120  void Inicializa (void)
121  {
122      // Define a cor de fundo da janela de visualização como branca
123      glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
124      win = 50.0f;
125  }
126
127  // Programa Principal
128  int main(int argc, char *argv[])
129  {
130      glutInit(&argc, argv);
131      // Define do modo de operação da GLUT
132      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
133
134      // Especifica a posição inicial da janela GLUT
135      glutInitWindowPosition(5,5);
136
137      // Especifica o tamanho inicial em pixels da janela GLUT
138      glutInitWindowSize(450,450);
139
140      // Cria a janela passando como argumento o título da mesma
141      glutCreateWindow("Exemplo com duas viewports");
142
143      // Registra a função callback de redesenho da janela de visualização
144      glutDisplayFunc(Desenha);
145
```

Programa 5

```
146     // Registra a função callback de redimensionamento da janela de visualização
147     glutReshapeFunc(AlteraTamanhoJanela);
148
149     // Registra a função callback para tratamento das teclas ASCII
150     glutKeyboardFunc (Teclado);
151
152     // Chama a função responsável por fazer as inicializações
153     Inicializa();
154
155     // Inicia o processamento e aguarda interações do usuário
156     glutMainLoop();
157
158     return 0;
159 }
```

Referências desta aula

- AZEVEDO, Eduardo; CONCI, Aura. 2007. Computação Gráfica: Teoria e Prática. Elsevier, Vol. 2, 2007.

- Aula montada com base no material do Prof. Jorge Cavalcanti - UNIVASF.