Módulo 12

Sistemas Gráficos e Interação

Instituto Superior de Engenharia do Porto

Filipe Pacheco

ffp@isep.ipp.pt

Seleção & feedback

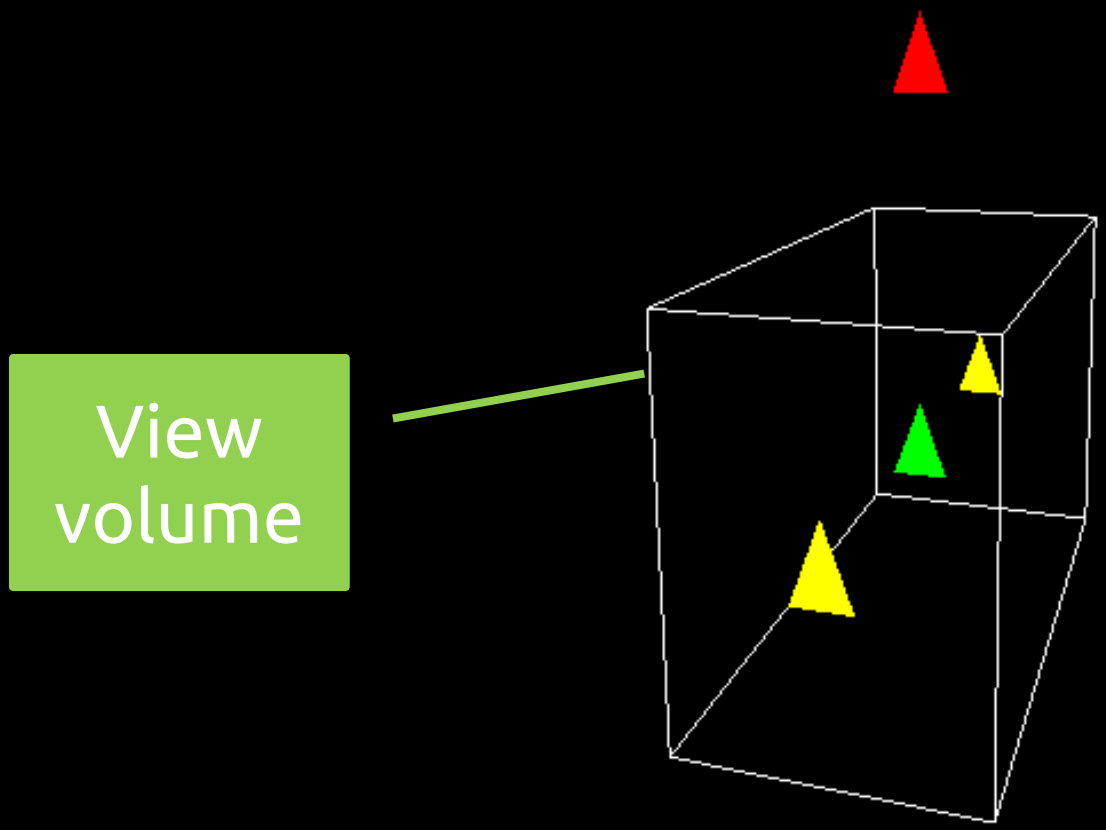
Conteúdo

- Modos do OpenGL
- Seleção
- Picking
- Feedback

Modos do OpenGL

- glRenderMode(***mode***)
  - GL_RENDER
    - Modo normal de funcionamento: desenho das primitivas no ecrã
  - GL_SELECTION
    - Modo de **seleção**: não desenha no ecrã mas devolve informação (nome simbólico) sobre os objetos que seriam desenhados
    - Modo ***picking***: idêntico mas restringe área de informação, usando a posição de um dispositivo de input (normalmente o rato)
  - GL_FEEDBACK
    - Modo ***feedback***: não desenha no ecrã mas devolve informação sobre os elementos gráficos que seriam desenhados no ecrã (vértices, cores, ...)

Exemplo seleção



Exemplo seleção

```
void display(void)
{
   glClearColor (0.0, 0.0, 0.0, 0.0);
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

   // desenhar cena normal
   drawScene();

   // "desenhar" cena em modo seleção
   selectObjects();

   glFlush();
}
```

Exemplo seleção

```
void selectObjects(void) {
   GLuint selectBuf[BUFSIZE];
   glSelectBuffer(BUFSIZE, selectBuf);
   glRenderMode(GL_SELECT);
   glInitNames();
   glPushName(0); //colocar nome inicial na stack - 0
    // definir projeção, visualização e "desenhar"
   glPushMatrix();
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity ();
   glOrtho(0.0, 5.0, 0.0, 5.0, 0.0, 10.0);
   glMatrixMode (GL_MODELVIEW);
   glLoadIdentity ();
   drawTriangles();
   glPopMatrix();
   glFlush();
   GLint hits = glRenderMode(GL_RENDER);
   processHits (hits, selectBuf);
}
```

Exemplo seleção

```
void drawTriangles(void)
{
 /* green triangle */
    glColor3f (0.0, 1.0, 0.0);
    glLoadName(1);
    drawTriangle (2.0, 2.0, 3.0, 2.0, 2.5, 3.0, -5.0);

 /* red triangle */
    glColor3f (1.0, 0.0, 0.0);
    glLoadName(2);
    drawTriangle (2.0, 7.0, 3.0, 7.0, 2.5, 8.0, -5.0);

 /* yellow triangles */
    glColor3f (1.0, 1.0, 0.0);
    glLoadName(3);
    drawTriangle (2.0, 2.0, 3.0, 2.0, 2.5, 3.0, 0.0);
    drawTriangle (2.0, 2.0, 3.0, 2.0, 2.5, 3.0, -10.0);
}
```

Instruções

- glSelectBuffer(size, buffer)
  - definir **antes** de entrar no modo seleção
- glInitNames()
  - invocar **antes** de desenhar a cena
- glPushName(name)/glPopName()
  - Inserir/retirar um nome simbólico da pilha
- glLoadName(name)
  - definir o nome simbólico no topo da pilha

Seleção

- Passos a seguir:
  1. Inicializar *buffer* de retorno
  2. Entrar modo de seleção
  3. Inicializar *stack* de nomes simbólicos
  4. Definir volume de visualização
  5. “Desenhar” a cena definindo o nome simbólico dos objetos
  6. Sair do modo seleção e processar os registos do *buffer* de retorno

Processar resultados

- Ao sair do modo seleção (invocando glRenderMode) recebe-se informação sobre seleção
  - *Array* de tamanho variável com registos de tamanho variável
  - *Hit* record = #names, z1, z2, (names)*

#names	z1	z2	names
1	0.3	0.43	27
2	0.55	0.62	27 35
1	0.23	0.23	29

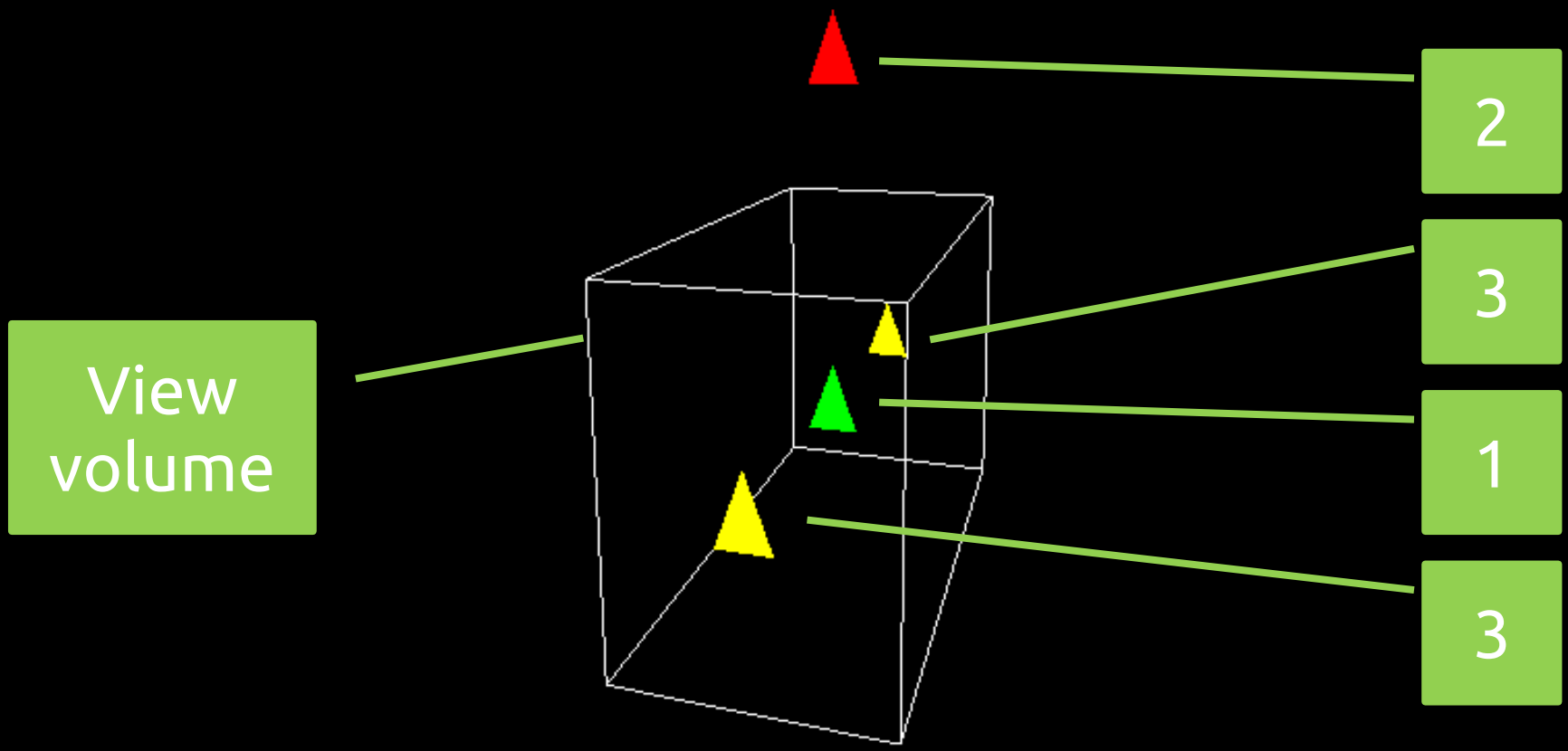
#n	z1	z2	name	#n	z1	z2	name1	name2	#n	z1	z2	name
1	3	43	27	2	55	62	27	35	1	23	23	29

Exemplo seleção

```
void processHits(GLint hits, GLuint buffer[])
{
   int i, j, names;
   GLuint *ptr;

   printf ("hits = %d\n", hits);
   ptr = (GLuint *) buffer;
   for (i = 0; i < hits; i++) { /*  for each hit  */
      names = (int) *ptr;
      printf (" number of names for hit = %d\n", names); ptr++;
      printf(" z1 is %g;", (float) *ptr/UINT_MAX); ptr++;
      printf(" z2 is %g\n", (float) *ptr/UINT_MAX); ptr++;
      printf ("   the name is ");
      for (j = 0; j < names; j++) { /*  for each name */
         printf ("%d ", *ptr); ptr++;
      }
      printf ("\n");
   }
}
```

Demo – select



Picking

- Passos a seguir:
  1. Inicializar *buffer* de retorno
  2. Entrar modo de seleção
  3. Inicializar *stack* de nomes simbólicos
  4. **Definir matriz de sensibilidade baseada na posição do dispositivo de *input***
  5. Definir volume de visualização
  6. "Desenhar" a cena incluindo o nome simbólico dos objetos
  7. Sair do modo seleção e processar os registos do *buffer* de retorno

Exemplo picking

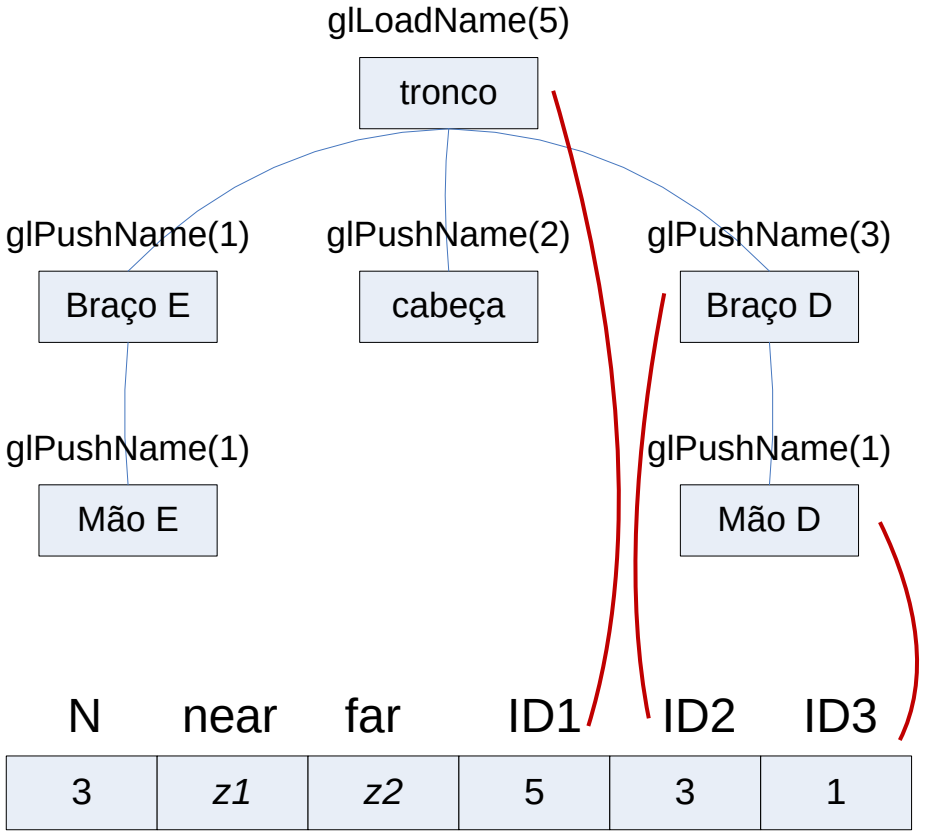
- Entrar no modo seleção através de evento do rato
  - glutPassiveMotionFunc
  - glutMouseFunc
- Definir a matriz de sensibilidade – GL_PROJECTION

```
glMatrixMode(GL_PROJECTION);
glPushMatrix();
glLoadIdentity();
GLint viewport[4];
glGetIntegerv(GL_VIEWPORT, viewport);
// criar região de picking de 5x5 pixel junto ao ponteiro
gluPickMatrix((GLdouble)x, (GLdouble)(viewport[3]-y),
    5.0, 5.0, viewport);
// definir matriz de projeção
...
```

Múltiplos níveis de nomes

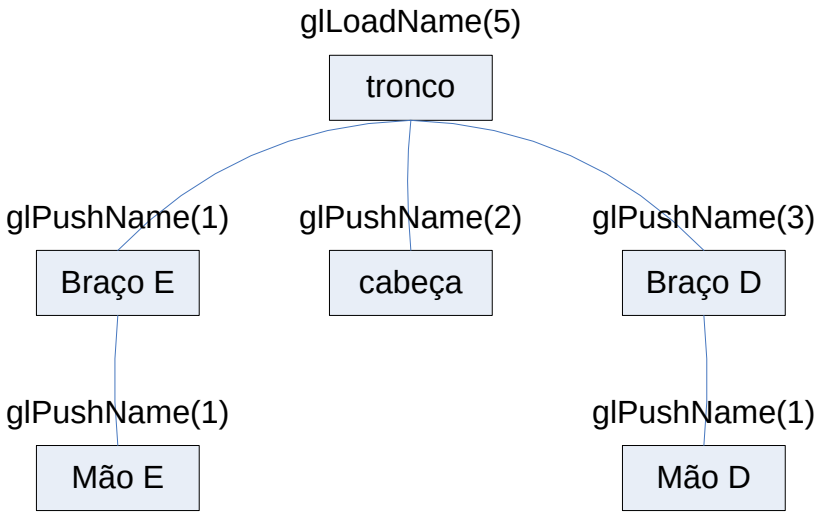
- Para permitir trabalhar com objetos hierárquicos é possível definir uma stack de nomes e não apenas um nome para o objecto

- Se utilizador selecionasse o objecto "mão D", o hit record seria:



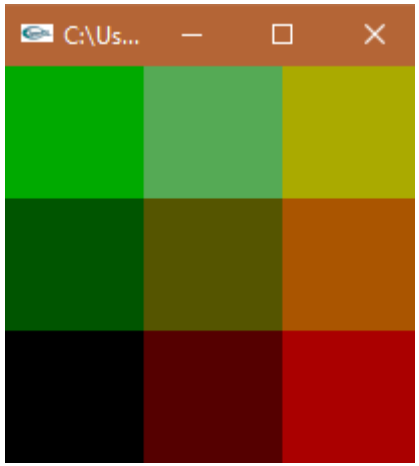
Exemplo

```
void drawRobot()
{
    glLoadName(5);
    drawTronco();
    // braço D
    glPushMatrix();
        glPushName(3);
        drawBraco();
            glPushName(1);
            drawMao();
            glPopName();
        glPopName();
    glPopMatrix();
    // cabeça
    glPushMatrix();
        glPushName(2);
        drawCabeca();
        glPopName();
    glPopMatrix();
    // braço E
    ...
}
```



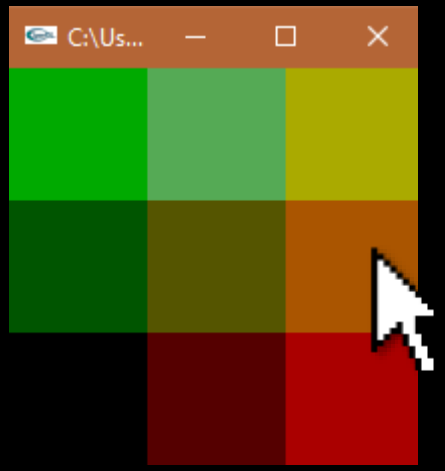
Exemplo

```
void drawSquares(GLenum mode)
{
   GLuint i, j;
   for (i = 0; i < 3; i++)
    {
      if (mode == GL_SELECT)
         glLoadName(i);
      for(j = 0; j < 3; j ++)
       {
         if (mode == GL_SELECT)
            glPushName(j);
         glColor3f(i/3.0, j/3.0, board[i][j]/3.0);
         glRecti(i, j, i+1, j+1);
         if (mode == GL_SELECT)
            glPopName();
      }
   }
}
```



Demo - picksquare

```
hits = 1
 number of names for this hit = 2
  z1 is 0.999999; z2 is 0.999999
   names are 1 2
```




```
hits = 1
 number of names for this hit = 2
  z1 is 0.999999; z2 is 0.999999
   names are 0 1
```

feedback

- O que faz?
  - Saber o que seria desenhado
    - Informação vetorial
- Para que serve?
  - Gerar comandos de impressora
  - Gravar em ficheiro num formato vetorial, ex., DXF, *windows metafile*

feedback

- Passos a executar:
  1. Definir *buffer* de retorno
  2. Entrar no modo de feedback
  3. “Desenhar” os objetos
  4. Sair do modo de feedback
  5. Processar informação de retorno

Exemplo

```
void display(void)
{
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glOrtho(0.0, 100.0, 0.0, 100.0, 0.0, 1.0);

   glClearColor(0.0, 0.0, 0.0, 0.0);
   glClear(GL_COLOR_BUFFER_BIT);
   drawGeometry(GL_RENDER);

   GLfloat feedBuffer[1024];
   glFeedbackBuffer (1024, GL_3D_COLOR, feedBuffer);
   glRenderMode(GL_FEEDBACK);
   drawGeometry(GL_FEEDBACK);

   GLint size = glRenderMode (GL_RENDER);
   printBuffer (size, feedBuffer);
}
```

Exemplo

```
void drawGeometry(GLenum mode)
{
   glBegin(GL_LINE_STRIP);
      glNormal3f(0.0, 0.0, 1.0);
      glVertex3f(30.0, 30.0, 0.0);
      glVertex3f(50.0, 60.0, 0.0);
      glVertex3f(70.0, 40.0, 0.0);
   glEnd();
   if (mode == GL_FEEDBACK)
      glPassThrough(1.0);
   glBegin(GL_POINTS);
       /* will be clipped  */
      glVertex3f(-100.0, -100.0, -100.0);
   glEnd();
   if (mode == GL_FEEDBACK)
      glPassThrough(2.0);
   glBegin(GL_POINTS);
      glNormal3f(0.0, 0.0, 1.0);
      glVertex3f(50.0, 50.0, 0.0);
   glEnd();
}
```

Informação de retorno

- glFeedbackBuffer(*size*, *type*, *buffer*)

Type	Coord.	Cor	Texturas	Nr. Valores
GL_2D	x, y			2
GL_3D	x, y, z			3
GL_3D_COLOR	x, y, z	k		3 + k
GL_3D_COLOR_TEXTURE	x, y, z	k	4	7 + k
GL_4D_COLOR_TEXTURE	x, y, z, w	k	4	8 + k

k=1 (Color Index) k=4 (RGBA)

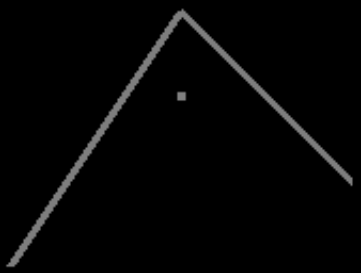
Informação de retorno

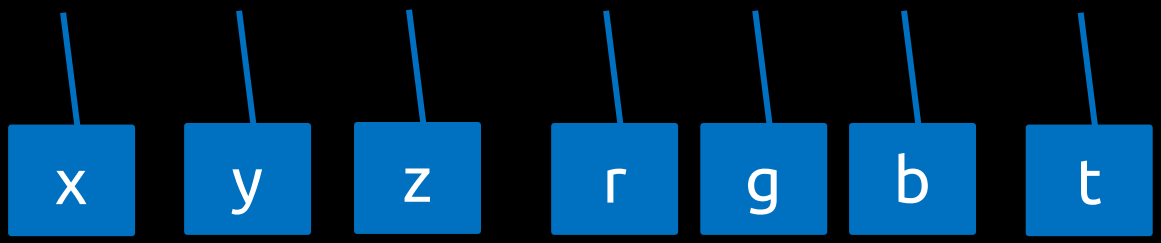
- O *buffer* de retorno é preenchido com informação de tamanho variável
  - Código da primitiva, dados
  - Vértices são coordenadas de janela

Primitive	Type Code	Associated Data
Point	GL_POINT_TOKEN	vertex
Line	GL_LINE_TOKEN or GL_LINE_RESET_TOKEN	vertex vertex
Polygon	GL_POLYGON_TOKEN	n vertex vertex ... vertex
Bitmap	GL_BITMAP_TOKEN	vertex
Pixel Rectangle	GL_DRAW_PIXEL_TOKEN or GL_COPY_PIXEL_TOKEN	vertex
Pass-through	GL_PASS_THROUGH_TOKEN	a floating-point number

Demo

```
GL_LINE_RESET_TOKEN
  36.00 30.00 0.00 0.84 0.84 0.84 1.00
  60.00 60.00 0.00 0.84 0.84 0.84 1.00
GL_LINE_TOKEN
  60.00 60.00 0.00 0.84 0.84 0.84 1.00
  84.00 40.00 0.00 0.84 0.84 0.84 1.00
GL_PASS_THROUGH_TOKEN
  1.00
GL_PASS_THROUGH_TOKEN
  2.00
GL_POINT_TOKEN
  60.00 50.00 0.00 0.84 0.84 0.84 1.00
```





Exportar para PS e WMF

- **Podem ver exemplos em:**
  - http://www.codeproject.com/opengl/glexport.asp
  - http://www.geuz.org/gl2ps/