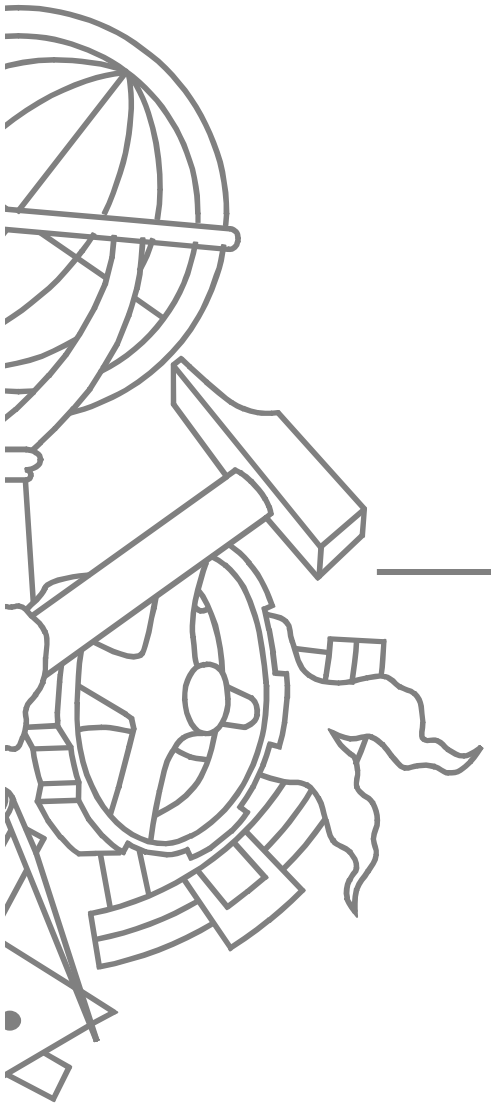


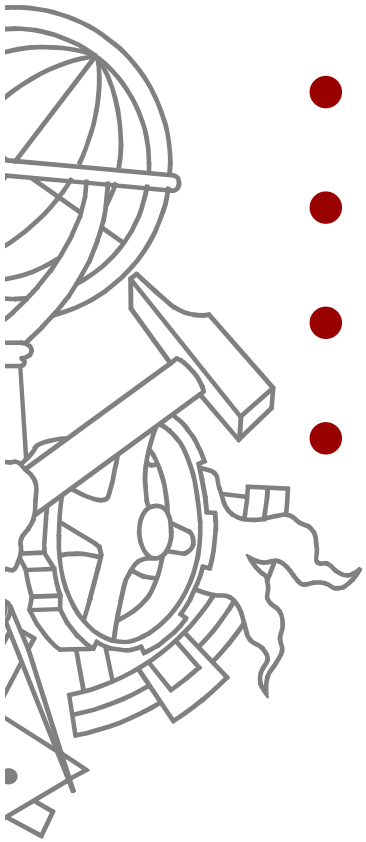
Selecção & *feedback*

Aula 8

Sistemas Gráficos e Interactivos
Instituto Superior de Engenharia do Porto

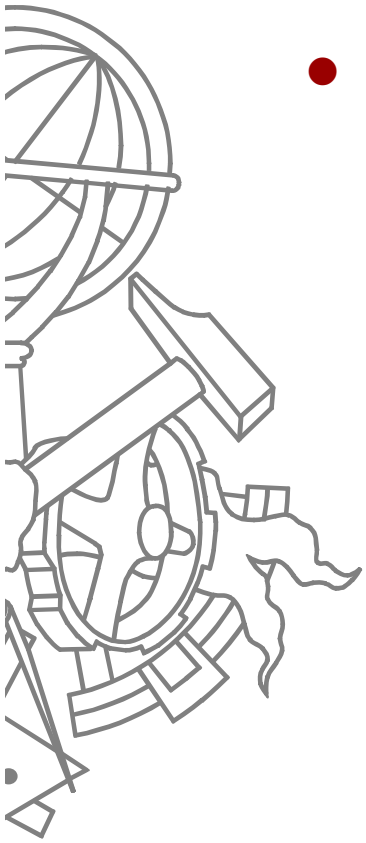
Paulo Gandra de Sousa
psousa@dei.isep.ipp.pt

Conteúdo



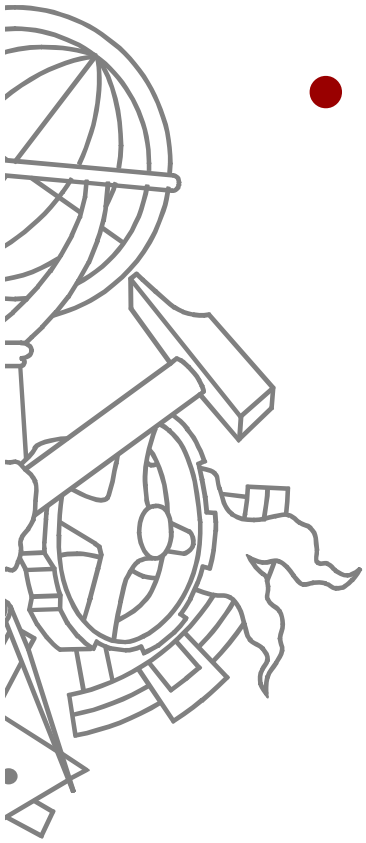
- Modos do opengl
- Selecção
- Picking
- feedback

Modos do OpenGL



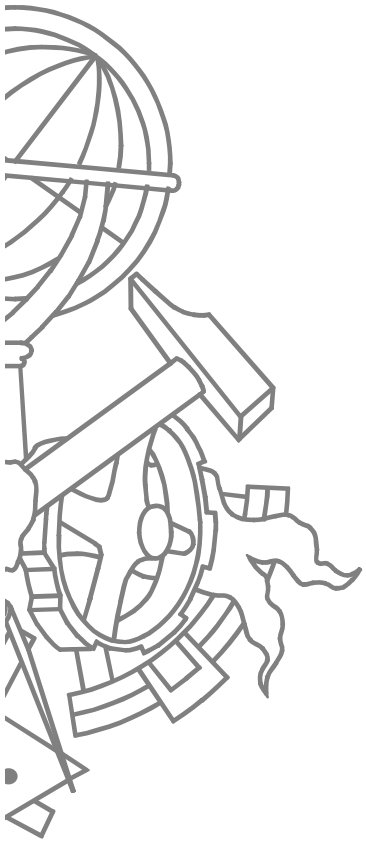
- glRenderMode(*mode*)
  - GL_RENDER
    - Modo normal de funcionamento: desenho das primitivas no ecrã
  - GL_SELECTION
    - Modo de selecção: não desenha no ecrã mas devolve informação (nome simbólico) sobre os objectos que seriam desenhados
    - Modo picking: idêntico mas com base na posição de dispositivo de input
  - GL_FEEDBACK
    - Modo feedback: não desenha no ecrã mas devolve informação sobre os elementos gráficos que seriam desenhados no ecrã (vértices, cores, ...)

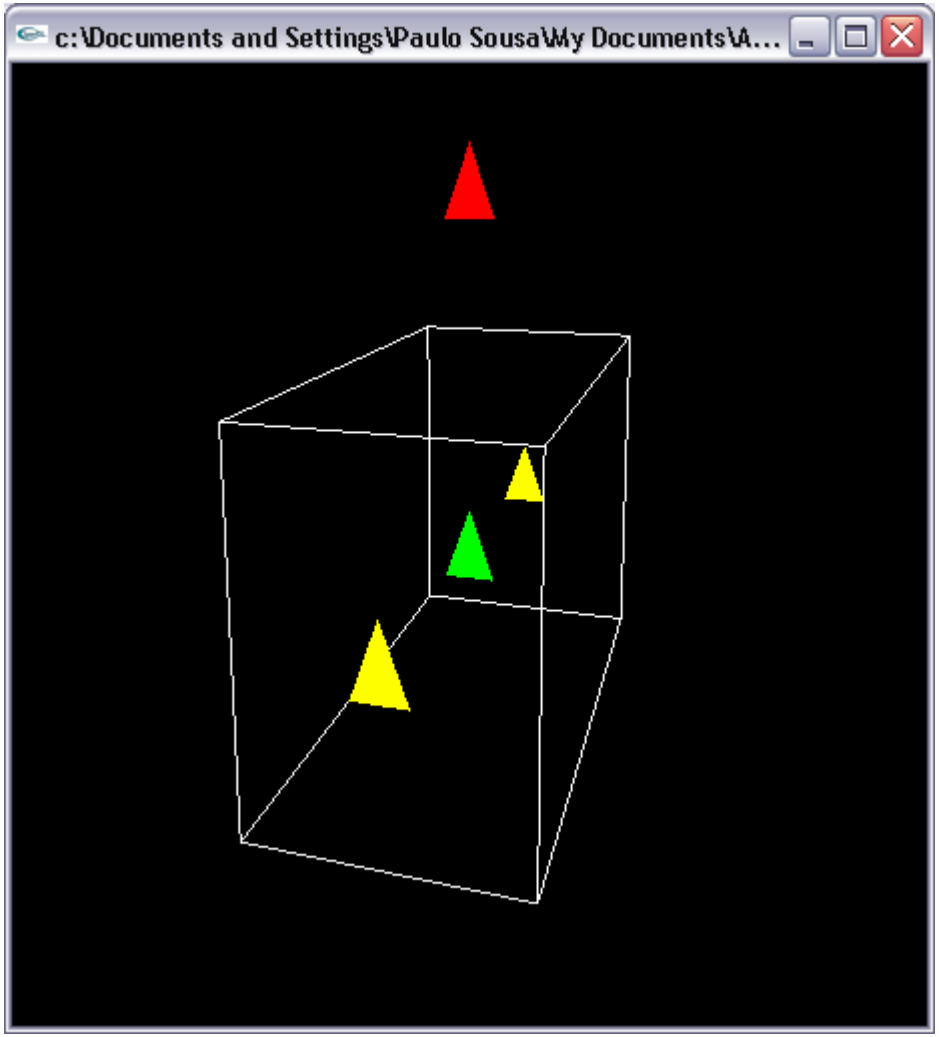
Selecção



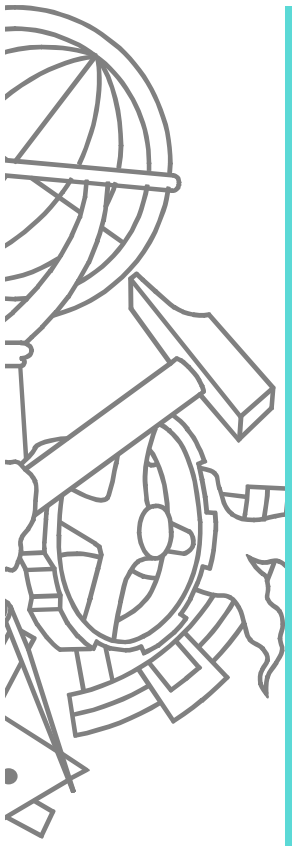
- Passos a seguir:
  1. Inicializar buffer de retorno
  2. Entrar modo de selecção
  3. Inicializar stack de nomes simbolicos
  4. Definir volume de visualização
  5. “Desenhar” a cena contendo o nome simbolico dos objectos
  6. Sair do modo selecção e processar os registos do buffer de retorno

Exemplo selecção



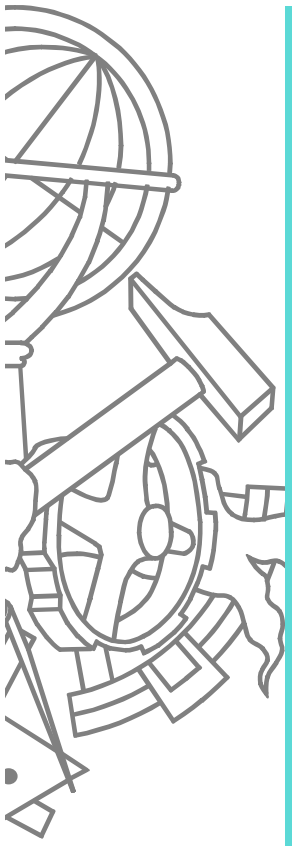


Exemplo selecção



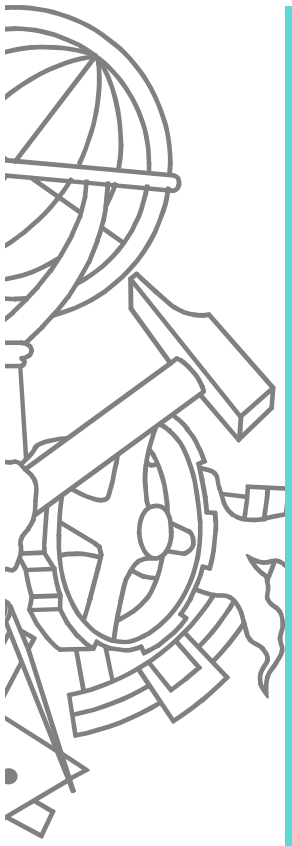
```
void display(void)
{
   glClearColor (0.0, 0.0, 0.0, 0.0);
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

   // desenhar cena normal
   drawScene();

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

   glFlush();
}
```

Exemplo selecção



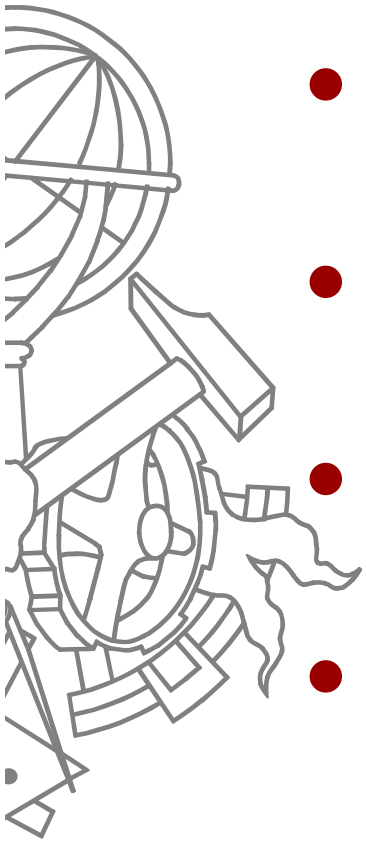
```
void selectObjects(void) {
   GLuint selectBuf[BUFSIZE];
   glSelectBuffer(BUFSIZE, selectBuf);
   glRenderMode(GL_SELECT);
   glInitNames();
   glPushName(0); //colocar nome inicial na stack - 0

   // definir projecção, 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 selecção



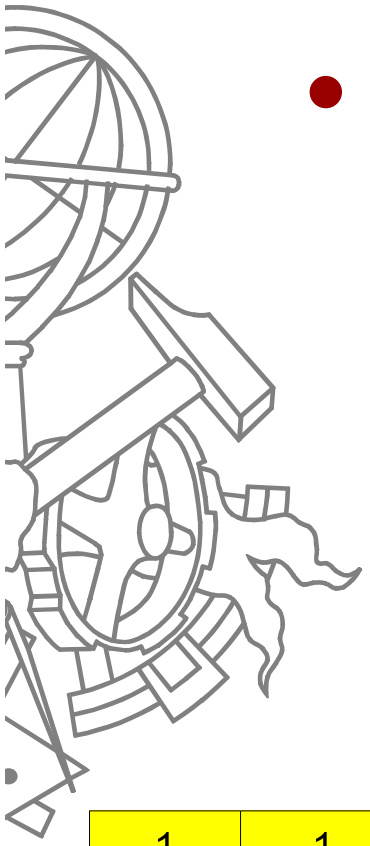
```
void drawTriangles(void)
{
/*green triangle*/
   glColor3f (0.0, 1.0, 0.0);
   glLoadName(1);
   drawTriangle (2.0, 2.0, 3.0, 2.0, 2.5, 3.0, -5.0);

/*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
  - Definir **antes** de entrar no modo selecção
- glInitNames
  - Invocar **antes** de desenhar os objectos
- glPushName
  - Inserir um nome simbólico
- glLoadName
  - Definir o nome simbólico do(s) proximo(s) objecto(s)

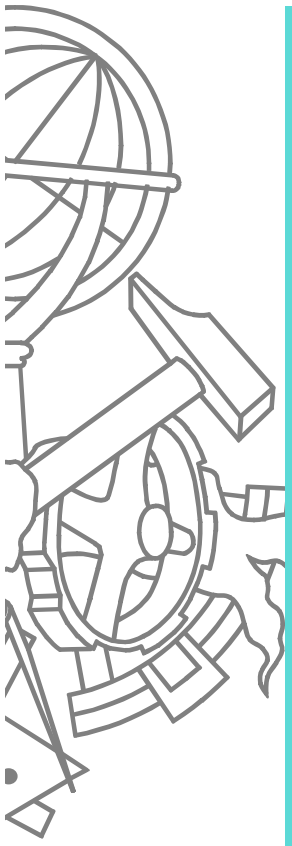
Processar resultados



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

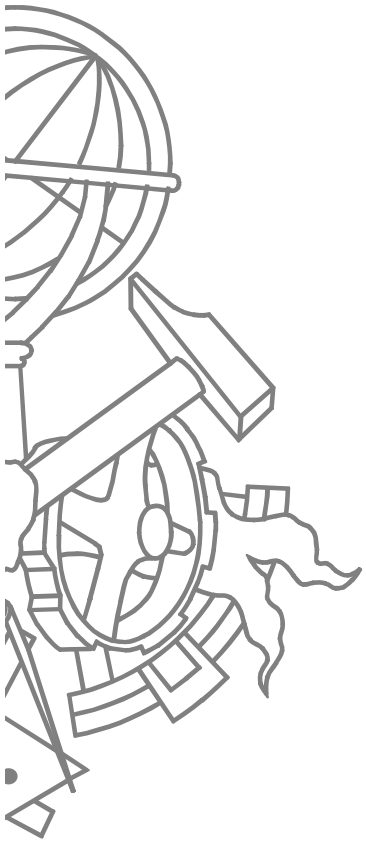
1	-1	3	27	2	-1	3	27	35	0	-1	3

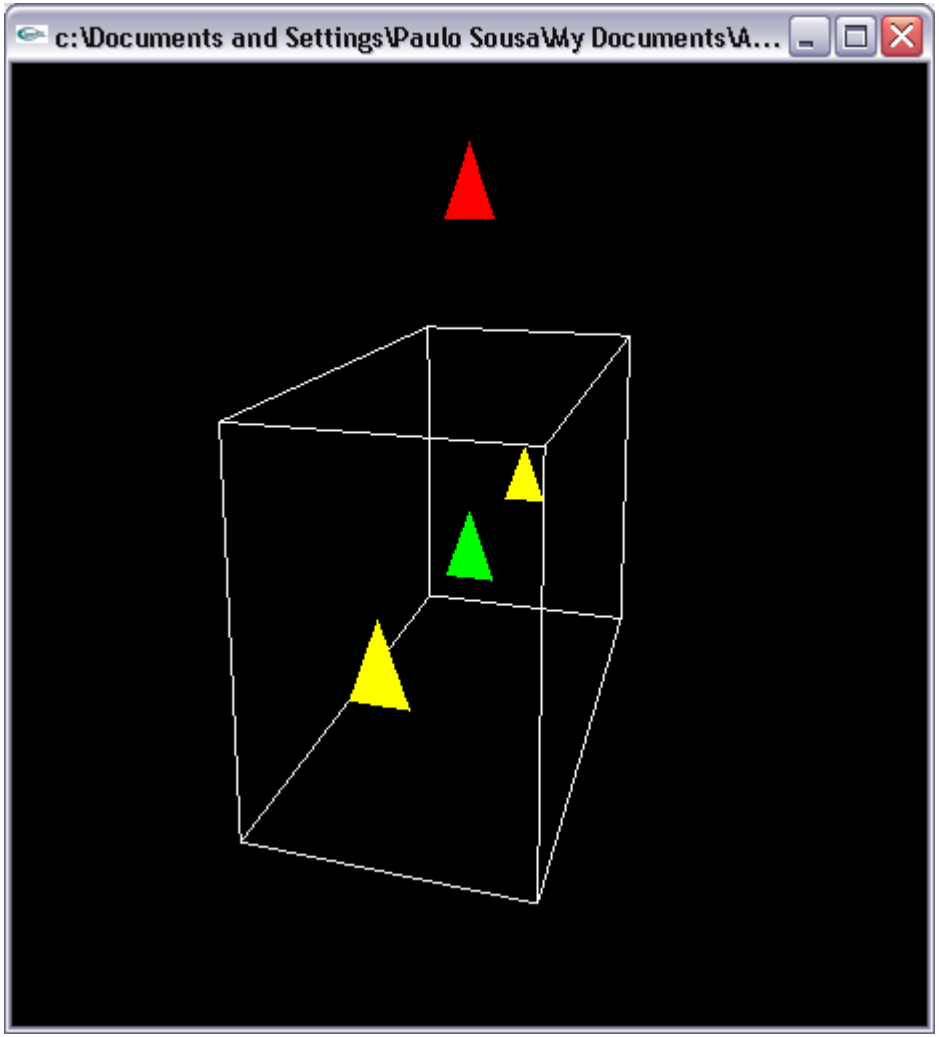
Exemplo selecção



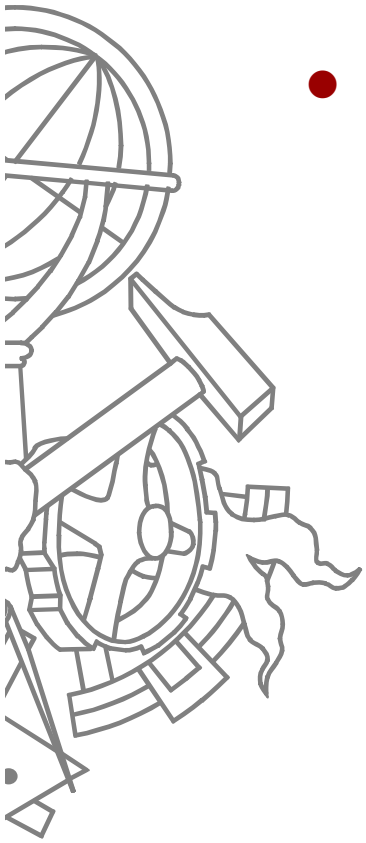
```
void processHits(GLint hits, GLuint buffer[])
{
   int i, j, names;
   GLuint *ptr;

   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/0x7fffffff); ptr++;
      printf(" z2 is %g\n", (float) *ptr/0x7fffffff); ptr++;
      printf ("   the name is ");
      for (j = 0; j < names; j++) {     /*  for each name */
         printf ("%d ", *ptr); ptr++;
      }
      printf ("\n");
   }
}
```

Demo - select



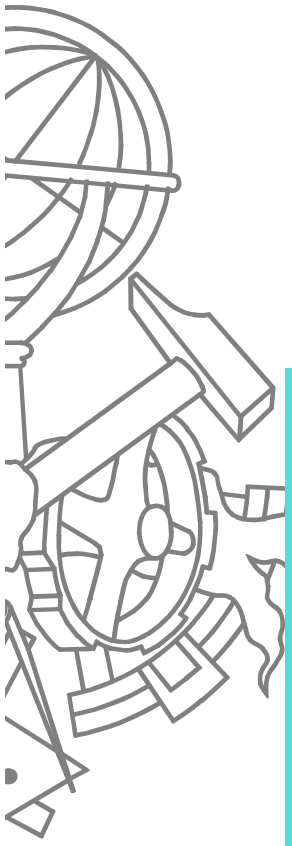


Picking



- Passos a seguir:
  1. Inicializar buffer de retorno
  2. Entrar modo de selecção
  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 contendo o nome simbólico dos objectos
  7. Sair do modo selecção e processar os registos do buffer de retorno

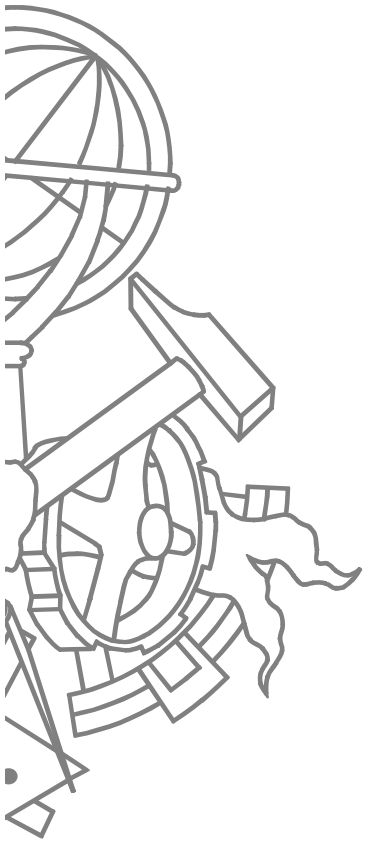
Exemplo picking

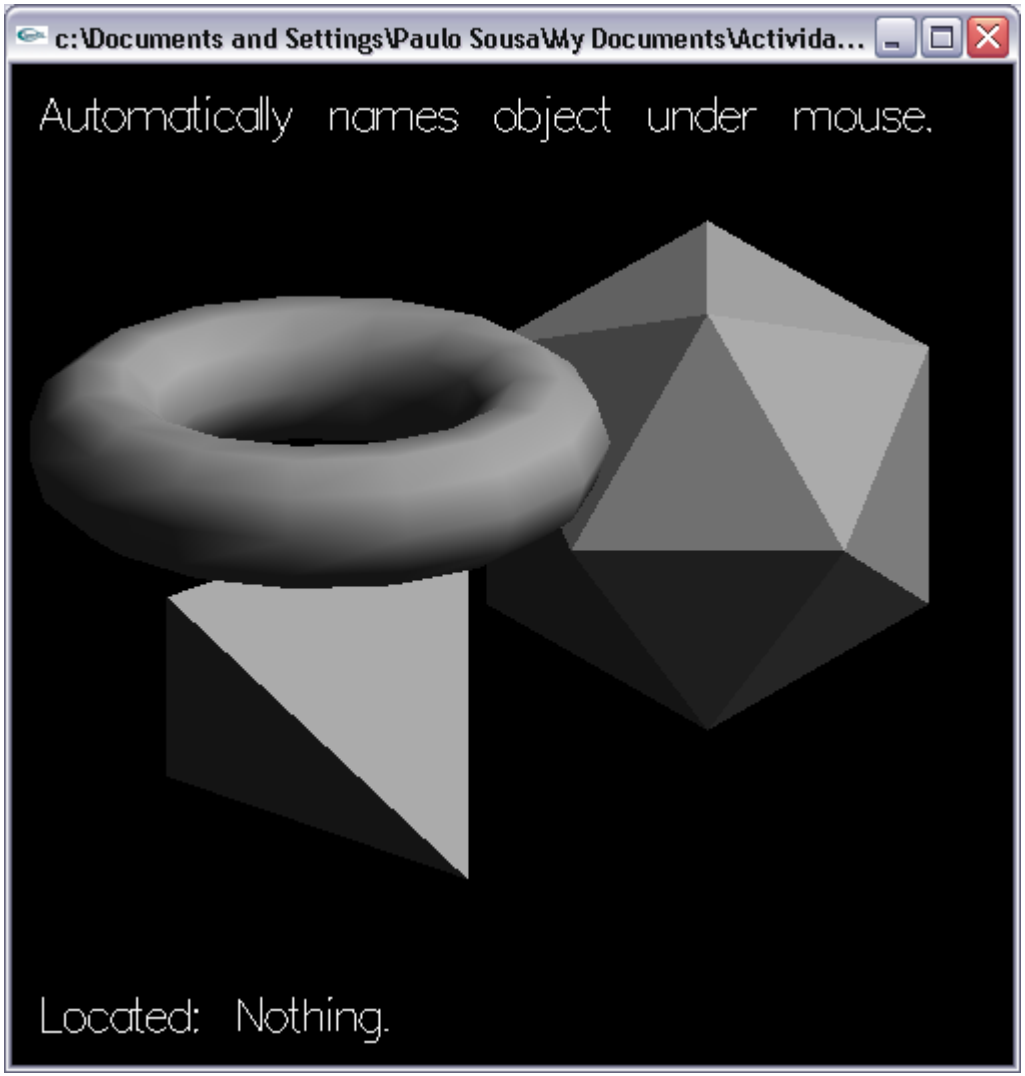


- Entrar no modo selecção através de evento do rato
  - glutPassiveMotionFunc
  - glutMouseFunc
- Definir a matriz de sensibilidade – GL_PROJECTION

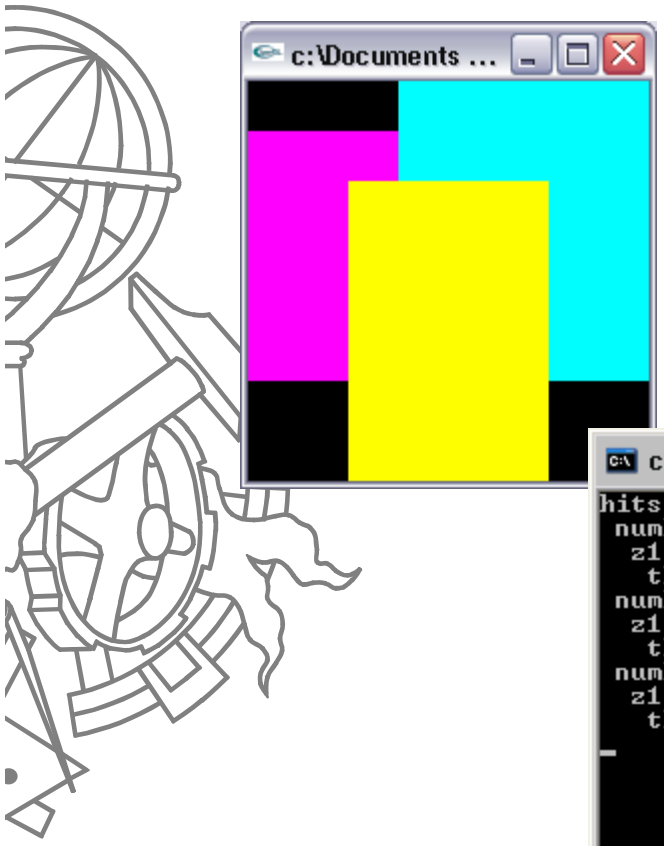
```
GLint viewport[4];
glGetIntegerv(GL_VIEWPORT, viewport);
//create 5x5 pixel picking region near cursor location
gluPickMatrix((GLdouble)x, (GLdouble)(viewport[3]-y),
   5.0, 5.0, viewport);
// definir matriz de projecção
...
```

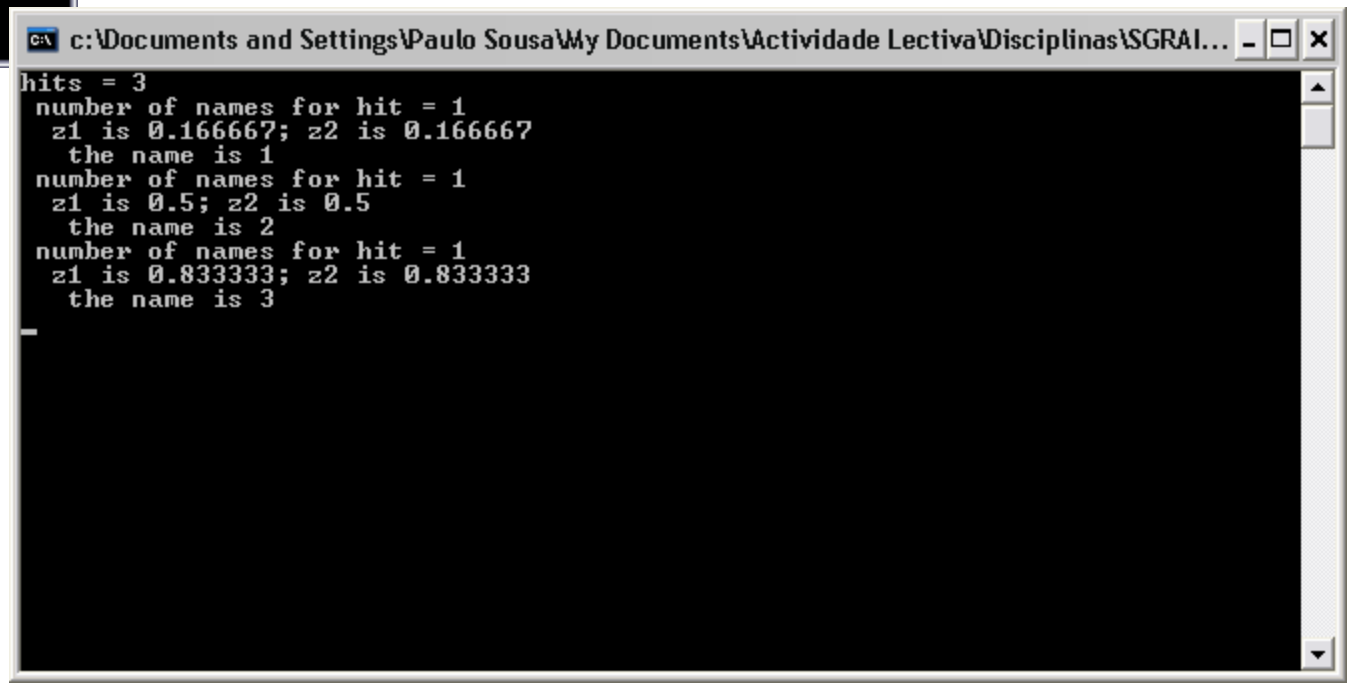
Demo - highlight



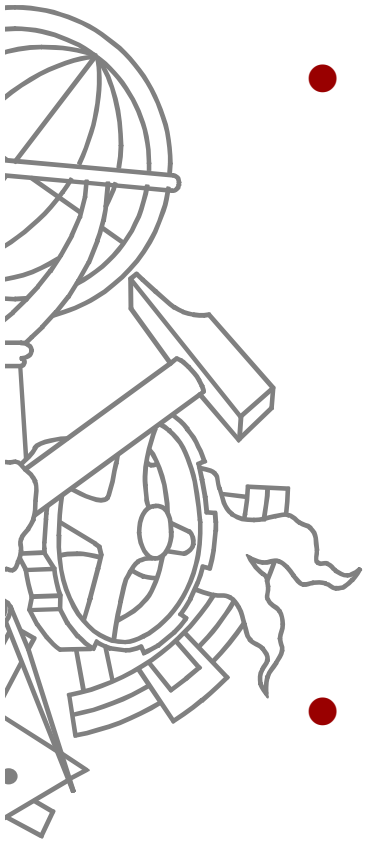


Demo - pickdepth

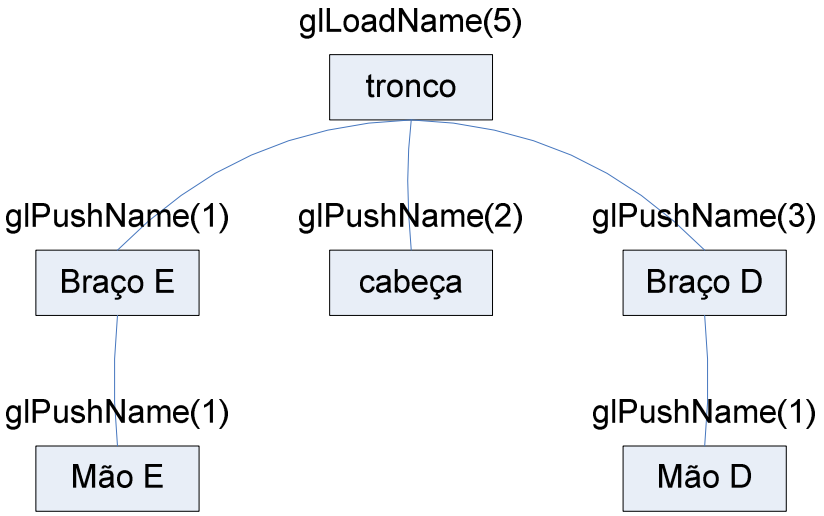




Múltiplos níveis de nomes



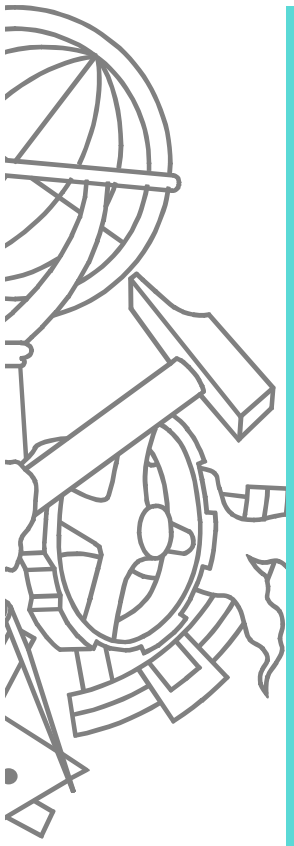
- Para permitir trabalhar com objectos hierárquicos é possível definir uma stack de nomes e não apenas um nome para o objecto



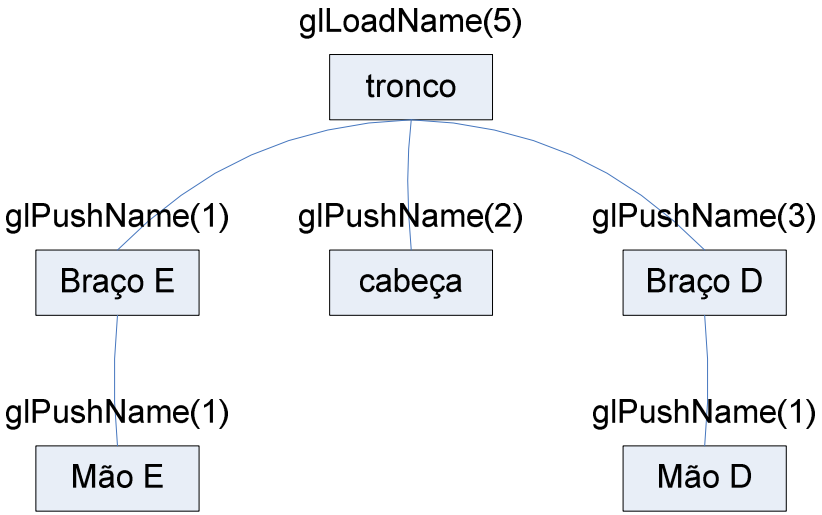
- Se utilizador selecciona-se o objecto “mão D”, o *hit record* seria:

3	*z1*	*z2*	5	3	1

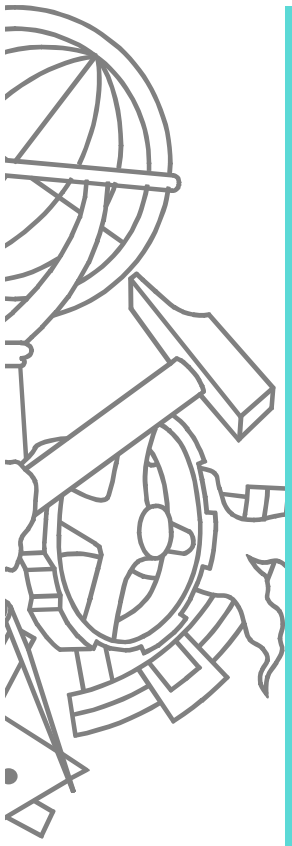
Exemplo



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

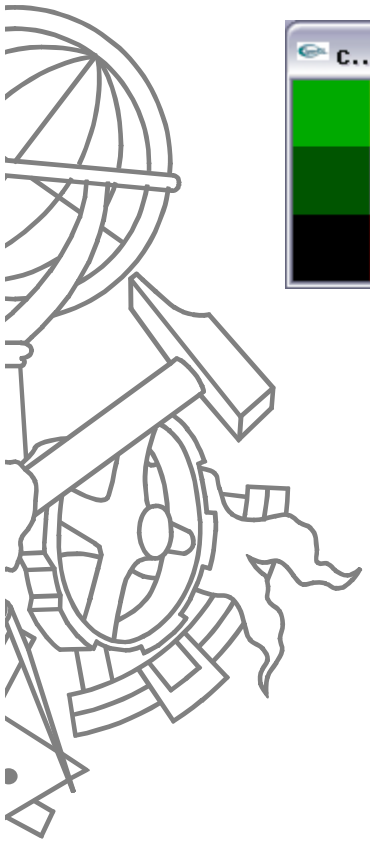


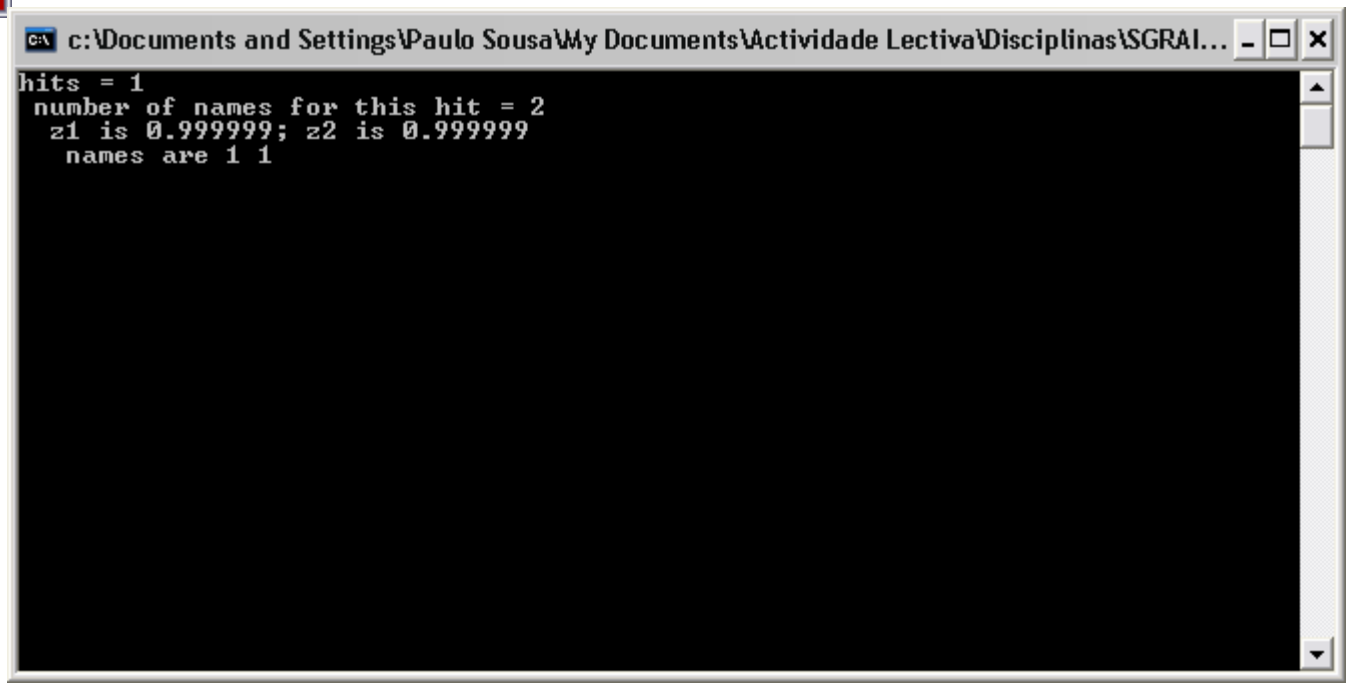
Exemplo



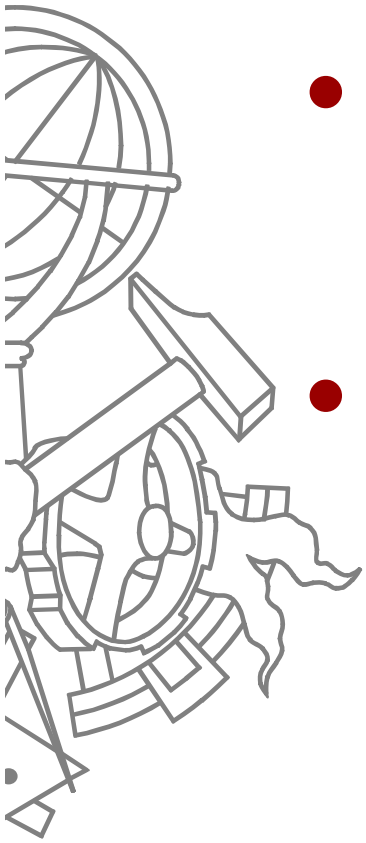
```
void drawSquares(GLenum mode)
{
    GLuint i, j;
    for (i = 0; i < 3; i++)
    {
        if (mode == GL_SELECT)
            glLoadName(i);
        for(j = 0; j < 3; j ++)
         {
            if (mode == GL_SELECT)
                glPushName(100+j);
            glColor3f(i/3.0, j/3.0, board[i][j]/3.0);
            glRecti(i, j, i+1, j+1);
            if (mode == GL_SELECT)
                glPopName();
        }
    }
}
```

Demo - picksquare



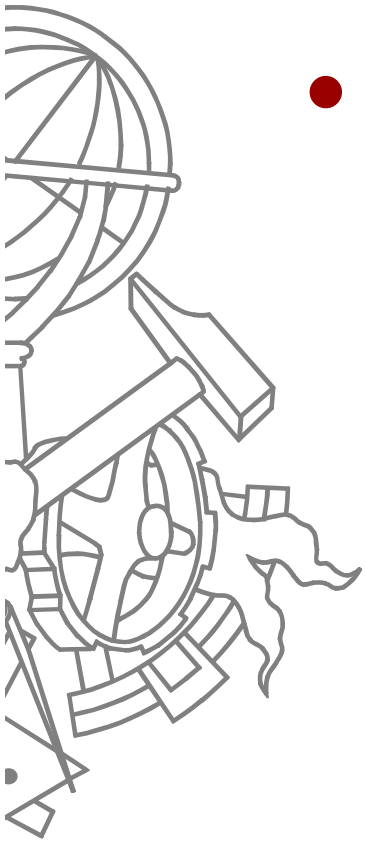


feedback



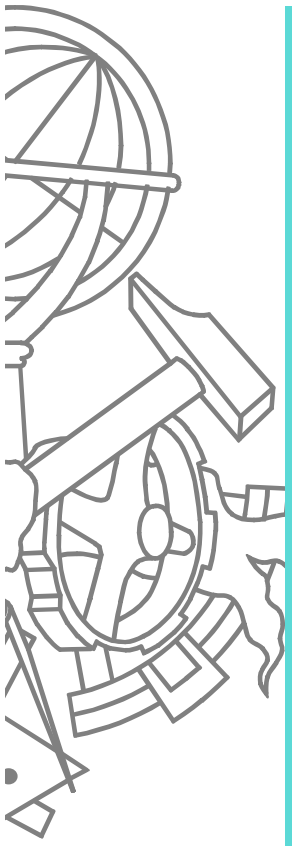
- O que faz?
  - Saber o que seria desenhado
    - Informação vectorial
- Para que serve?
  - Gerar comandos de impressora
  - Gravar em ficheiro num formato vectorial, ex., DXF, windows metafile

feedback



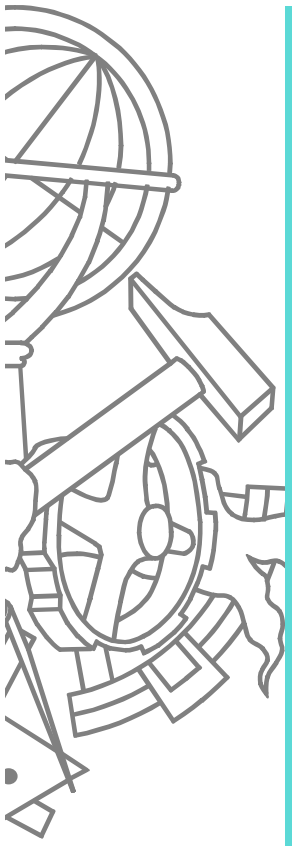
- Passos a executar:
  1. Definir buffer de retorno
  2. Entrar no modo de feedback
  3. “Desenhar” os objectos
  4. Sair do modo de feedback
  5. Processar informação de retorno

Exemplo



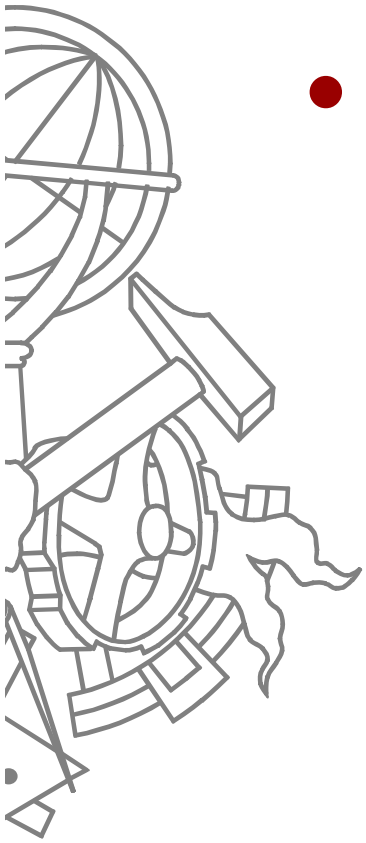
```
void display(void)
{

   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



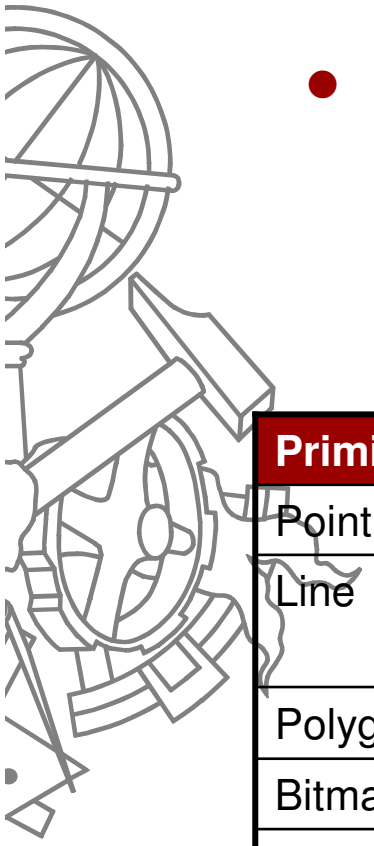
```
void drawGeometry(GLenum mode)
{
   glBegin (GL_LINE_STRIP);
   glNormal3f (0.0, 0.0, 1.0);
   glVertex3f (30.0, 30.0, 0.0);
   glVertex3f (50.0, 60.0, 0.0);
   glVertex3f (70.0, 40.0, 0.0);
   glEnd ();
   if (mode == GL_FEEDBACK)
      glPassThrough (1.0);
   glBegin (GL_POINTS);
   glVertex3f (-100.0, -100.0, -100.0);  /*  will be clipped
   */
   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 :=
    - GL_2D
    - GL_3D
    - GL_3D_COLOR
    - GL_3D_COLOR_TEXTURE

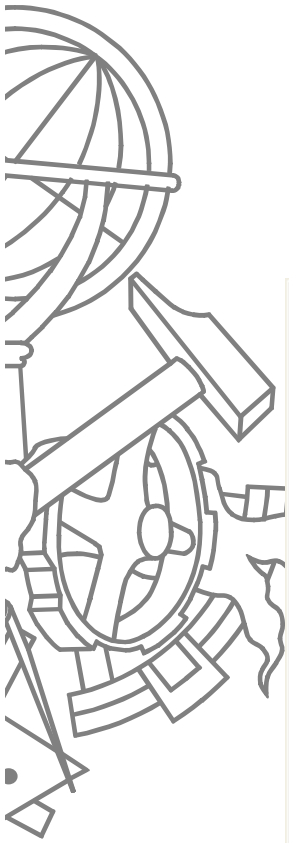
Informação de retorno

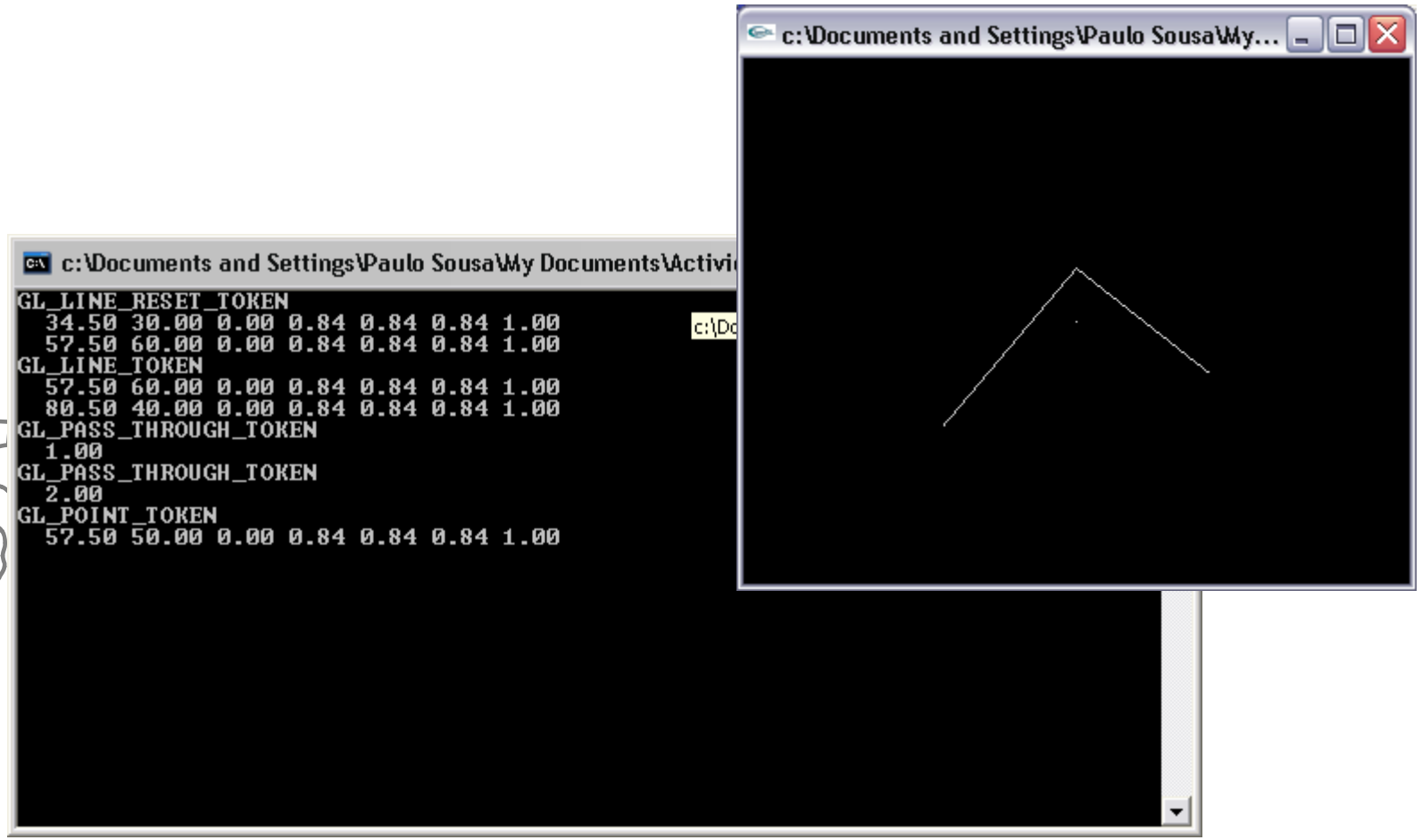


- O buffer de retorno é preenchido com informação de tamanho variável
  - Código da primitiva, dados
  - Vértices são coordenadas de janela

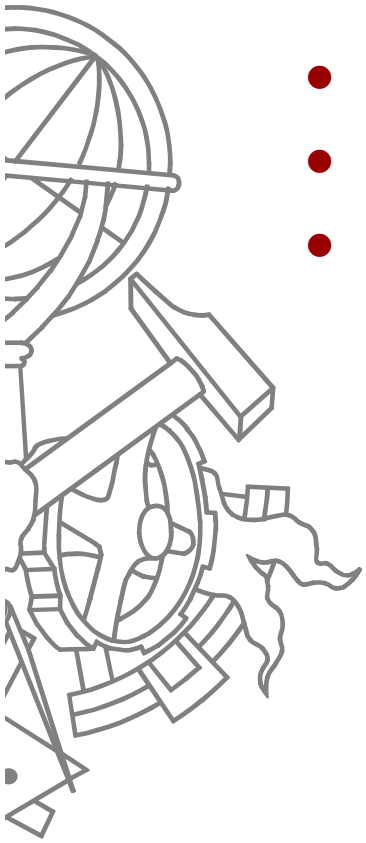
Primitive	Type Code	Associated Data
Point	GL_POINT_TOKEN	vertex
Line	GL_LINE_TOKEN or GL_LINE_RESET_TOKEN	vertex vertex
Polygon	GL_POLYGON_TOKEN	n vertex vertex ... vertex
Bitmap	GL_BITMAP_TOKEN	vertex
Pixel Rectangle	GL_DRAW_PIXEL_TOKEN or GL_COPY_PIXEL_TOKEN	vertex
Pass-through	GL_PASS_THROUGH_TOKEN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

Demo




c:\Documents and Settings\Paulo Sousa\My Documents\Activi
GL_LINE_RESET_TOKEN
34.50 30.00 0.00 0.84 0.84 0.84 1.00
57.50 60.00 0.00 0.84 0.84 0.84 1.00
GL_LINE_TOKEN
57.50 60.00 0.00 0.84 0.84 0.84 1.00
80.50 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
57.50 50.00 0.00 0.84 0.84 0.84 1.00
c:\Documents and Settings\Paulo Sousa\My...

Exportar para PS e WMF



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