



Módulo 5

Sistemas Gráficos e Interação
Instituto Superior de Engenharia do Porto

Filipe Pacheco
ffp@isep.ipp.pt

Display lists

Display lists

- ◎ Sequências pré-compiladas de comandos OpenGL
 - ◎ Evitam chamadas ao OpenGL
 - ◎ Melhoram performance ao evitar cálculos repetitivos
 - ◎ Facilitam leitura de código
 - ◎ Guardam os valores calculados dos parâmetros
 - ◎ Não podem ser alteradas após criação

Exemplo

```
void torus(int numc, int numt)
{
    double s, t, x, y, z, twopi;
    twopi = 2 * (double)M_PI;
    for (int i = 0; i < numc; i++)    {
        glBegin(GL_QUAD_STRIP);
        for (int j = 0; j <= numt; j++)
        {
            for (int k = 1; k >= 0; k--)
            {
                s = (i + k) % numc + 0.5;
                t = j % numt;
                x = (1+.1*cos(s*twopi/numc))*cos(t*twopi/numt);
                y = (1+.1*cos(s*twopi/numc))*sin(t*twopi/numt);
                z = .1 * sin(s * twopi / numc);
                glVertex3f(x, y, z);
            }
        }
        glEnd();
    }
}
```

Cálculos complexos
de cada vez que
desenha um *torus*

Múltiplas chamadas
ao OpenGL

Exemplo

```
GLuint theTorus;
```

```
void init(void)
```

```
{
```

```
    theTorus = glGenLists(1);
```

```
    glNewList(theTorus, GL_COMPILE);
```

```
        torus(8, 25);
```

```
    glEndList();
```

```
    ...
```

```
}
```

```
void display(void)
```

```
{
```

```
    glClear(GL_COLOR_BUFFER_BIT);
```

```
    glColor3f (1.0, 1.0, 1.0);
```

```
    glCallList(theTorus);
```

```
    glFlush();
```

```
}
```

Comandos **OpenGL** são guardados na *display list*

Display list pode ser invocada quantas vezes quiser. Cálculos só são efectuados uma vez

Criar lista

- ⊙ `GLuint glGenLists(quantidade)`
- ⊙ `void glNewList(list, mode)`
 - ⊙ `GL_COMPILE`
 - ⊙ `GL_COMPILE_AND_EXECUTE`
- ⊙ `void glEndList()`

- ⊙ Que instruções se podem usar
 - ⊙ `glBegin`, `glEnd` e todas as possíveis aí “dentro” (`glColor`, `glVertex`, ...)
 - ⊙ Transformações, projeções
 - ⊙ Execução de outras *display lists*

Utilizar lista

- ⊙ `glCallList(list)`
- ⊙ `glCallLists(GLsizei n,
GLenum type,
const GLvoid *lists)`

Outras operações

- ⊙ `glIsList(GLuint i)`
- ⊙ `glDeleteLists(GLuint list, GLsizei range)`

Cilindro com topo e fundo

```
int cylinderWithTopAndBottom(GLenum mode)
{
    int list = glGenLists(2);
    GLUquadricObj* qobj = gluNewQuadric();
    gluQuadricDrawStyle(qobj, mode);
    gluQuadricNormals(qobj, GLU_SMOOTH);

    glNewList(list+1, GL_COMPILE);
        gluDisk(qobj, 0, 1.5, stacks, slices);
    glEndList();
    glNewList(list, GL_COMPILE);
        glCallList(list+1);
        gluCylinder(qobj, 1.5, 1.5, 2, slices, stacks);
        glTranslatef(0, 0, +2);
        glRotatef(180,1,0,0);
        glCallList(list+1);
    glEndList();

    gluDeleteQuadric(qobj);
    return list;
}
```

