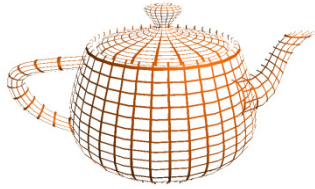




Computação Gráfica

Display Lists



OpenGL - Display Lists

- Setup

```
GLuint glGenLists( GLsizei range );
```

Gera um conjunto de listas sequenciais. Devolve o índice da primeira lista.

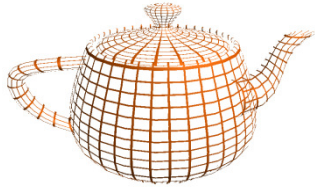
- Definição

```
void glNewList( GLuint list, GLenum mode );
```

```
void glEndList( void );
```

list: o índice da lista

mode: `GL_COMPILE` ou `GL_COMPILE_AND_EXECUTE`



OpenGL - Display Lists

- Utilização

glCallList(GLuint list);

realiza o render de uma *display list* com o índice *list*.

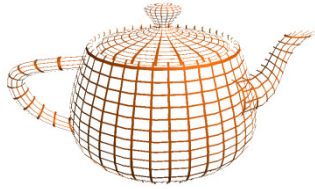
glCallLists(GLsizei n, GLenum type, const GLvoid *lists);

render de um conjunto de *display lists*, cujos índices estão especificados no *array lists*.

- Libertar recursos

void glDeleteLists(GLuint listID, GLsizei numberOfLists);

Liberta recursos de uma sequência de comprimento *numberOfLists*, a partir do índice *listID*.



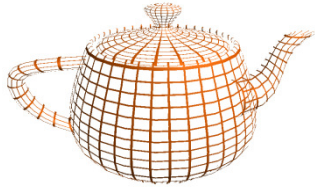
OpenGL - Display Lists

Setup e Geração

```
id = GenLists(1);
glNewList(id, GL_COMPILE);
    glTranslatef(10.0f, 5.0f, 3.0f);
    glBegin(GL_TRIANGLES);
        glVertex3f(1.0f, 1.0f, 1.0f);
        ...
    glEnd();
glEndList();
```

Utilização

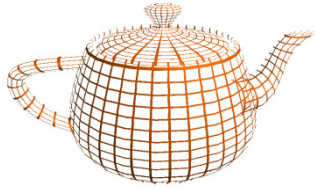
```
glCallList(id);
```



OpenGL - Display Lists

- Sem Display Lists

```
for(i = -3; i < 3; i++)
    for(int j=-3; j < 3; j++) {
        glPushMatrix();
        glTranslatef(i*10.0,0,j * 10.0);
        drawSnowMan();
        glPopMatrix();
    }
```

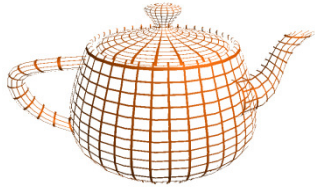


OpenGL - Display Lists

- Display List para o boneco

Setup

```
snowManDL = glGenLists(1);  
  
glNewList(snowManDL, GL_COMPILE);  
    drawSnowMan();  
glEndList();
```

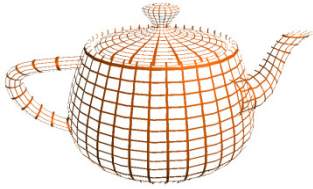


OpenGL - Display Lists

- Display List para o boneco

Utilização

```
for(i = -3; i < 3; i++)  
    for(int j=-3; j < 3; j++) {  
        glPushMatrix();  
        glTranslatef(i*10.0,0,j * 10.0);  
        glCallList (snowManDL);  
        glPopMatrix();  
    }
```



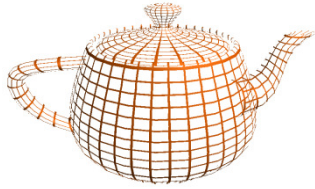
OpenGL - Display Lists

- Display List com todos os bonecos de neve
- Setup

```
glNewList(loopDL, GL_COMPILE);  
for(int i = -3; i < 3; i++)  
    for(int j=-3; j < 3; j++) {  
        glPushMatrix();  
        glTranslatef(i*10.0, 0, j * 10.0);  
        drawSnowMan();  
        glPopMatrix();  
    }  
glEndList();
```

- Utilização

```
glCallList(loopDL);
```

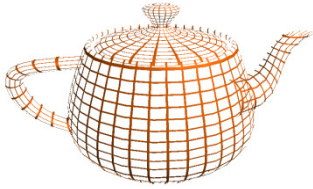



OpenGL - Display Lists

- *Display List Hierárquica*

Setup

```
loopDL = glGenLists(2);
glNewList(loopDL+1, GL_COMPILE);
    drawSnowMan();
glEndList();
glNewList(loopDL, GL_COMPILE);
for(int i = -3; i < 3; i++)
    for(int j=-3; j < 3; j++) {
        glPushMatrix();
        glTranslatef(i*10.0, 0, j * 10.0);
        glCallList(loopDL+1);
        glPopMatrix();
    }
glEndList();
```



FPS

```
int time;  
time = glutGet(GLUT_ELAPSED_TIME);
```

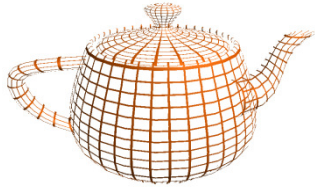
- devolve o número de milisegundos desde a invocação de glutInit

...

```
frame++;  
time=glutGet(GLUT_ELAPSED_TIME);  
if (time - timebase > 1000) {  
    fps = frame*1000.0/(time-timebase);  
    timebase = time;  
    frame = 0;  
}
```

...

Para escrever o valor utilizar a função `glutSetWindowTitle(char *s);`



Exercício

- Partindo do exemplo da aula passada:
 - Utilizar display lists para as árvores do cenário virtual
 - Explorar as diferentes hipóteses de utilização de display lists considerando ainda números de lados das árvores diferentes (10,1000,1000000)
 - Medir FPS