

Exercício: Construindo Quádricas em OpenGL

André Tavares da Silva

andre.silva@udesc.br

Construindo (modelando) Objetos

- Em OpenGL é possível modelar combinando objetos pré-definidos.
- Formas quádricas.
- Objetos complexos requerem maior complexidade matemática. OpenGL dá suporte a geração de curvas e superfícies Bézier e NURBS que *veremos mais adiante*.

Quádricas

- Diversos objetos podem ser modelados apenas com o uso de formas quádricas.
- Para tanto, basta combiná-las para compor objetos mais complexos, como:
 - Visualização de moléculas em 3D (combinando esferas e cilindros)
 - Diversos objetos utilizados em jogos, como por exemplos, armas, veículos, projéteis, edificações e etc.

Quádricas no OpenGL

- Semelhante ao modelo OO, mantendo um estado do objeto "quádrico" na memória:
- 1. Criar um objeto que representa o estado da quádrlica deve ser mantido para definir propriedades das quádrlicas.
- 2. Definir propriedades para o objeto
- 3. Desenhar a forma específica

```
GLUquadricObj *pObj;  
pObj = gluNewQuadric();  
// ... Definições (estados)  
gluDeleteQuadric(pObj);
```

Quádricas no OpenGL

- Funções para definição de estados:

- Estilo de desenho:

```
gluQuadricDrawStyle (GLUquadricObj *obj,  
GLenum drawStyle)
```

```
GLU_FILL, GLU_LINE, GLU_POINT e  
GLU_SILHOUETTE
```

- Normais da superfície:

```
gluQuadricNormals (GLUquadricObj *obj, GLenum  
normals)
```

```
GLU_NONE, GLU_FLAT (normais dos vértices  
perpendiculares as faces) e GLU_SMOOTH  
(pondera normais com as das faces  
adjacentes)
```

Quádricas no OpenGL

- Funções para definição de estados:

- Orientação das normais:

- `gluQuadricOrientation (GLUquadricObj *obj, GLenum orientation)`

- `GLU_OUTSIDE` e `GLU_INSIDE`

- Considerar textura ou não:

- `gluQuadricTexture (GLUquadricObj *obj, GLenum texture)`

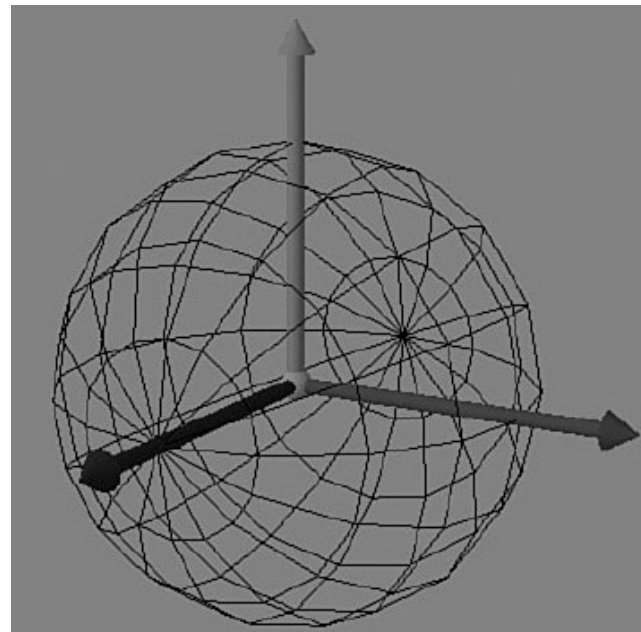
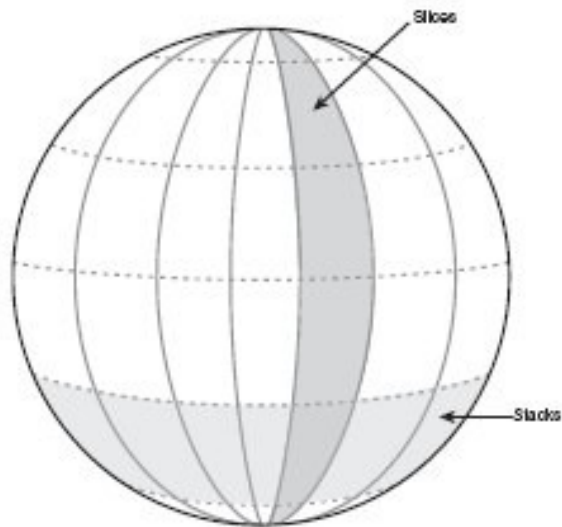
- `GL_TRUE` (considera mapeamento de texturas) e `GLU_FALSE` (não considera o mapeamento de textura) .

Desenhando quádricas

- Esferas:

```
gluSphere(GLUquadricObj *obj, GLdouble  
radius, GLint slices, GLint stacks)
```

Slices == gomos. Stacks == rodela

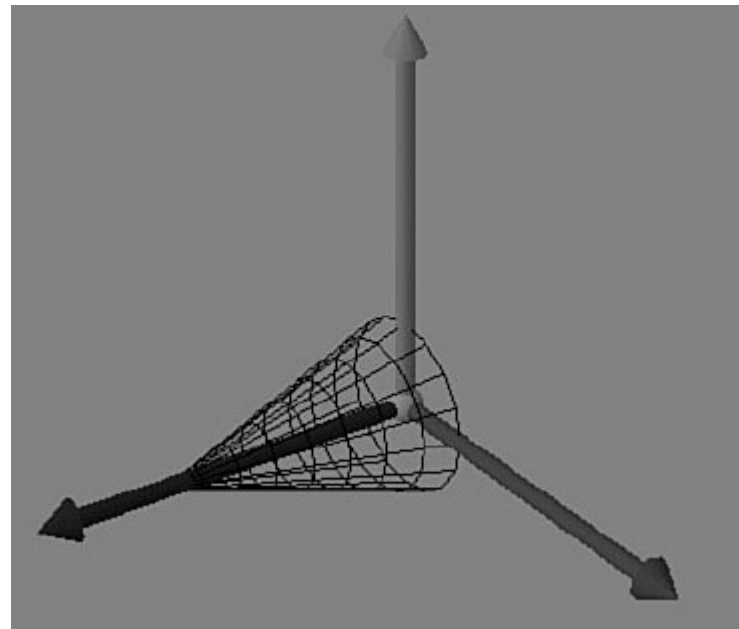
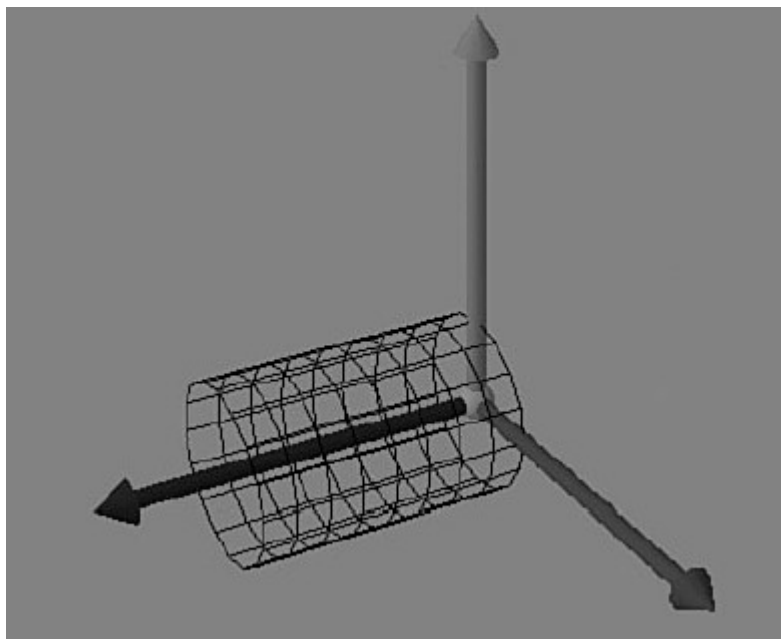


Desenhando quádricas

- Cilindros:

```
gluCylinder(GLUquadricObj *obj, GLdouble  
baseRadius, GLdouble topRadius, GLdouble  
height, GLint slices, GLint stacks)
```

topRadius == zero, permite criar cone.

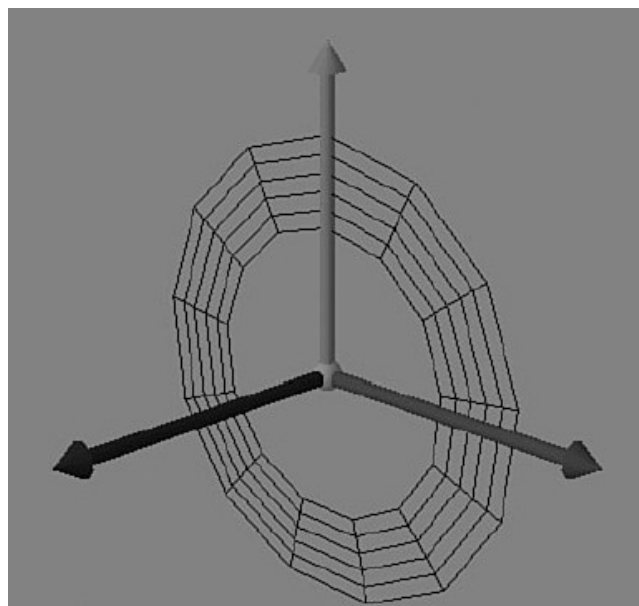
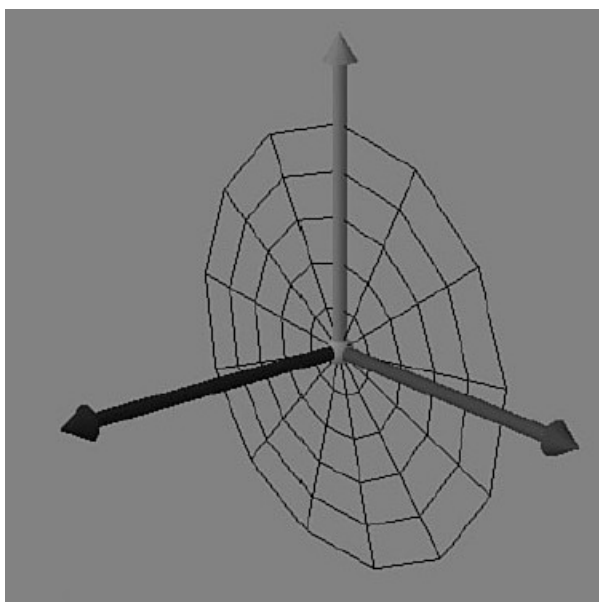


Desenhando quádricas

- Discos:

```
gluDisk(GLUquadricObj *obj, GLdouble  
innerRadius, GLdouble outerRadius, GLint  
slices, GLint loops)
```

innerRadius != zero, permite criar
discos com furos.



Desenhando quádricas

- Variável de classe (declaração):

```
GLUquadriObj *obj;
```

- *init*:

```
obj = gluNewQuadric();
```

- *RenderScene* (ou *display*):

```
gluQuadricNormals(obj, GLU_SMOOTH);
```

```
glTranslatef(x,y,z);
```

```
gluSphere(obj, 5, 5, 5);
```

- No final da execução:

```
gluDeleteQuadric (obj);
```

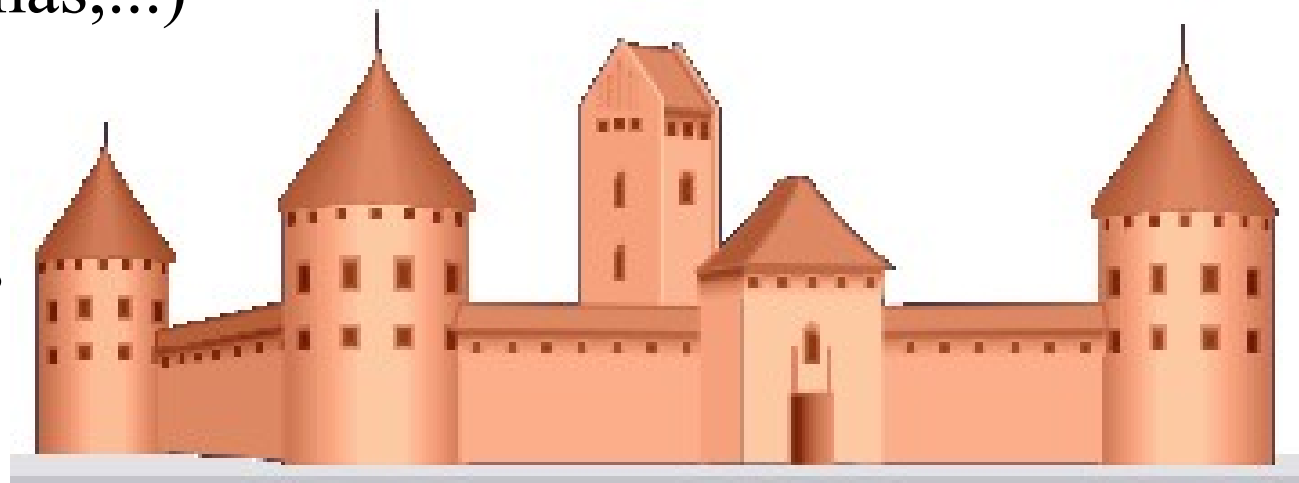
Modelagem com quádricas

- Vantagem:
 - Facilidade de modelagem (criação, propriedades, composição e desenho)
- Desvantagens:
 - Limitação de interação com o usuário
 - Pode-se tornar difícil a criação de um cenário grande.
 - Limitação de formas matemáticas

Trabalho Complementar

- 1) Crie um boneco de neve utilizando quádricas
- 2) Modele a construção abaixo utilizando quádricas
- 3) Desenhe um robô humanoide utilizando quádricas e garanta que ele possa ser articulado no futuro (mover braços, pernas,...)

Entrega:
ver slide de datas



Exemplos



Exemplo



Exemplos

