

COMPUTAÇÃO GRÁFICA E REALIDADE VIRTUAL

Primitivas Gráficas 3D

Prof. Dr. Fernando Kakugawa

fernando.kakugawa@animaeducacao.com.br

Cena – Modificações

Import para a GLUT

```
import com.jogamp.opengl.util.gl2.GLUT; //primitivas 3D
```

init() - Habilitar o buffer de profundidade

```
gl.glEnable(GL2.GL_DEPTH_TEST);
```

reshape()

```
//xMin, xMax, yMin, yMax, zMin, zMax
```

```
gl.glOrtho(-100, 100, -100, 100, -100, 100);
```

display()

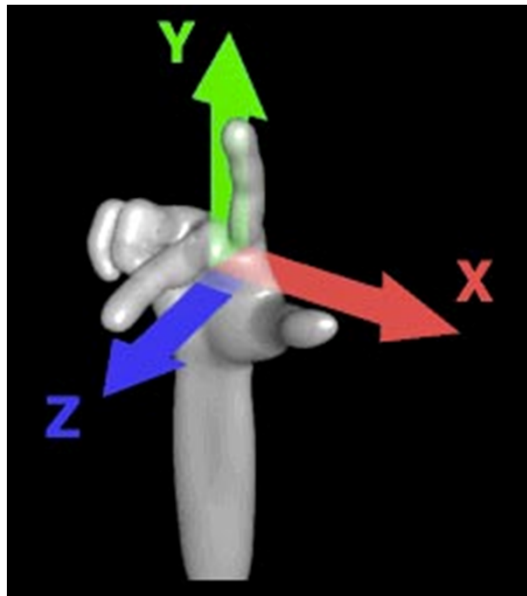
```
GLUT glut = new GLUT(); //objeto da biblioteca glut
```

Limpar o buffer de profundidade

```
gl.glClear(GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);
```

Sistema de Referência do Universo – SRU

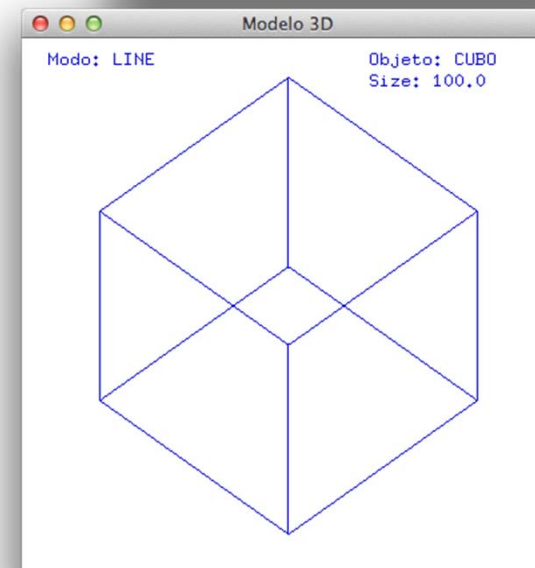
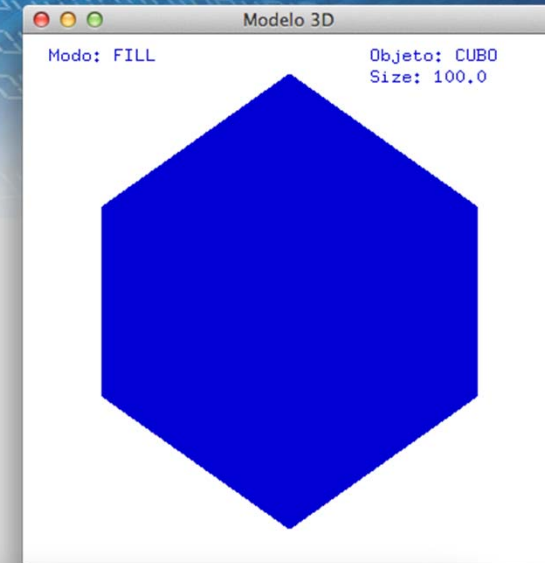
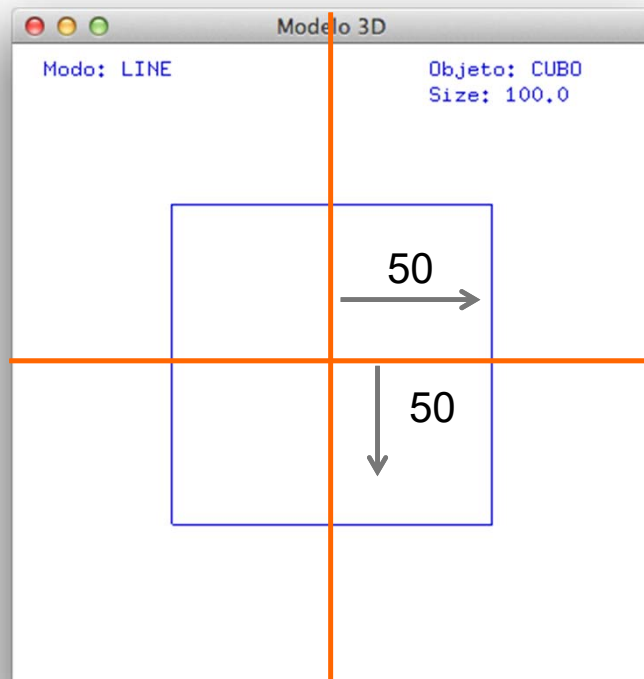
- SRU – agora formado por 3 eixos ortogonais (x, y, z) com origem em (0, 0, 0) – centro do sistema
- Usar a regra da “mão direita” para identificar como o eixo z é posicionado em relação à x e y



- Indicador aponte para direção positiva de y (**para cima**),
- Polegar aponte para a direção positiva de x (**para o lado direito**)
- Dedo do meio aponte para a direção positiva de z (**para frente**, como se estivesse “saindo da tela do computador”)

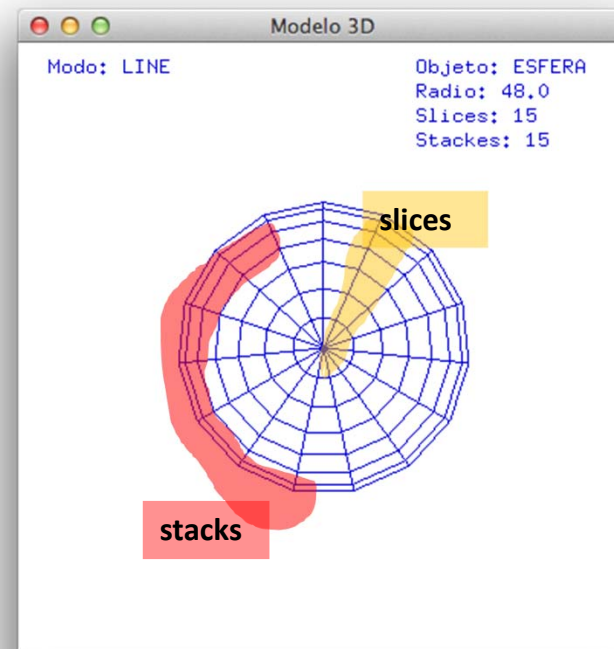
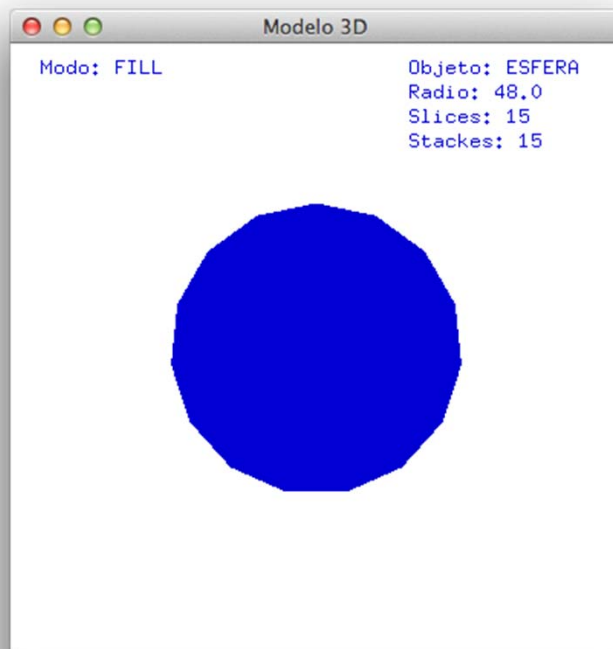
`glutSolidCube(float size)`

- Se size for 100



Esfera

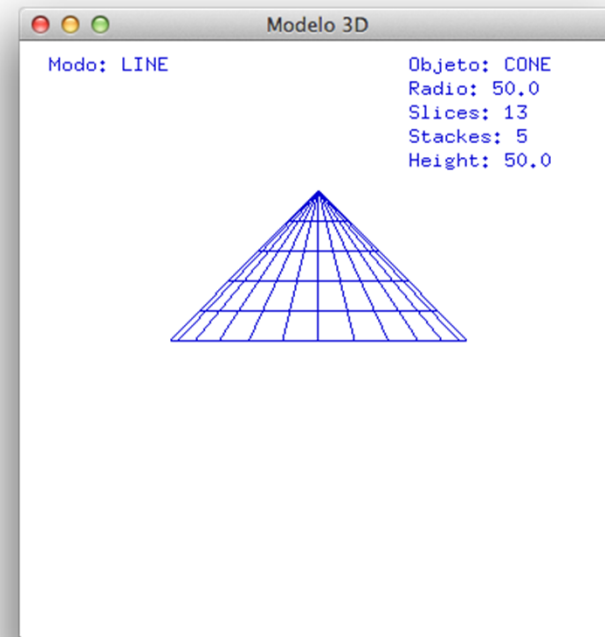
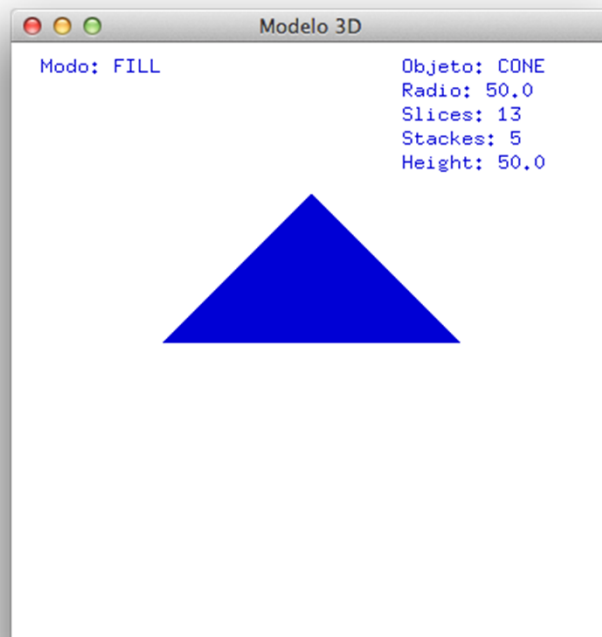
```
glutSolidSphere(double radius, int slices, int stacks)
```



Cone

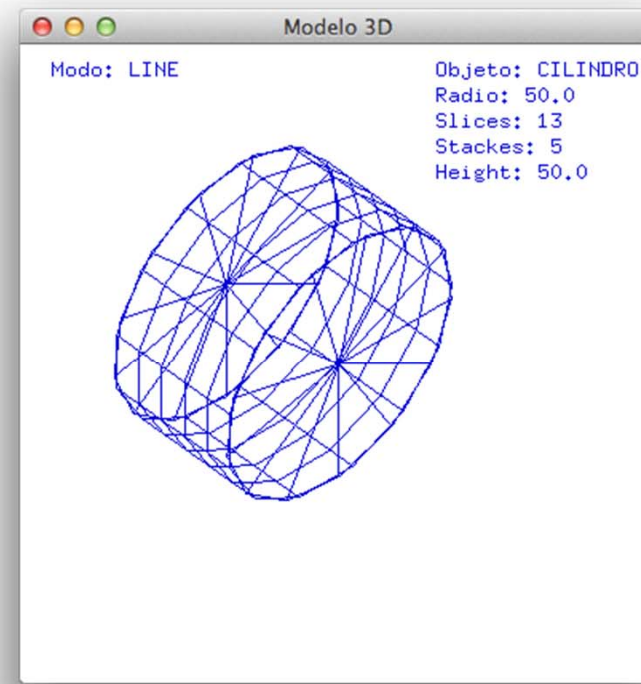
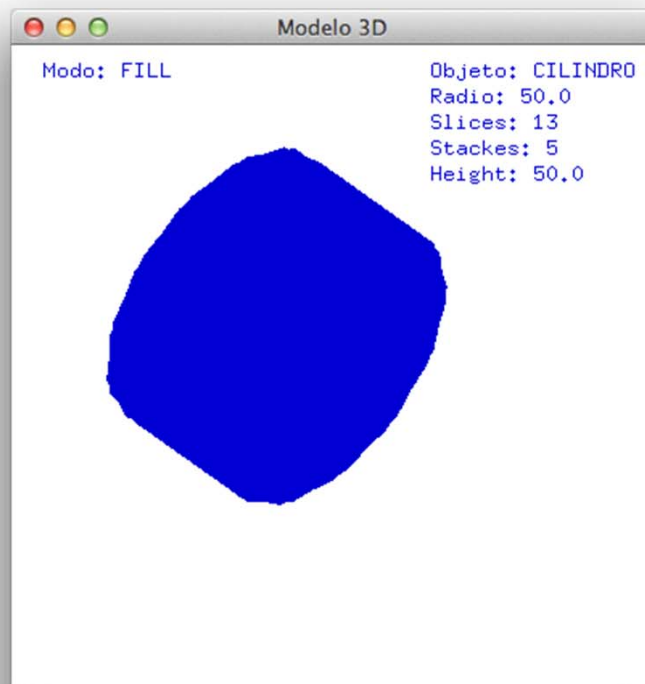
`glutSolidCone(double radius, double height, int slices, int stacks)`

- Rotação de 90° em relação ao eixo X no sentido horário



Cilindro

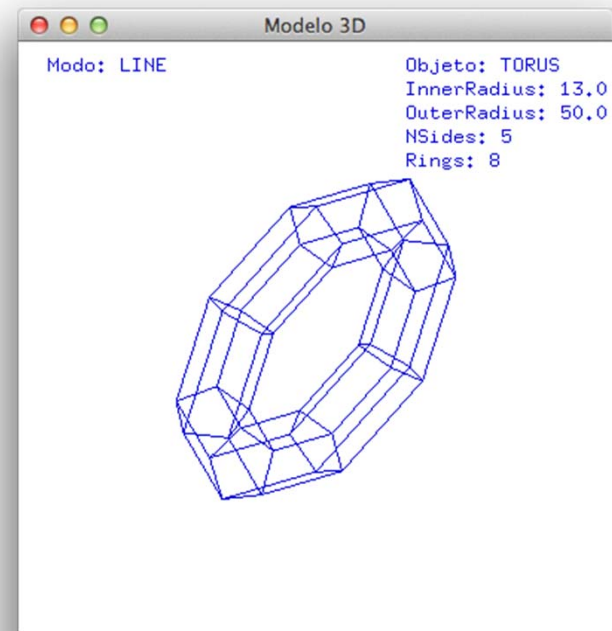
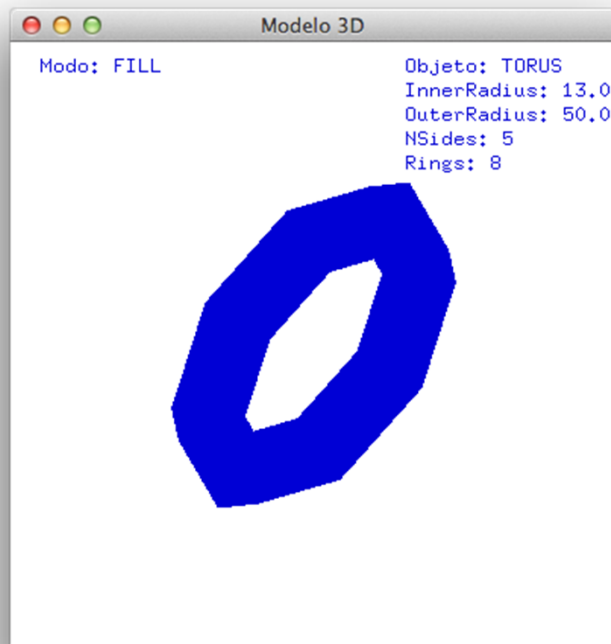
```
glutSolidCylinder(double radius, double height,  
int slices, int stacks)
```



TORÓIDE

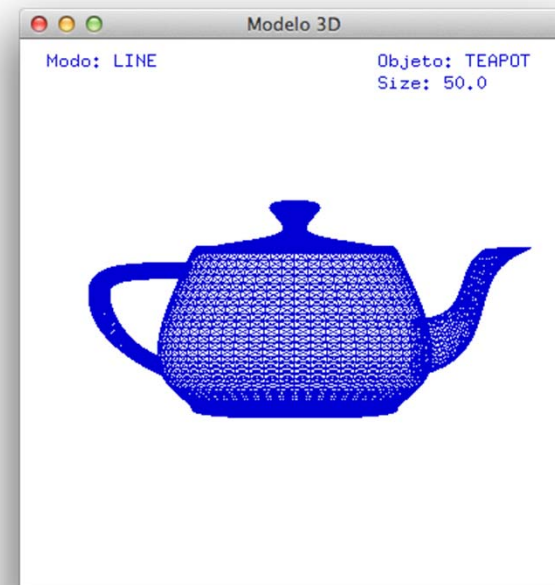
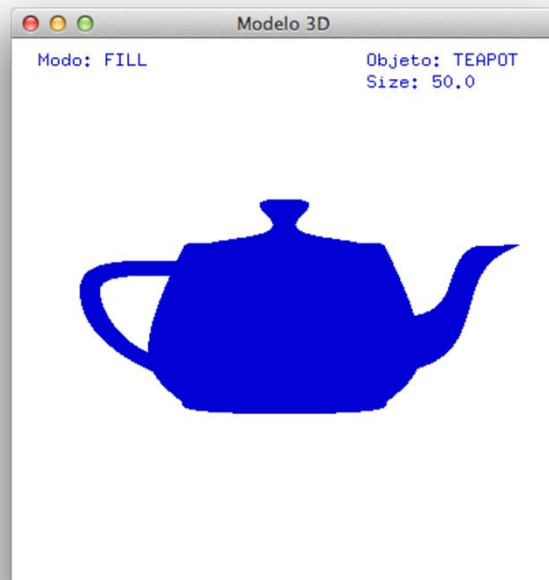
`glutSolidTorus(double innerRadius, double outerRadius, int nsides, int rings)`

- `innerRadius` – raio do anel – espessura do cilindro
- `outerRadius` – raio do torus
- `rings` – quantidade de anéis



CHALEIRA

```
glutSolidTeapot(double scale);
```



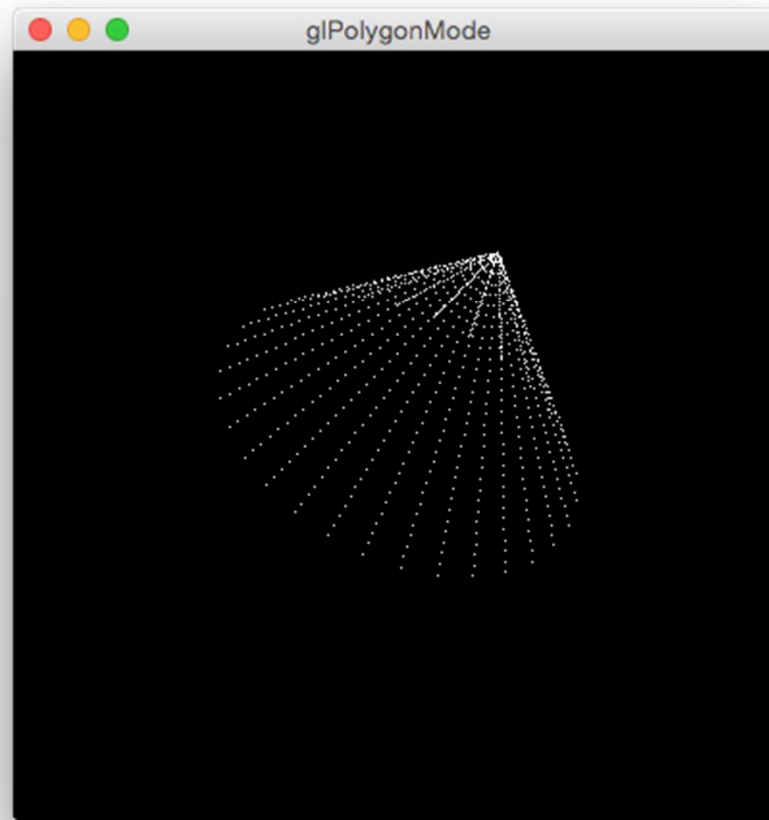
Modo de DESENHO – glPolygonMode()

`glPolygonMode(GLenum face, GLenum mode)`

- Método que exibe objetos de vários modos
- Parâmetros:
 - Face
 - GL_FRONT somente frente
 - GL_BACK somente fundo
 - GL_FRONT_AND_BACK frente e fundo
 - Mode
 - GL_LINE somente linhas
 - GL_FILL preenchido
 - GL_POINT somente vértices
- Utilizar antes de desenhar o objeto

glPolygonMode()

```
// criar a cena aqui....  
gl.glPolygonMode(GL2.GL_FRONT_AND_BACK, GL2.GL_POINT);  
  
gl.glRotatef(angulo, 1, 0, 1);  
glut.glutSolidCone(50, 70, 30, 30);
```





SUPERFÍCIES QUÁDRICAS

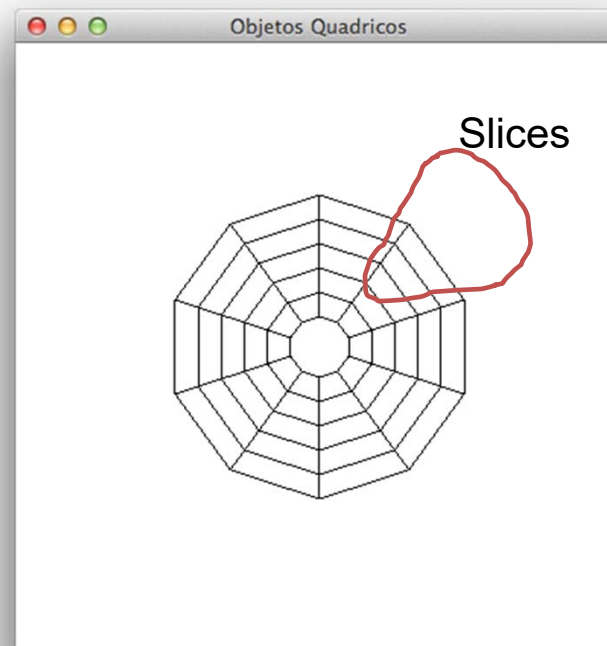
Superfícies Quádricas

- Uma superfície quádrlica é representada por uma equação do segundo grau
- Podem ser utilizadas para modelar algumas formas comuns
 - Cilindros
 - Discos
 - Discos Parciais
 - Esferas

Discos

```
gluDisk( GLUquadric quad, double inner, double  
outer, int slices, int loops)
```

```
glu.gluDisk(glu.gluNewQuadric(), 10, 50, 10, 5);
```

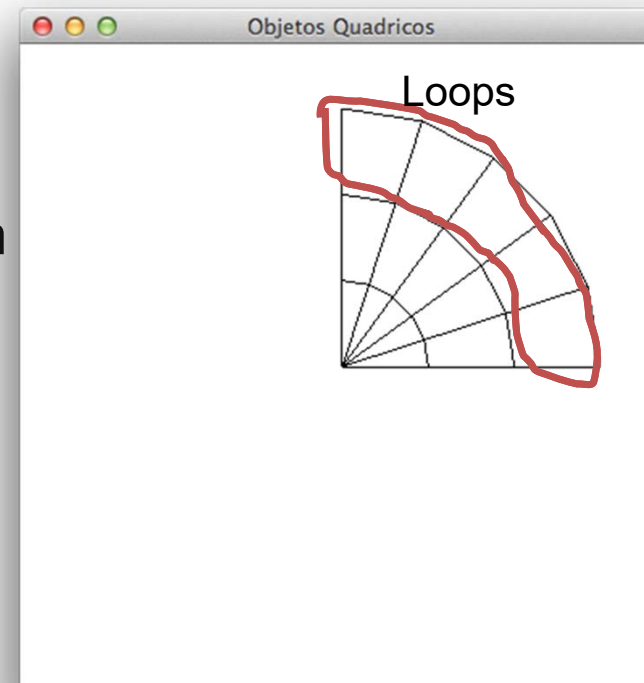


Discos Parciais

`gluPartialDisk(GLUquadric quad, double inner, double outer, slices, loops, double startAngle, double sweepAngle`

- **Parâmetro**

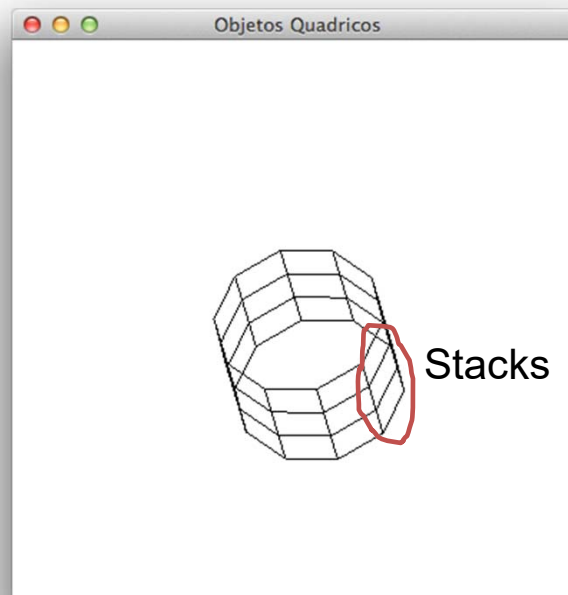
- loops – número de anéis concêntricos sobre a origem do disco
- startAngle – ângulo de início
- sweepAngle – ângulo de varredura



```
glu.gluPartialDisk(glu.gluNewQuadric(), 0, 80, 5, 3, 0.0, 90.0);
```

Cilindros

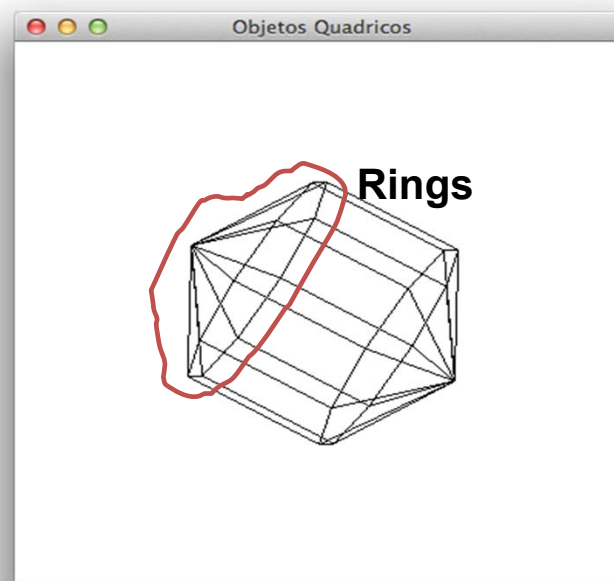
```
gluCylinder(GLUquadric quad, double base, double top,  
double height, int slices, int stacks)
```



```
glu.gluCylinder(glu.gluNewQuadric(), 30, 30, 50, 10, 3);
```


Esferas

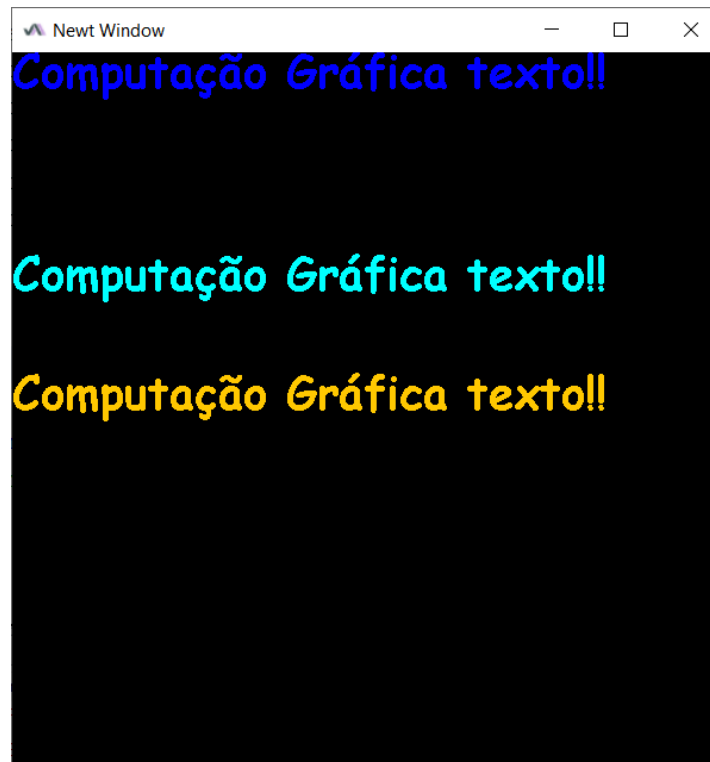
```
gluSphere(GLUquadric quad, double radius, int  
slices, int rings)
```



```
glu.gluSphere(glu.gluNewQuadric(), 50, 10, 3);
```

Texto em JOGL

- Posição do texto na janela – 0,0 está no canto inferior esquerdo da Tela!



Texto em JOGL

```
//na class
private TextRenderer textRenderer;

//init()
textRenderer = new TextRenderer(new Font("Comic Sans MS Negrito", Font.PLAIN, 38));

//display()
desenhaTexto(gl, 0, 570, Color.BLUE, "Computação Gráfica texto!!");

//método para desenho de texto
public void desenhaTexto(GL2 gl, int xPosicao, int yPosicao, Color cor, String frase)
{
    gl.glPolygonMode(GL2.GL_FRONT_AND_BACK, GL2.GL_FILL);
    //Retorna a largura e altura da janela
    textRenderer.beginRendering(Renderer.screenWidth, Renderer.screenHeight);
    textRenderer.setColor(cor);
    textRenderer.draw(frase, xPosicao, yPosicao);
    textRenderer.endRendering();
}
```



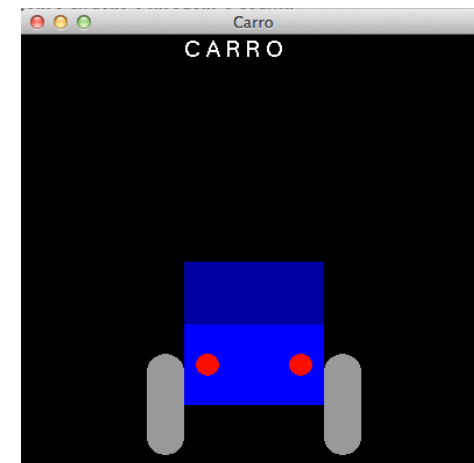
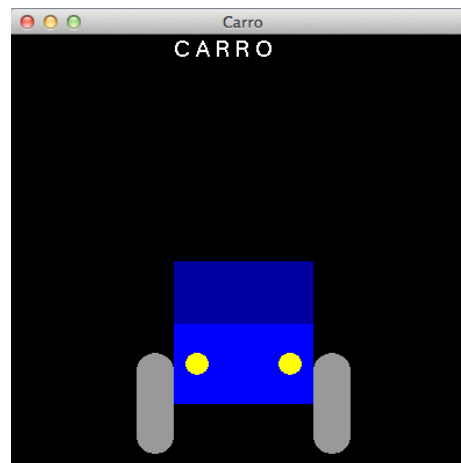
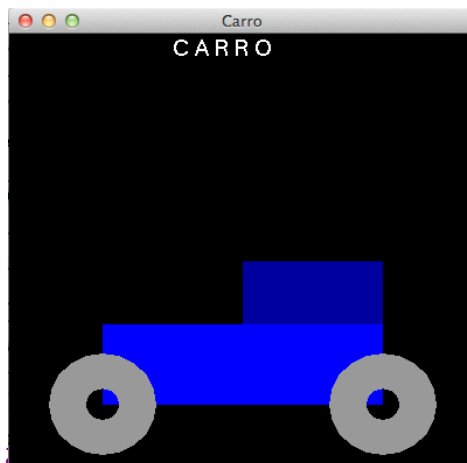
**KEEP
CALM
AND
VAMOS
PRATICAR**

Exercício 1

- Desenhar um cubo 3D na cena e utilizar as setas de direção para girar o cubo (usar wireframe)
 - ← e → girar o cubo 45° (eixo y)
 - ↑ e ↓ girar o cubo 45° (eixos x e z)

Exercício 2

- Desenhar um carro 3D como ilustra as imagens:
 - Implementar uma rotação global para visualizar a construção do objeto adequadamente
 - Criar métodos para desenhar as partes do carro e utilizar as transformações para duplicar objetos como a roda.
 - Utilize os métodos `pushMatrix()` e `popMatrix()` para isolar as transformações



Material elaborado por:

Prof. Ms. Simone de Abreu

siabreu@gmail.com

Prof. Dr. Fernando Kakugawa

fernando.kakugawa@animaeducacao.com.br

