

Entidades cuádricas

Informática Gráfica I

Material de: **Antonio Gavilanes**
Adaptado por: **Elena Gómez y Rubén Rubio**
{mariaelena.gomez,rubenrub}@ucm.es



Contenido

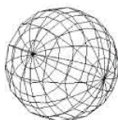
1 Entidades Cuádricas

- Definición
- En OpenGL

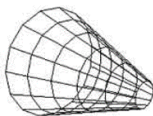
2 Clase QuadricEntity

Entidades cuádricas

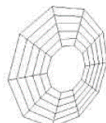
- Superficie geométrica tridimensional definida por una ecuación polinómica de segundo grado en tres variables (x , y y z).
- Se pueden crear cuatro tipos de entidades: (a) esferas, (b) cilindros, (c) discos y (d) discos parciales.



(a)



(b)

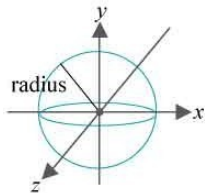


(c)

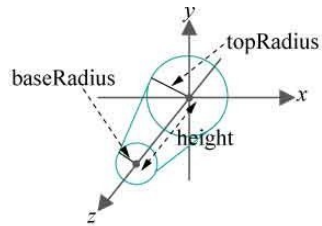


(d)

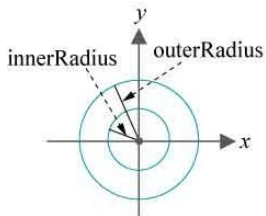
Entidades cuádricas



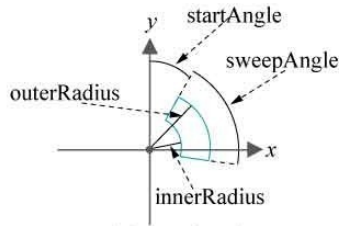
Sphere



Tapered Cylinder



Annular Disc



Partial Annular Disc

Entidades cuádricas en OpenGL

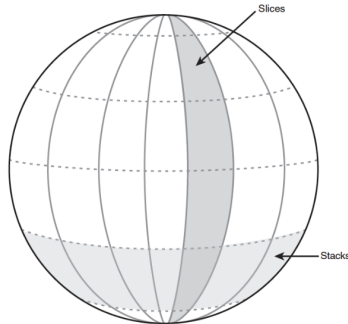
- Pueden ser dibujadas con cierta flexibilidad, incluyendo texturas.
- Las funciones cuádricas utilizan un modelo orientado a objetos donde se crea un objeto cuádrico y se establece el estado de renderización.

Entidades cuádricas en OpenGL

- Se proporcionan con la librería GLU.
- Las entidades se declaran así: `GLUquadricObj* q;`
- Las entidades se construyen así: `q=gluNewQuadric();`
- Las entidades se destruyen así: `gluDeleteQuadric(q);`

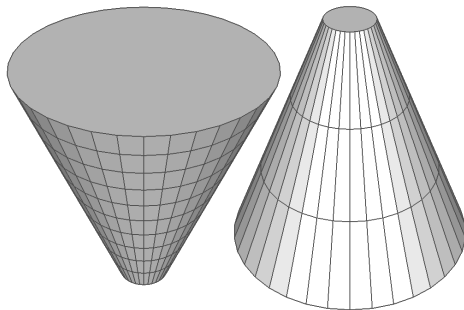
Entidades cuádricas en OpenGL

- Los comandos para dibujar las entidades cuádricas son:
 - `gluSphere(q, radius, slices, stacks);`
 - `radius` es `GLdouble`.
 - `slices`= n° de meridianos, es `int`.
 - `stacks`= n° de paralelos, es `int`.



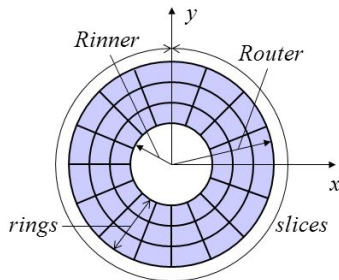
Entidades cuádricas en OpenGL

- Los comandos para dibujar las entidades cuádricas son:
 - `gluCylinder(q, baseRadius, topRadius, height, slices, stacks);`
 - Se construyen sobre el eje Z
 - `slices` = nº de lados y `stacks` = nº de rodajas
 - `baseRadius`, `topRadius`, `height` son `GLdouble`, y `slices`, `stacks` son `int`
 - Cuando cualquiera de los radios es 0 se obtienen conos
 - `baseRadius` es el radio del cilindro en $z = 0$ y `topRadius` es el radio del cilindro en $z = height$



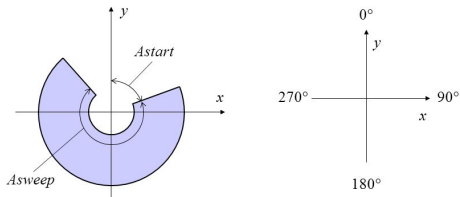
Entidades cuádricas en OpenGL

- Los comandos para dibujar las entidades cuádricas son:
 - `gluDisk(q, innerRadius, outerRadius, slices, rings);`
 - Se construyen en el plano XY
 - `slices`= nº de lados y `stacks`= nº de anillos
 - `innerRadius`, `outerRadius` son `GLdouble` y `slices` y `rings` son `int`



Entidades cuádricas en OpenGL

- Los comandos para dibujar las entidades cuádricas son:
 - `gluPartialDisk(q, innerRadius, outerRadius, slices, rings, startAngle, sweepAngle);`
 - Se construyen en el plano XY
 - `innerRadius`, `outerRadius` son `GLdouble`; `slices` y `rings` son `int`, y `startAngle`, `sweepAngle` son ángulos en grados
 - Los ángulos se miden en sentido horario, mirando desde la parte positiva del eje Z, sobre el plano XY, empezando en el eje Y



Entidades cuádricas

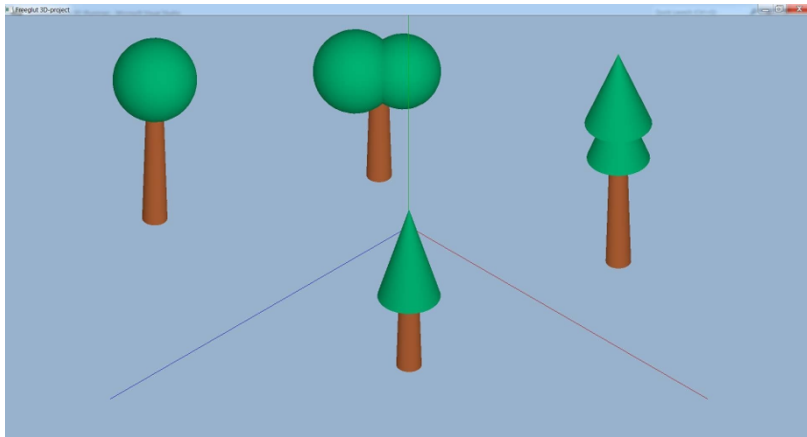
- Los comandos para especificar el modo en que se dibujan las entidades cuádricas son:

```
gluQuadricDrawStyle(q, tipoDeRecubrimiento);
```

- Los tipos de recubrimiento son:
 - **GLU_POINT**: Solamente se muestran los puntos del armazón del objeto cuádrico.
 - **GLU_LINE**: Solamente se muestran las líneas del armazón del objeto cuádrico.
 - **GLU_FILL**: Rellena cada cara del armazón del objeto cuádrico, teniendo en cuenta la iluminación.

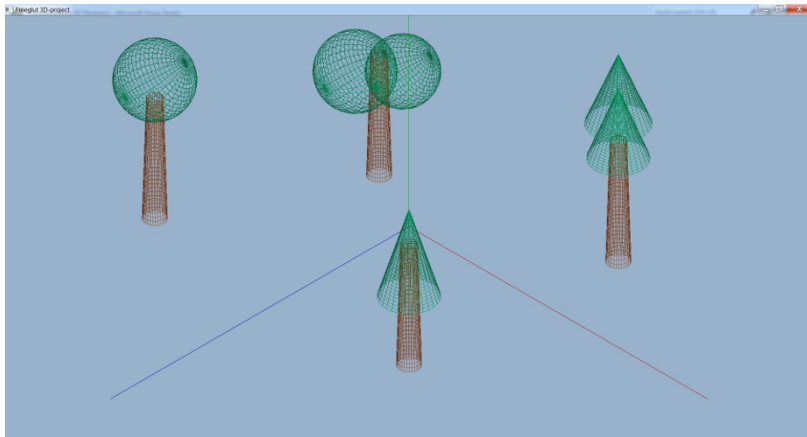
Entidades cuádricas

- `gluQuadricDrawStyle(q, GLU_FILL);`



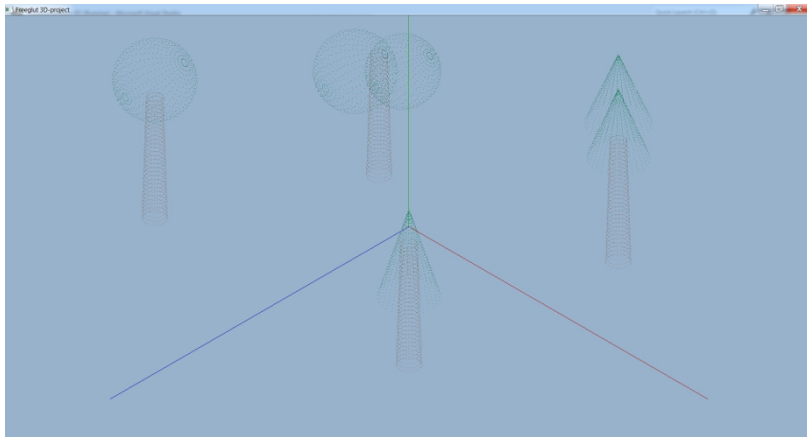
Entidades cuádricas

- `gluQuadricDrawStyle(q, GLU_LINE);`



Entidades cuádricas

- `gluQuadricDrawStyle(q, GLU_POINT);`



Entidades cuádricas en el proyecto

- La clase `QuadricEntity`:

```
class QuadricEntity : public Abs_Entity {  
public:  
    QuadricEntity();  
    ~QuadricEntity() { gluDeleteQuadric(q); };  
protected:  
    GLUquadricObj* q;  
};  
  
QuadricEntity::QuadricEntity() {  
    q = gluNewQuadric();  
}
```

Entidades cuádricas en el proyecto

- La clase `Sphere`:

```
class Sphere : public QuadricEntity {  
public:  
    Sphere(GLdouble r);    // r es el radio de la esfera  
    void render(glm::dmat4 const& modelViewMat) const;  
protected:  
    GLdouble r;  
};
```

- Análogamente se definen las clases `Cylinder`, `Disk`, `PartialDisk`

Entidades cuádricas en el proyecto

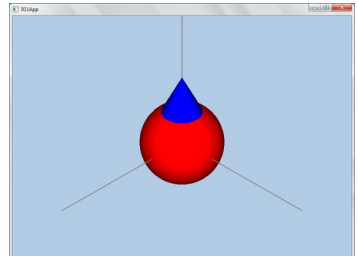
- La clase `Sphere`:

```
Sphere::Sphere(GLdouble rr) { r = rr; }  
void Sphere::render(glm::dmat4 const& modelViewMat) const {  
    dmat4 aMat = modelViewMat * mModelMat;  
    upload(aMat);  
    // Aquí se puede fijar el color de la esfera así:  
    // glEnable(GL_COLOR_MATERIAL);  
    // glColor3f(...);  
    // Aquí se puede fijar el modo de dibujar la esfera:  
    // gluQuadricDrawStyle(q, ...);  
    gluSphere(q, r, 50, 50);  
  
    // Aquí se debe recuperar el color:  
    // glColor3f(1.0, 1.0, 1.0);  
}
```

Entidades cuádricas en el proyecto: Ejemplo

```
Sphere* esfera = new Sphere(100.0);  
gObjects.push_back(esfera);
```

```
Cylinder* cono = new Cylinder(50.0, 0, 100.0);  
glm::dmat4 mAux = cono->modelMat();  
mAux = translate(mAux, dvec3(0, 85, 0));  
mAux = rotate(mAux, radians(-90.0), dvec3(1.0, 0, 0));  
cono->setModelMat(mAux);  
gObjects.push_back(cono);
```



Aplicación de texturas a entidades cuádricas

- Las cuádricas de GLU admiten que se adosen texturas sobre ellas
- Para poner una textura a una cuádrica de GLU, después de crearla, se debe activar la propiedad de textura con el comando:

```
// GLUquadricObj* q;  
gluQuadricTexture(q, GL_TRUE);
```

- Como con el resto de entidades, antes de dibujar una cuádrica con textura es necesario activar el objeto de textura. GLU dibujará entonces la cuádrica asociando a sus vértices, además de los vectores normales, las coordenadas de textura