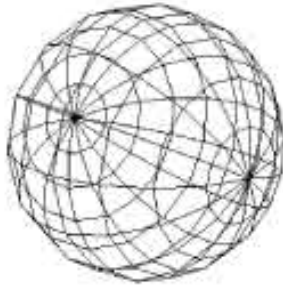
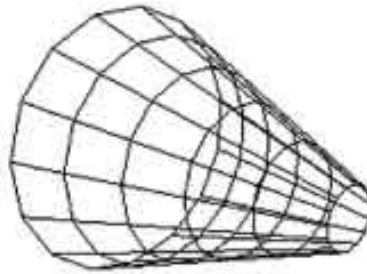


# Entidades cuádricas

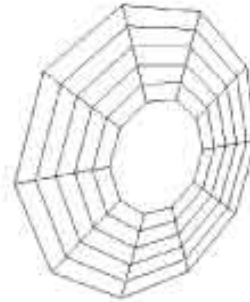
A. Gavilanes  
Departamento de Sistemas Informáticos y Computación  
Facultad de Informática  
Universidad Complutense de Madrid



(a)



(b)

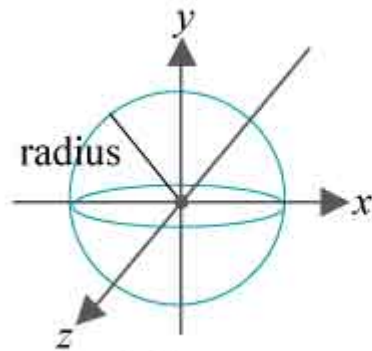


(c)

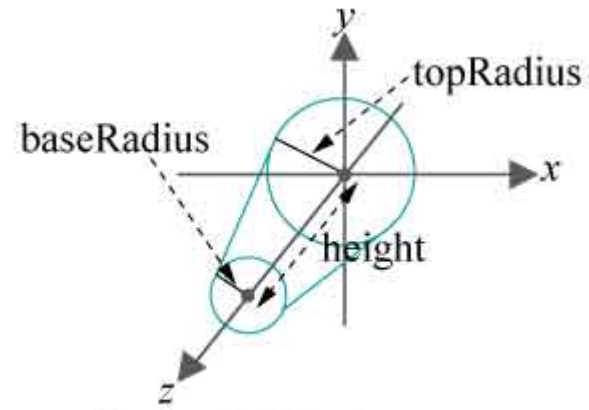


(d)

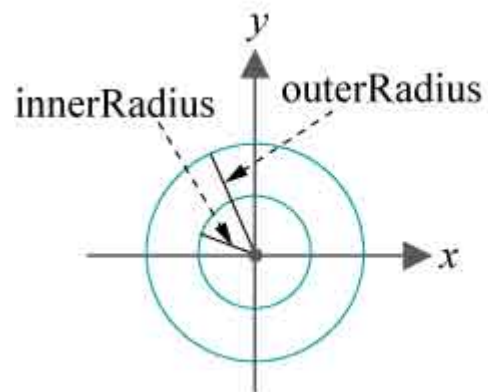
- ❑ Se proporcionan con la librería GLU.
- ❑ Se pueden crear cuatro tipos de entidades: (a) esferas, (b) cilindros, (c) discos y (d) discos parciales.
- ❑ Las entidades se declaran así: **GLUquadricObj\* q;**
- ❑ Las entidades se construyen así: **q=gluNewQuadric();**
- ❑ Las entidades se destruyen así: **gluDeleteQuadric(q);**



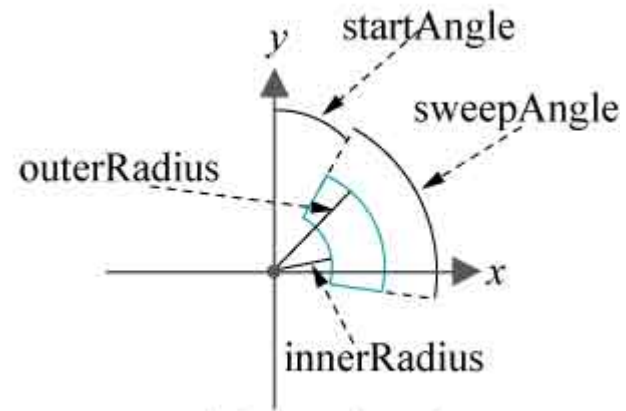
Sphere



Tapered Cylinder



Annular Disc



Partial Annular Disc

- ❑ Los comandos para dibujar las entidades cuádricas son:
  - ❑ **`gluSphere(q, radius, slices, stacks);`**
    - ❑ `slices`=nº de meridianos; `stacks`=nº de paralelos
    - ❑ `radius` es `GLdouble`, y `slices`, `stacks` son `int`
  - ❑ **`gluCylinder(q, baseRadius, topRadius, height, slices, stacks);`**
    - ❑ Se construyen sobre el eje Z
    - ❑ `slices`=nº de lados, `stacks`=nº de rodajas
    - ❑ `baseRadius`, `topRadius`, `height` son `GLdouble`, y `slices`, `stacks` son `int`
    - ❑ Cuando cualquiera de los radios es 0 se obtienen conos

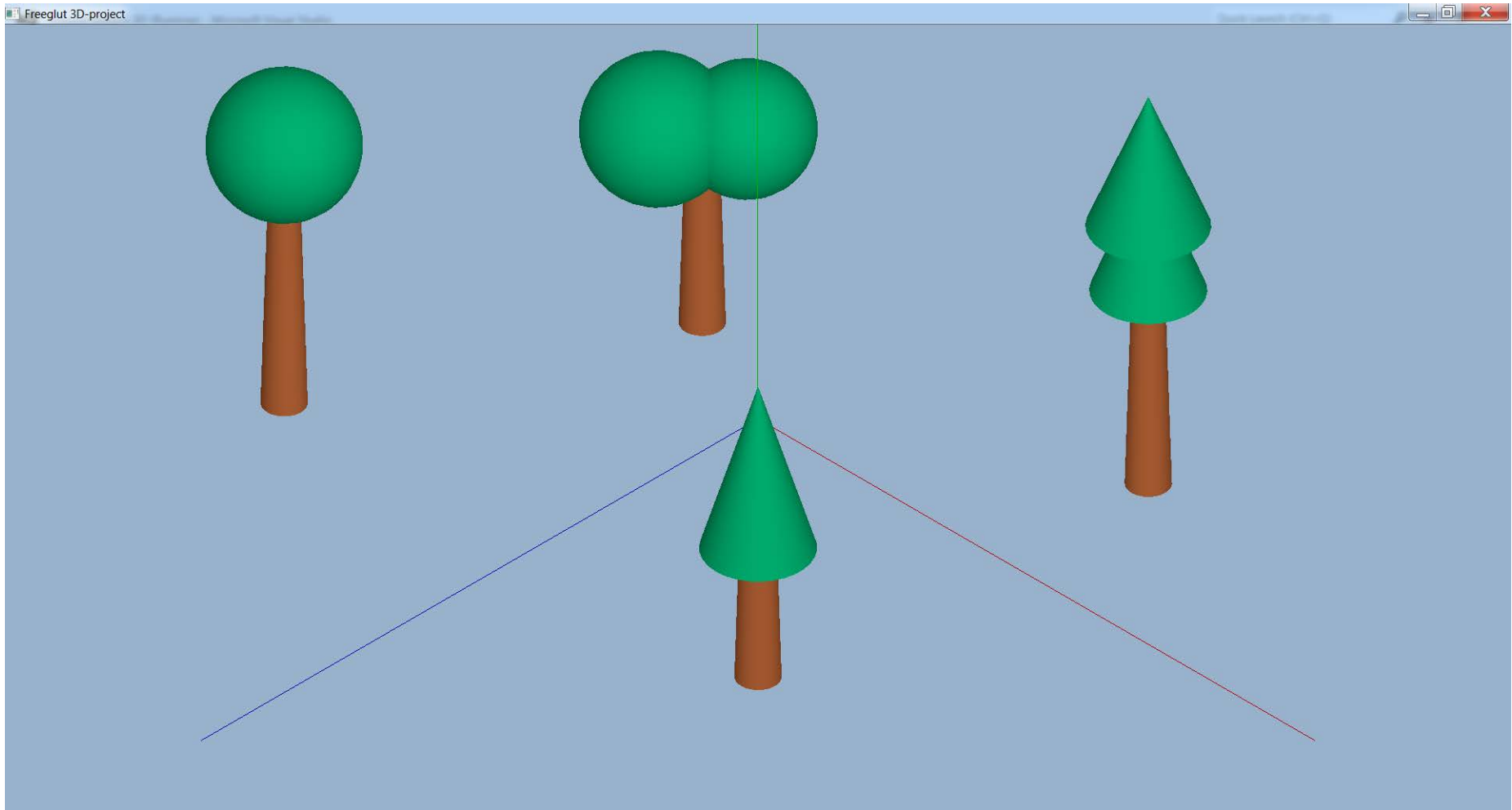
- ❑ `gluDisk(q, innerRadius, outerRadius, slices, rings);`
  - ❑ `slices`=nº de lados, `rings`=nº de anillos
  - ❑ `innerRadius`, `outerRadius` son `Gldouble`, y `slices`, `rings` son `int`
- ❑ `gluPartialDisk(q, innerRadius, outerRadius, slices, rings, startAngle, sweepAngle);`
  - ❑ `innerRadius`, `outerRadius` son `Gldouble`; `slices`, `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.

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

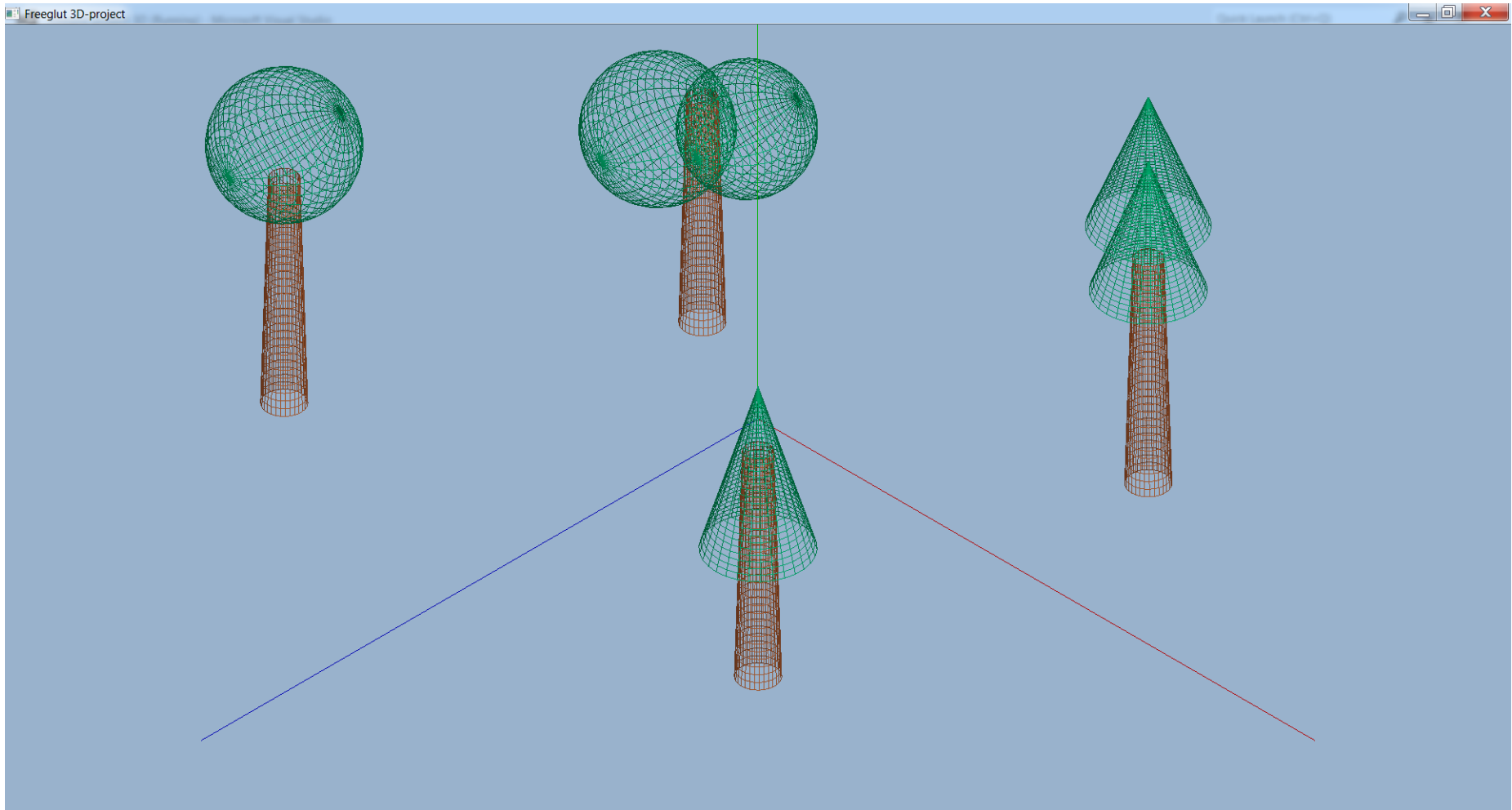
**`gluQuadricDrawStyle(q, tipoDeRecubrimiento);`**

- ❑ 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

❏ `gluQuadricDrawStyle(q, GLU_FILL);`

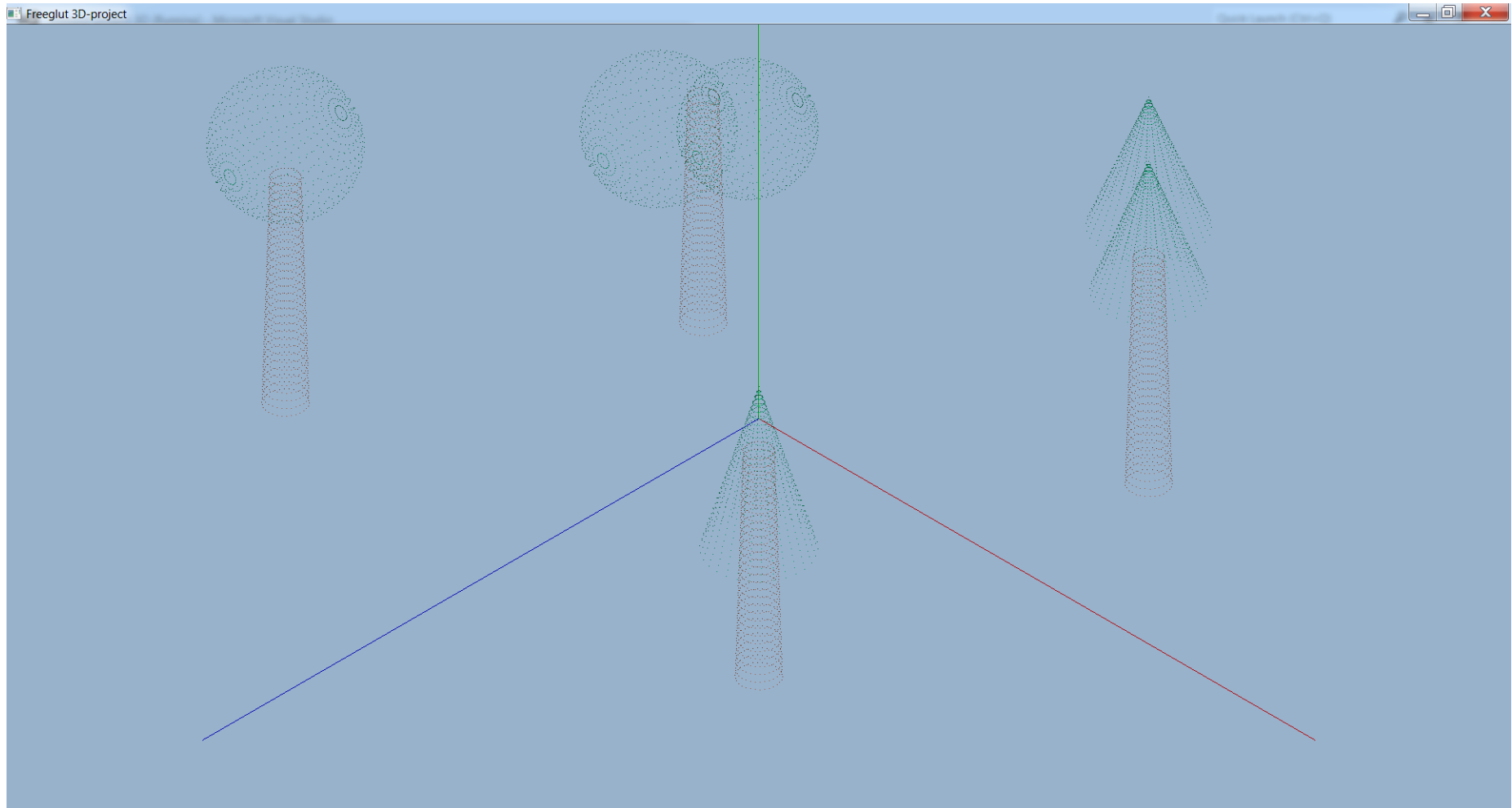


❏ `gluQuadricDrawStyle(q, GLU_LINE);`





❏ `gluQuadricDrawStyle(q, GLU_POINT);`



## ❑ La clase **QuadricEntity.h**

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

## ❑ La clase **QuadricEntity.cpp**

```
QuadricEntity::QuadricEntity() {  
    q = gluNewQuadric();  
}
```

## ❑ La clase **Sphere.h**

```
class Sphere : public QuadricEntity {  
    public:  
        Sphere(GLdouble r); // r is the radius of the sphere  
        void render(glm::dmat4 const& modelViewMat);  
    protected:  
        GLdouble r;  
};
```

## ❑ Análogamente las clases **Cylinder.h**, **Disk.h**, **PartialDisk.h**

## ❑ La clase `Sphere.cpp`

```
Sphere::Sphere(GLdouble rr) {  
    r = rr;  
}  
  
void Sphere::render(glm::dmat4 const& modelViewMat) {  
    uploadMvM(modelViewMat);  
    // Fijar el color con glColor3f(...);  
    // Fijar el modo en que se dibuja la entidad con  
    // gluQuadricDrawStyle(q, ...);  
    gluSphere(q, r, 50, 50);  
}
```