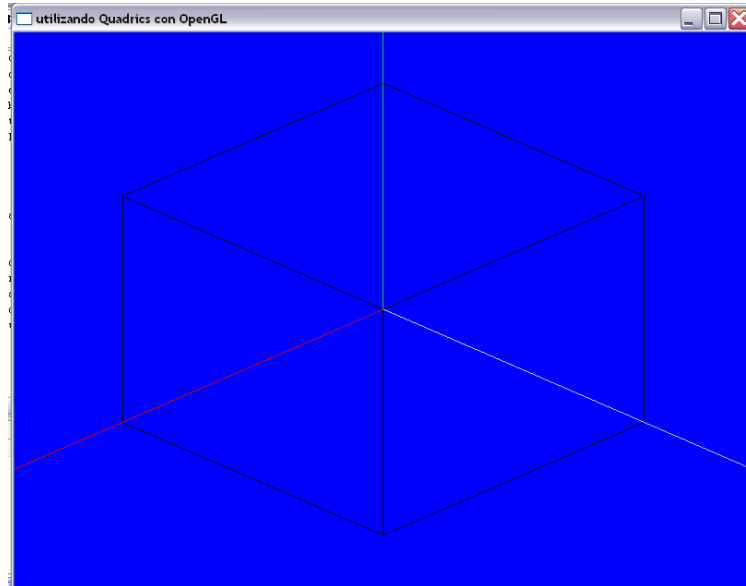


**Práctica2:** Modelado

**Objetivo:** El alumno aprenderá a utilizar la jerarquía para el uso de objetos en el escenario.

**Paso 1.** Utilizar “base.cpp”, cambiarle de nombre: “jerarquia.cpp”. Correr nuevamente programa.

**Paso 2.** Editar programa. Este programa debe mostrar dos funciones: Una, eje de referencia. Dos, cubo. Demás funciones de líneas y triángulos se deberán borrar ó comentar.



**Paso 3.** Trabajar con programa “jerarquia.cpp”.

- I. Crear función llamada jerarquia() { procesos}. Antes de función: renderscene().
- II. Ubicar llamado de función jerarquia(); dentro de función renderscene() antes de glFlush();
- III. En la parte de “procesos” de la función nueva jerarquia():

**Nota:** El código agregado se encuentra en negritas

```
void jerarquia()
{

}
void RenderScene(void)
{

    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    luces();
    EjesReferencia();
    cuboreferencia();
    materiales();
jerarquia();

    glFlush();

}
```

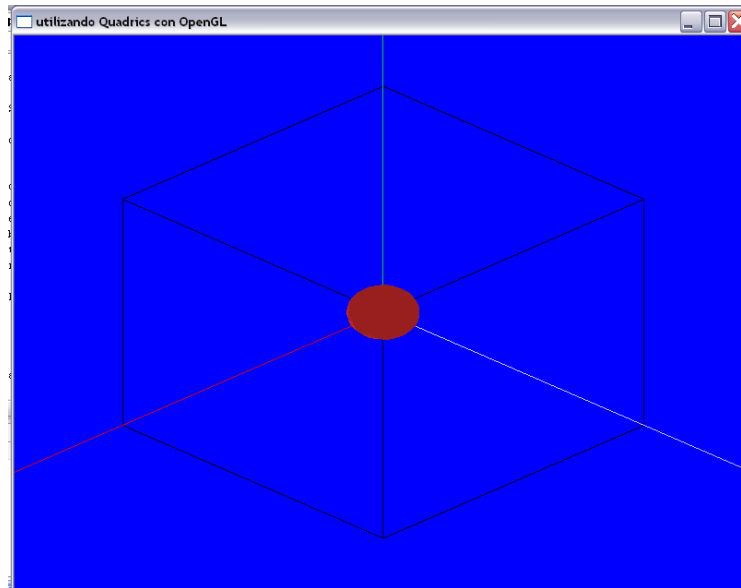
**Práctica2: Modelado**

---

**Preparación.****1. Dibujar esfera.**

Código agregado:

```
void jerarquia()
{
    glutSolidSphere(0.5f,15,15);
}
```

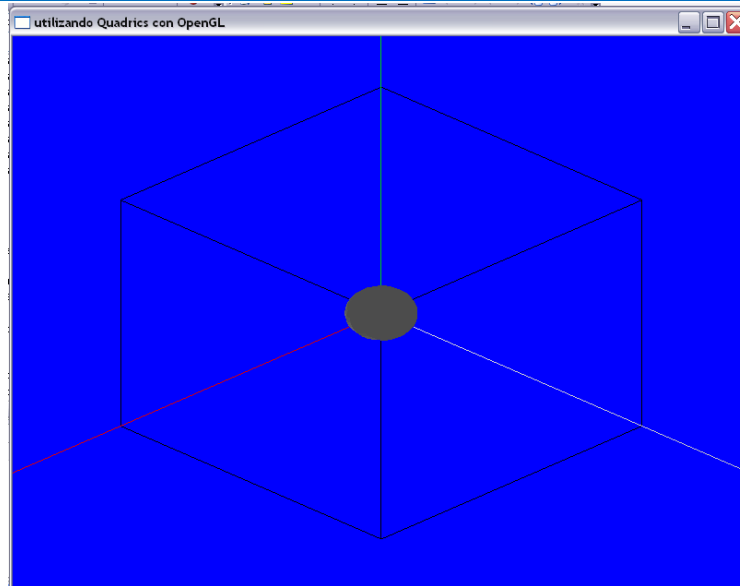
**2. Asignar color. cambiar de color. Asigne a esfera un color gris.**

Código agregado:

**GLfloat gris [4] = {0.5, 0.5, 0.5, 0.5 };**//Código agregado donde se declaran los colores de los materiales.

```
void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glutSolidSphere(0.5f,15,15);
}
```

## Práctica2: Modelado

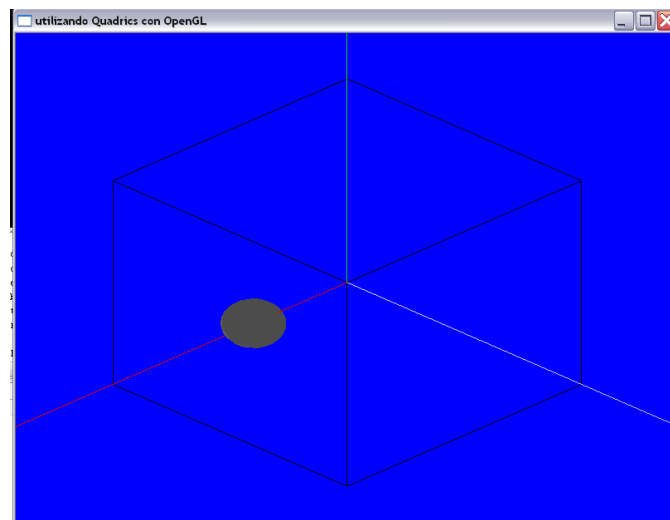
**3. Transformaciones de INICIO.**

- Traslade la esfera 2 unidades en Z.
- Rote 45 grados en Y. (Si no se observa cambio. Primero rote y luego traslade.)

a)

Código agregadi

```
void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glTranslatef(2,0,0);
    glutSolidSphere(0.5f,15,15);
}
```



**Práctica2: Modelado**

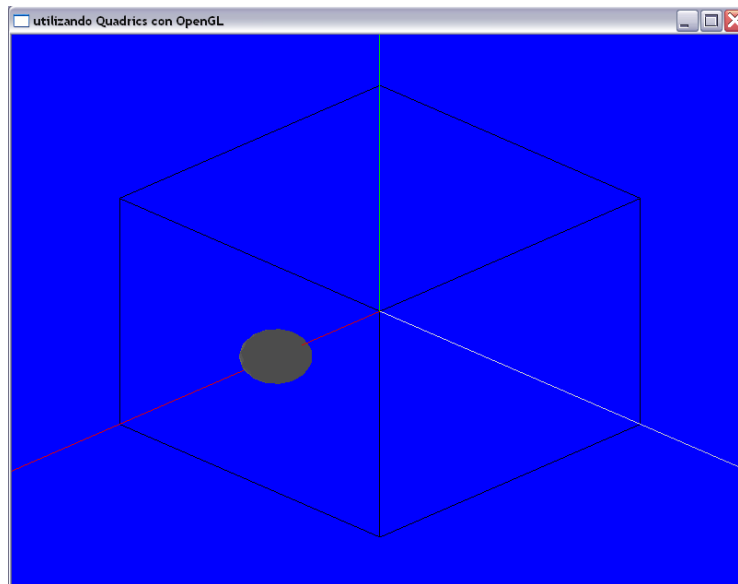
b)

Código Agregado:

```

void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glTranslatef(2,0,0);
    glRotatef(45,0,1,0);
    glutSolidSphere(0.5f,15,15);
}

```



Nota: Aunque se realizó la rotación no se alcanza a observar por que la rotación realizada es en una esfera.

**Modelado Jerárquico 1.****4. Preparar 2 esferas.**

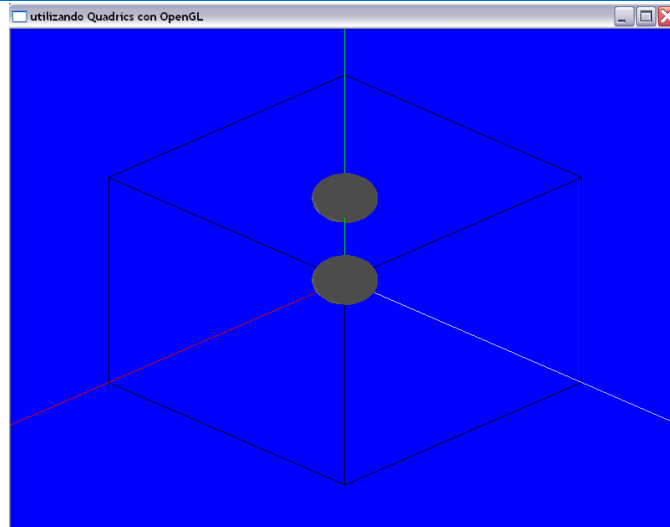
- Dibuje Esfera 1 en origen de coordenadas (0, 0, 0)
- Dibuje Esfera 2 en posición 2 unidades más, desplazadas en Y.

Código Agregado:

```

void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    //glTranslatef(2,0,0);
    //glRotatef(45,0,1,0);
    //glutSolidSphere(0.5f,15,15);
    glutSolidSphere(0.5f,15,15);
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f,15,15);
}

```

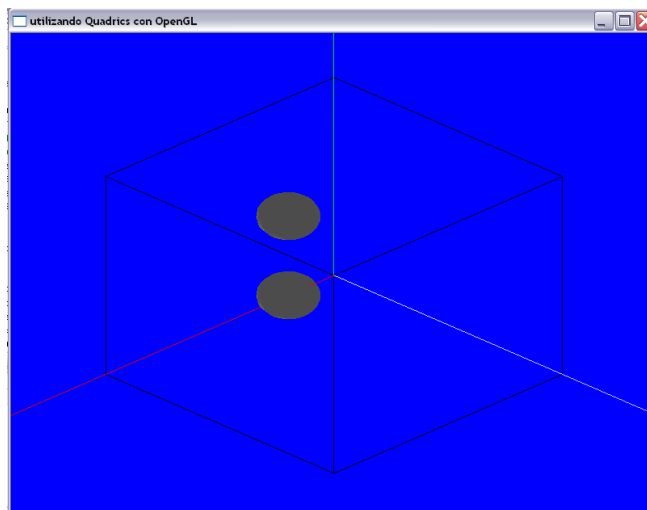
**Práctica2: Modelado****5. PILA GENERAL.**

Se tiene un trabajo de matrices interno dentro de los procesos de OPENGL. Las transformaciones se van adicionando a las transformaciones ya hechas anteriormente.

- a. Traslade la esfera 1 dos unidades en X. Observe y anote que pasa. Regrésela a su posición.

Código Agregado:

```
void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    //glTranslatef(2,0,0);
    //glRotatef(45,0,1,0);
    //glutSolidSphere(0.5f,15,15);
    glTranslatef(1,0,0);
    glutSolidSphere(0.5f,15,15);
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f,15,15);
}
```



**Práctica2: Modelado**

Cuando se dio la instrucción de trasladar la primera esfera también traslado la segunda.

En este caso:¿ la esfera 1 es Padre o hijo? ¿Y La esfera 2?

La esfera 1 es **padre**.

La esfera 2 es **hijo**.

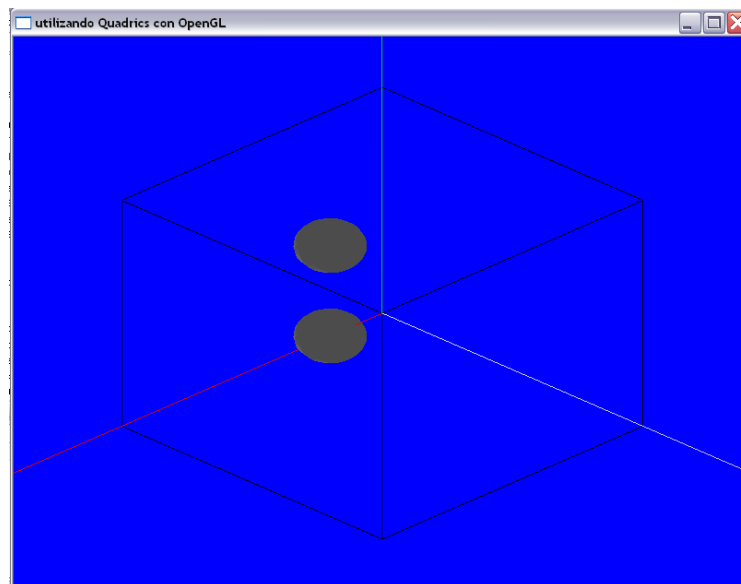
**6. PILA Push-Pop. Manejo de Pilas**

a. General Coloque los nuevos comandos: Push y Pop entre las transformaciones y la esfera.

Código Agregado:

```
void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);

    glPushMatrix();
    glTranslatef(2,0,0);
    glutSolidSphere(0.5f,15,15);
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f,15,15);
    glPopMatrix();
}
```



Al correrlo el funcionamiento siguió siendo el mismo.

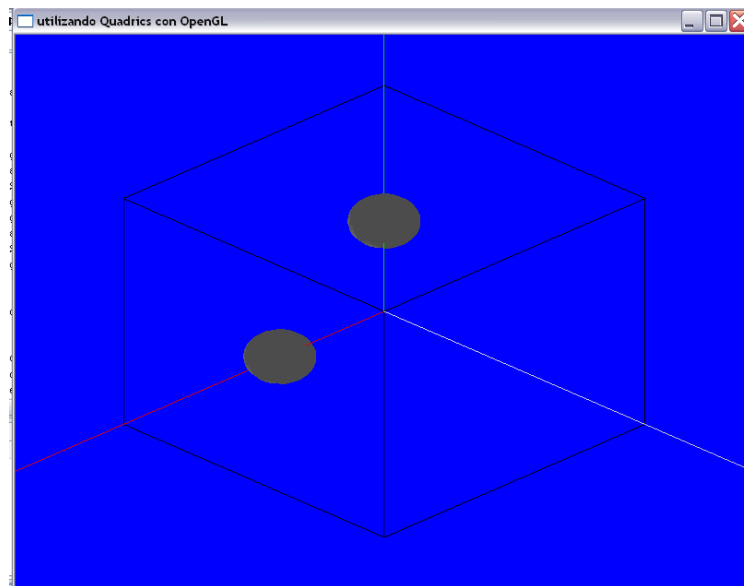
**Práctica2: Modelado**

- b. Particular. Coloque el par: Push y Pop para cada esfera.

Código Agregado:

```
void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);

    glPushMatrix();
    glTranslatef(2,0,0);
    glutSolidSphere(0.5f,15,15);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f,15,15);
    glPopMatrix();
}
```



Ahora cada esfera se queda en la posición indicada.

## 7. Regresando a coordenadas de origen con “LoadIdentity()”.

- a. Quite push y pop y solo coloque LoadIdentity entre cada objeto antes de sus transformaciones.

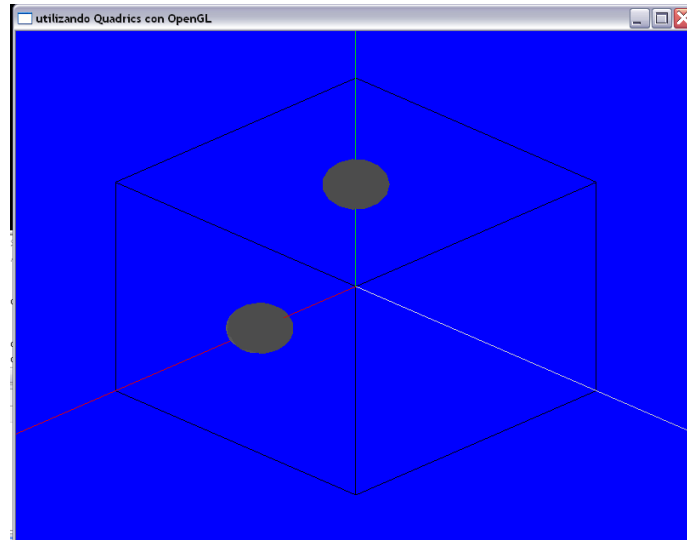
Código agregado:

```
void jerarquia()
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);

    //glPushMatrix();
    glTranslatef(2,0,0);
    glutSolidSphere(0.5f,15,15);
    //glPopMatrix();
}
```

## Práctica2: Modelado

```
//glPopMatrix();  
glLoadIdentity();  
//glPushMatrix();  
glTranslatef(0,2,0);  
glutSolidSphere(0.5f,15,15);  
//glPopMatrix();  
  
}
```



Las esferas siguen tomando las instrucciones indicadas

Esqueleto que se obtiene con el código de muestra:

