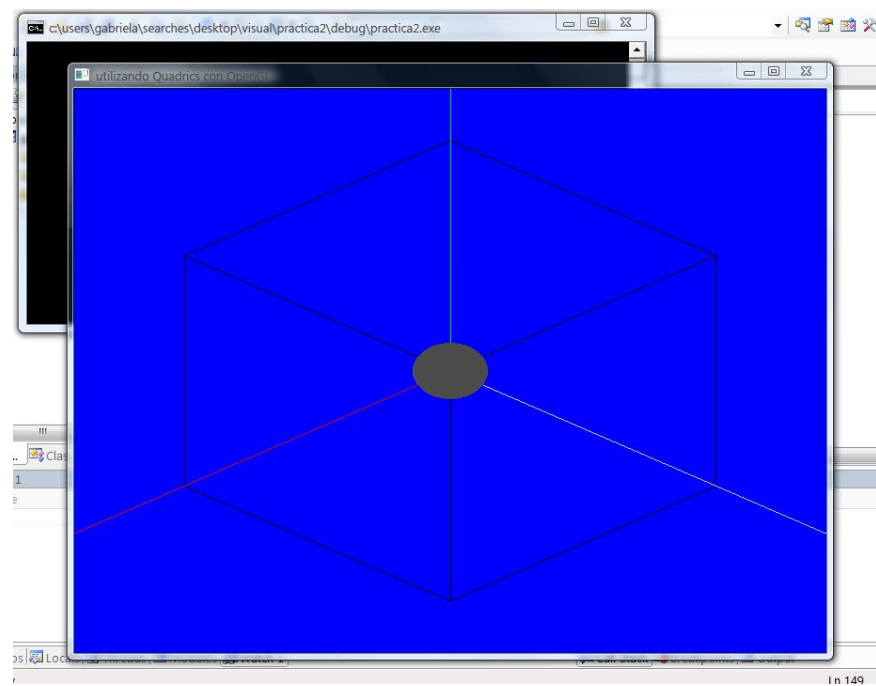
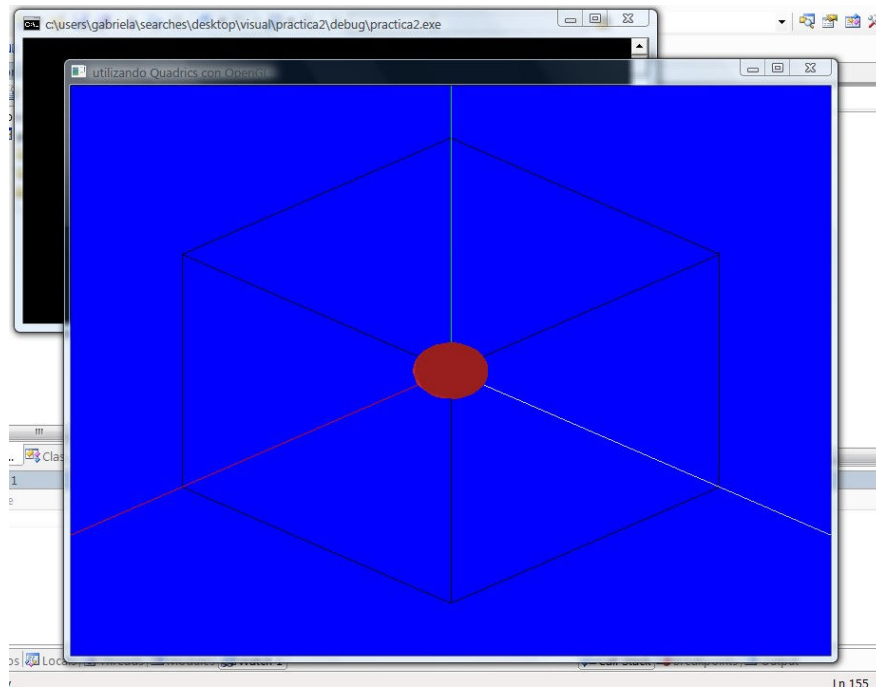


PRACTICA N.-2

MODELADO JERARQUICO

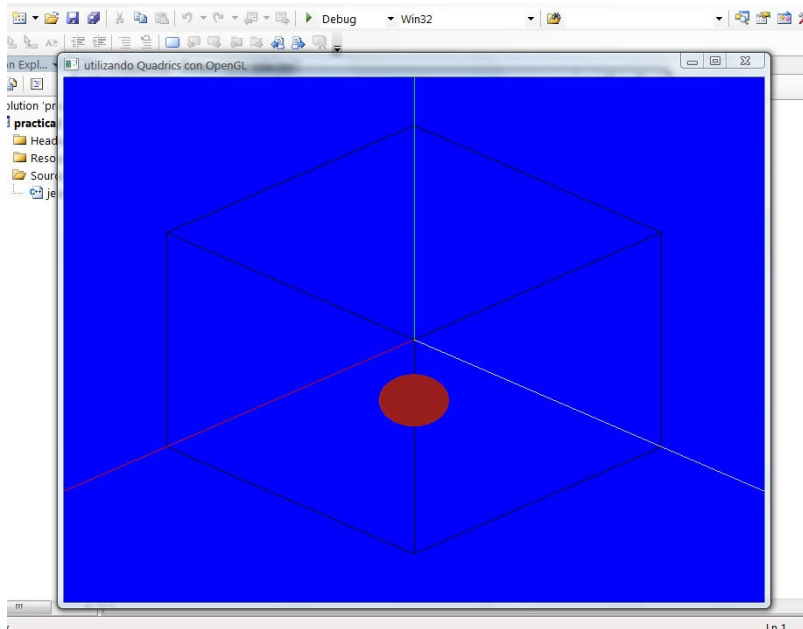
1.-Dibujar Esfera.

2.-Asignarle un color diferente.



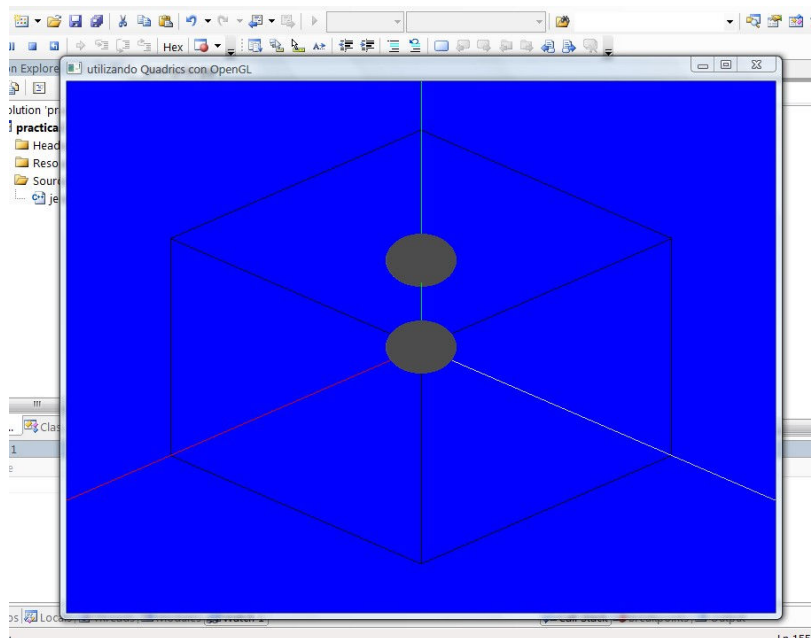
3.-Transformaciones de Inicio. (Rotación y translación)

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



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.



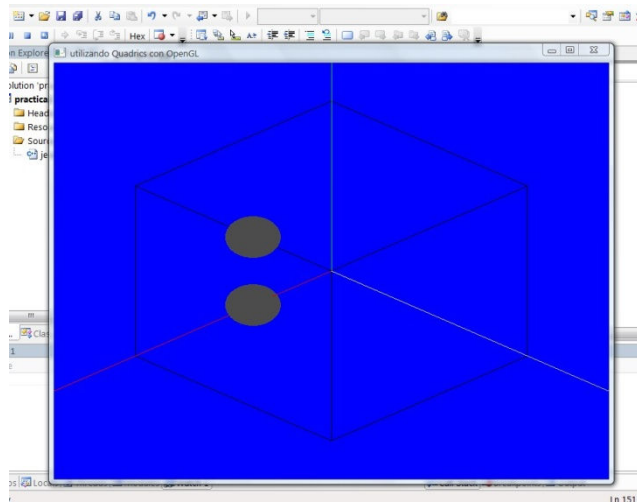
5. PILA GENERAL.

- Traslade la esfera 1 dos unidades en X. Observe y anote que pasa. Regrésela a su posición.
- Traslade la esfera 2 dos unidades en X. Observe y anote que pasa. Regrésela a su posición.

El objeto que se mueve y mueve a otros con él, se llamará: “Padre ó Raíz” y los demás objetos que se mueven y no jalan a otros objetos se llaman “Hijos u Hojas”.

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

Respuesta: La esfera uno es Padre y la esfera 2 es hijo



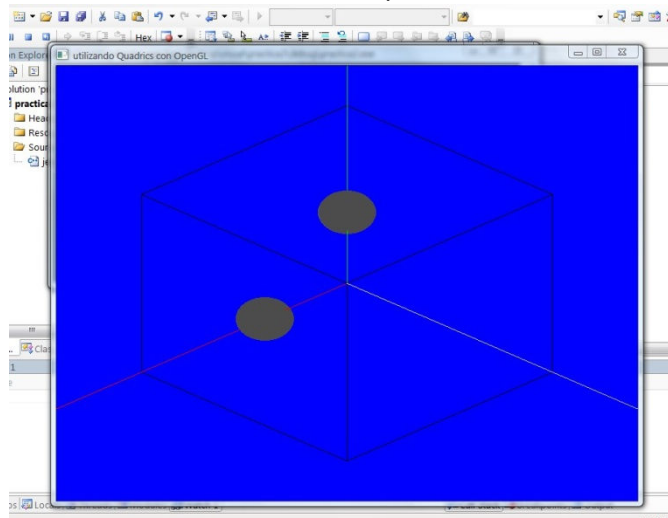
6. PILA Push-Pop. Manejo de Pilas

- General Coloque los nuevos comandos: Push y Pop entre las transformaciones y la esfera. ¿Hubo cambio o no?

Respuesta: No hubo cambio en la ejecución del programa

- Particular. Coloque el par: Push y Pop para cada esfera. ¿Hubo cambio o no?

Respuesta: Cada proceso se realiza de manera independiente



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

a. Quite push y pop y solo coloque LoadIdentity entre cada objeto antes de sus transformaciones
¿Hubo cambio o no, que pasó?

Respuesta: Sucedió lo mismo que cuando se puso push y pop antes de cada esfera

Conclusión: Push y pop y LoadIdentity regresan coordenadas al origen.

```

void jerarquia()
{
    /*glRotatef(45,0,1,0);
    glTranslatef(0,0,2);
    glutSolidSphere(0.5f,45,45);*/

    //glPushMatrix();
    glTranslatef(2,0,0);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glutSolidSphere(0.5f,80,80); //Esfera1
    //glPopMatrix();

    glLoadIdentity();

    //glPushMatrix();
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f,80,80); //Esfera2
    //glPopMatrix();
}

void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    luces();
    EjesReferencia();
    cuboreferencia();
    //triangulo();
    materiales();
    cilindro();
}
    
```

Conclusión:

Esta práctica fue muy informativa en cuanto a las cuestiones de manejo de instrucciones en open gl, como por ejemplo poder llevar procesos de manera paralela sin que uno afecte al otro, o en ocasiones tomar un proceso como “proceso-padre” y si realizamos algún cambio sobre él, las demás funciones que se encuentren debajo de esta también serán modificadas. Por lo que las instrucciones Push y Pop serán de mucha ayuda en nuestras futuras practicas.