

52. Supón que dispones de dos objetos simples: Semiesfera y Cilindro, con su correspondiente función dibujar(). La semiesfera (en coordenadas maestras) tiene radio unidad, centro en el origen y el eje vertical en el eje Y. Igualmente el cilindro tiene radio y altura unidad, el centro de la base está en el origen, y su eje es el eje Y. Con estas dos primitivas queremos construir la figura símbolo de Android
- - Diseña el grafo de escena correspondiente, ten en cuenta que hay objetos compuestos que se pueden instanciar más de una vez (cada brazo o pierna se puede construir con un objeto compuesto de dos semiesferas en los extremos de un cilindro).
 - - Escribe el código OpenGL para visualizarlo, usando transformaciones y push/pop de la matriz modelview.

```
glPushMatrix();
    glScalef(50,50,50);
    cilindro->draw(modo_diferido,modo,conTapa); // torso
    glPushMatrix(); //cabeza del muñeco
        glTranslatef(0, 0.85, 0); //altura cabeza
        semiesfera->draw(modo_diferido,modo,conTapa); // cabeza
        glPushMatrix(); //ojos
            glTranslatef(0,0,0.4); //eje z de los ojos
            glPushMatrix(); //ojo derecho
                glTranslatef(0.2,0,0);
                glRotatef(45,1,0,0);
                glRotatef(-25,0,0,1);
                glScalef(0.1,0.1,0.1);
                semiesfera_negra->draw(modo_diferido,modo,conTapa);
            glPopMatrix();
            glPushMatrix(); //ojo izquierdo
                glTranslatef(-0.2,0,0);
                glRotatef(45,1,0,0);
                glRotatef(25,0,0,1);
                glScalef(0.1,0.1,0.1);
                semiesfera_negra->draw(modo_diferido,modo,conTapa);

        glPopMatrix();
    glPopMatrix(); //fin ojos
    glPushMatrix(); //antenas
        glTranslatef(0,0.2,0); //eje x de las antenas, la subimos un poco
        glPushMatrix(); //antena derecha
            glTranslatef(0.2,0,0);
            glRotatef(-35,0,0,1);
            //glRotatef(-25,0,0,1);
            glScalef(0.1,0.5,0.1);
            cilindro->draw(modo_diferido,modo,conTapa);
        glPopMatrix();
        glPushMatrix(); //antena izda
            glTranslatef(-0.2,0,0);
```

```

    glRotatef(35,0,0,1);
    //glRotatef(-25,0,0,1);
    glScalef(0.1,0.5,0.1);
    cilindro->draw(modos_diferido,modos,conTapa);
    glPopMatrix();
glPopMatrix();//fin antenas
glPopMatrix();//fin cabeza
glPushMatrix();// brazo izquierdo
    glTranslatef(-0.75, 0 , 0); //mover a la x del brazo izquierdo
    glPushMatrix(); //hombro
        glTranslatef(0,0.564,0);//regulamos la altura del hombro, contamos con que se va
a hacer más pequeña la esfera
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modos,conTapa); //hombro izquierdo
    glPopMatrix(); //fin hombro
    glPushMatrix(); //brazo
        glScalef(0.25,1,0.25);
        cilindro->draw(modos_diferido,modos,conTapa);//brazo
    glPopMatrix(); //fin brazo

    glPushMatrix(); //mano
        glTranslatef(0,-0.564, 0);
        glRotatef(180, 1, 0,0);
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modos,conTapa); //mano
    glPopMatrix(); //fin mano
glPopMatrix();//fin brazo izquierdo
glPushMatrix();// brazo derecho
    glTranslatef(0.75, 0 , 0); //mover a la x del brazo derecho
    glPushMatrix(); //hombro
        glTranslatef(0,0.5,0);//regulamos la altura del hombro, contamos con que se
va a hacer más pequeña la esfera
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modos,conTapa); //hombro derecho
    glPopMatrix(); //fin hombro
    glPushMatrix(); //brazo
        glScalef(0.25,1,0.25);
        cilindro->draw(modos_diferido,modos,conTapa);//brazo
    glPopMatrix(); //fin brazo
    glPushMatrix(); //mano
        glTranslatef(0,-0.5, 0);
        glRotatef(180, 1, 0,0);
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modos,conTapa); //mano
    glPopMatrix(); //fin mano
glPopMatrix();//fin brazo derecho
glPushMatrix();//pierna derecha
    glTranslatef(0.2,0,0); //mover a la x de la pierna
    glPushMatrix();//pierna

```

```

        glTranslatef(0,-0.7,0);
        glScalef(0.25,0.35,0.25);
        cilindro->draw(modos_diferido,modo,conTapa);
        glPopMatrix();//fin pierna

glPushMatrix();
    glRotatef(180,1,0,0);
    glTranslatef(0,1,0);
    glScalef(0.25,0.25,0.25);
    semiesfera->draw(modos_diferido,modo,conTapa);
    glPopMatrix();
glPopMatrix();//fin pierna derecha
glPushMatrix();//pierna izquierda
    glTranslatef(-0.2,0,0); //mover a la x de la pierna
    glPushMatrix();//pierna
        glTranslatef(0,-0.7,0);
        glScalef(0.25,0.35,0.25);
        cilindro->draw(modos_diferido,modo,conTapa);
    glPopMatrix();//fin pierna
    glPushMatrix();
        glRotatef(180,1,0,0);
        glTranslatef(0,1,0);
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modo,conTapa);
    glPopMatrix();
glPopMatrix();//fin pierna derecha
glPopMatrix();

```

53. Escribe una segunda versión del grafo de escena del ejercicio anterior, de forma que las transformaciones estén parametrizadas por dos valores reales (alpha y beta) que expresan el ángulo de rotación del brazo izquierdo y derecho (respectivamente), en torno al eje que pasa por los centros de las dos semiesferas superiores de los brazos. Asimismo, habrá otro parámetro (tita) que es el ángulo de rotación de la cabeza (completa: con los ojos y antenas) entorno al eje vertical que pasa por su centro (cuando estos ángulo valen 0, el androide está en reposo y tiene exactamente la forma de la figura del problema anterior). - Escribe el método `Androide::dibuja(float a, float b, float c)` para visualizar el androide parametrizado de esta forma. El método `Androide::dibuja()` debe aceptar como parámetros los tres ángulos.

```

void Androide::dibuja_androide(float a, float b, float c){
    modos_dibujado modos = SOLIDO;
    glPushMatrix();
        glScalef(50,50,50);
        cilindro->draw(modos_diferido,modo,conTapa);// torso
    glPushMatrix();//cabeza del muñeco
        glTranslatef(0, 0.85, 0); //altura cabeza
        glRotatef(a,0,1,0); //giro de la cabeza
    }
}

```

```

        semiesfera->draw(modo_diferido,modo,conTapa); // cabeza
glPushMatrix(); //ojos
glTranslatef(0,0,0.4); //eje z de los ojos
glPushMatrix(); //ojo izquierdo
    glTranslatef(0.2,0,0);
    glRotatef(45,1,0,0);
    glRotatef(-25,0,0,1);
    glScalef(0.1,0.1,0.1);
    semiesfera_negra->draw(modo_diferido,modo,conTapa);
glPopMatrix();
glPushMatrix(); //ojo derecho
    glTranslatef(-0.2,0,0);
    glRotatef(45,1,0,0);
    glRotatef(25,0,0,1);
    glScalef(0.1,0.1,0.1);
    semiesfera_negra->draw(modo_diferido,modo,conTapa);
glPopMatrix();
glPopMatrix(); //fin ojos
glPushMatrix(); //antenas
    glTranslatef(0,0.2,0); //eje x de las antenas, la subimos un poco
    glPushMatrix(); //antena izda
        glTranslatef(0.2,0,0);
        glRotatef(-35,0,0,1);
        //glRotatef(-25,0,0,1);
        glScalef(0.1,0.5,0.1);
cilindro->draw(modo_diferido,modo,conTapa);
    glPopMatrix();
    glPushMatrix(); //antena dcha
        glTranslatef(-0.2,0,0);
        glRotatef(35,0,0,1);
        //glRotatef(-25,0,0,1);
        glScalef(0.1,0.5,0.1);
        cilindro->draw(modo_diferido,modo,conTapa);
    glPopMatrix();
glPopMatrix(); //fin antenas
glPopMatrix(); //fin cabeza
glPushMatrix(); // brazo izquierdo
    glTranslatef(-0.75, 0 , 0); //mover a la x del brazo izquierdo
    glTranslatef(0,0.5,0); //hacemos que el punto de giro sea la altura del hombro
    glRotatef(b,1,0,0);
    glTranslatef(0,-0.5,0); //devolvemos el brazo a su altura
    glPushMatrix(); //hombro
        glTranslatef(0,0.5,0); //regulamos la altura del hombro, contamos con que se
va a hacer más pequeña la esfera
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modo_diferido,modo,conTapa); //hombro izquierdo
    glPopMatrix(); //fin hombro
    glPushMatrix(); //brazo
        glScalef(0.25,1,0.25);

```

```

        cilindro->draw(modos_diferido,modo,conTapa);//brazo
glPopMatrix(); //fin brazo
glPushMatrix(); //mano
    glTranslatef(0,-0.5, 0);
    glRotatef(180, 1, 0,0);
    glScalef(0.25,0.25,0.25);
    semiesfera->draw(modos_diferido,modo,conTapa); //mano
glPopMatrix(); //fin mano

glPopMatrix(); //fin brazo izquierdo
glPushMatrix();// brazo derecho
    glTranslatef(0.75, 0 , 0); //mover a la x del brazo derecho
    glTranslatef(0,0.5,0);//hacemos que el punto de giro sea la altura del hombro
    glRotatef(c,1,0,0);
    glTranslatef(0,-0.5,0); //devolvemos el brazo a su altura
    glPushMatrix(); //hombro
        glTranslatef(0,0.5,0);//regulamos la altura del hombro, contamos con que se
va a hacer más pequeña la esfera
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modo,conTapa); //hombro derecho
    glPopMatrix(); //fin hombro
    glPushMatrix(); //brazo
        glScalef(0.25,1,0.25);
        cilindro->draw(modos_diferido,modo,conTapa);//brazo
    glPopMatrix(); //fin brazo
    glPushMatrix(); //mano
        glTranslatef(0,-0.5, 0);
        glRotatef(180, 1, 0,0);
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modo,conTapa); //mano
    glPopMatrix(); //fin mano
glPopMatrix(); //fin brazo derecho
glPushMatrix();//pierna derecha
    glTranslatef(0.2,0,0); //mover a la x de la pierna
    glPushMatrix();//pierna
        glTranslatef(0,-0.7,0);
        glScalef(0.25,0.35,0.25);
        cilindro->draw(modos_diferido,modo,conTapa);
    glPopMatrix();//fin pierna
    glPushMatrix();//pie
        glRotatef(180,1,0,0);
        glTranslatef(0,1,0);
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferido,modo,conTapa);
    glPopMatrix();//fin pie
glPopMatrix();//fin pierna derecha
glPushMatrix();//pierna izquierda
    glTranslatef(-0.2,0,0); //mover a la x de la pierna

```

```

    glPushMatrix();//pierna
        glTranslatef(0,-0.7,0);
        glScalef(0.25,0.35,0.25);
        cilindro->draw(modos_diferidos,modos,conTapa);
    glPopMatrix();//fin pierna
    glPushMatrix();//pie
        glRotatef(180,1,0,0);
        glTranslatef(0,1,0);
        glScalef(0.25,0.25,0.25);
        semiesfera->draw(modos_diferidos,modos,conTapa);
    glPopMatrix();//fin pie
    glPopMatrix();//fin pierna derecha
glPopMatrix();
}

```

Grafo Modelo Jerárquico del Android



