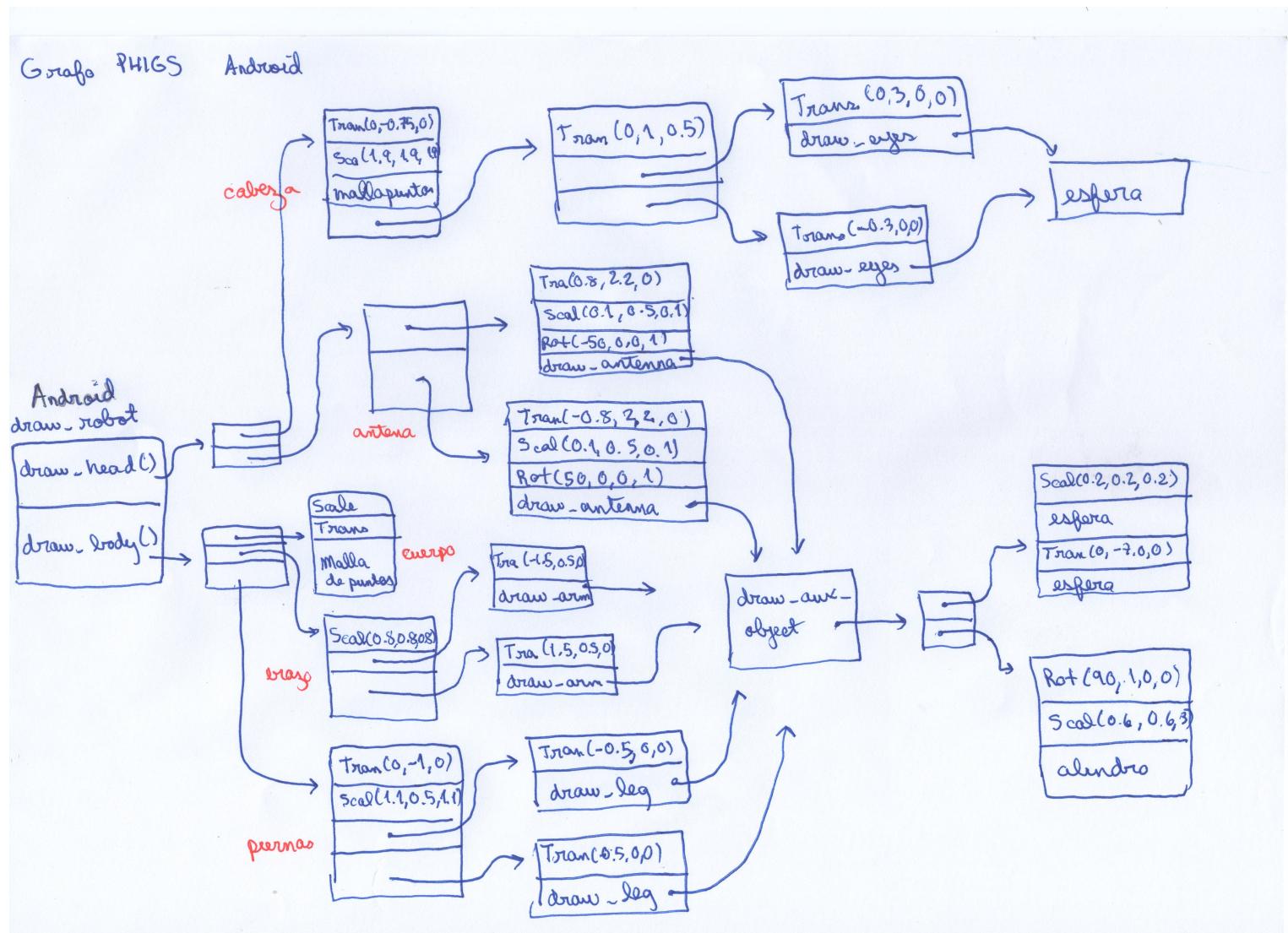


# Ejercicio 18

## Jose Miguel Colella

- Para la primera parte del ejercicio hay que diseñar el grafo PHIGS de dicho objeto. El grafo PHIGS que veremos en la siguiente imagen es el grafo basado en el código del segundo apartado. Pero basicamente se tiene un nodo raiz que hace dos cosas. Un metodo se encarga de hacer la cabeza, mientras el segundo metodo se encarga del cuerpo. Los brazos, piernas y antenas esta construido de la misma estructura, solo se diferencia en las transformaciones aplicadas. El cuerpo y la cabeza son malla de puntos construido de dos ficheros ply. Finalmente los ojos son esferas.



- El código que visualizar el Android es el siguiente. La construcción del mismo esta compuesta de objetos compuestos. Y esos objetos compuestos estan construidos de objetos basicos; mallas de puntos, cilindros, esferas. La función `draw_robot` llama a `draw_body` y a `draw_head` que gestionan la construcción del cuerpo y de la cabeza, y todos los objetos que pertenecen a ellos.

Primero se declar el fichero que contiene la declaración de la clase. Como podemos ver, el Android contiene dos MallaTVT, para contener el cuerpo y la cabeza.

```
#include <GL/gl.h>
#include <vector>
```

```

#include "vertex.h"
#include "visualtype.h" //Enum que denota que denota la visualizacion
#include "MallaTVT.h"

#ifndef _ANDROID_H_
#define _ANDROID_H_

class Android
{

private:
    visual_t visualization;
    MallaTVT * android_body;
    MallaTVT * android_head;
    //Metodos privados que dibujan objetos simples de glut
    void draw_cylinder();
    void draw_sphere(GLint slices, GLint stacks);
    void draw_cube();

public:
    Android();
    ~Android();
    //Metodo para dibujar el Android
    void draw(visual_t visualization);

private:
    //Metodos privados auxiliares para dibujar en Android entero
    void draw_aux_object();
    void draw_body();
    void draw_head();
    void draw_arm();
    void draw_leg();
    void draw_eyes();
    void draw_antenna();
    void draw_robot();

};

#endif

```

A continuación podemos ver la declaración de la clase.

```

#include "stdlib.h"
#include "stdio.h"
#include <GL/glut.h>
#include <ctype.h>
#include <iostream>
#include "Android.h"

using namespace std;

Android :: Android()
{
    visualization = POINT;
    android_body = new MallaTVT();
    android_head = new MallaTVT();
    android_body->initializeRotationalObject("android_body");
    android_head->initializeRotationalObject("android_head");
}

Android :: ~Android()
{
    delete android_head;
    delete android_body;
}

void Android :: draw_sphere(GLint slices, GLint stacks) {

```

```

switch(this->visualization) {
    case 1:
        glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
        glutSolidSphere(1.0,slices, stacks);
        break;
    case 2:
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
        glutWireSphere(1.0,slices, stacks);
        break;
    case 3:
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
        glutSolidSphere(1.0,slices, stacks);
        break;
    case 4:
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
        glutSolidSphere(1.0,slices, stacks);
        break;
}

}

void Android :: draw_cube() {
switch(this->visualization) {
    case 1:
        glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
        glutSolidCube(1.0);
        break;
    case 2:
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
        glutSolidCube(1.0);
        break;
    case 3:
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
        glutSolidCube(1.0);
        break;
    case 4:
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
        glutSolidCube(1.0);
        break;
}
}

void Android :: draw_cylinder(){
static GLUquadricObj * qobj = NULL;
if(qobj == NULL) {
    qobj = gluNewQuadric();
}
switch(this->visualization){
    case 1:
        gluQuadricDrawStyle(qobj, GLU_POINT);
        break;
    case 2:
        gluQuadricDrawStyle(qobj, GLU_LINE);
        break;
    case 3:
        gluQuadricDrawStyle(qobj, GLU_FILL);
        break;
    case 4:
        glColor3f(0.6,0.6,0.6); //Gray
        gluQuadricDrawStyle(qobj, GLU_FILL);
        break;
}
}

```

```
gluCylinder(qobj, 0.3, 0.3, 0.5, 25, 25);
glColor3f(0.0, 0.0, 1.0); //Blue
}

void Android :: draw_aux_object()
{
    glPushMatrix();
    glScalef(0.2, 0.2, 0.2);
    draw_sphere(20, 20);
    glTranslatef(0.0, -7.0, 0.0);
    draw_sphere(20, 20);
    glPopMatrix();
    glPushMatrix();
    glRotatef(90.0, 1.0, 0.0, 0.0);
    glScalef(0.6, 0.6, 3.0);
    draw_cylinder();
    glPopMatrix();
}

void Android :: draw_arm()
{
    draw_aux_object();

}

void Android :: draw_leg()
{
    glColor3f(0.0, 1.0, 0.0);
    draw_aux_object();
}

void Android :: draw_eyes()
{
    glPushMatrix();
    glScalef(0.05, 0.05, 0.05);
    glColor3f(0.0, 0.0, 0.0); //Black
    draw_sphere(20, 20);
    glColor3f(0.0, 1.0, 0.0);
    glPopMatrix();
}

void Android :: draw_antenna()
{
    draw_aux_object();
}

void Android :: draw_head()
{
    glPushMatrix();
    glTranslatef(0.0, -0.75, 0.0);
    glScalef(1.9, 1.9, 1.9);
    android_head->draw(visualization);
    glPushMatrix();
    glTranslatef(0.0, 1.0, 0.5);
    glPushMatrix();
    glTranslatef(0.3, 0.0, 0.0);
    draw_eyes();
    glPopMatrix();
    glPushMatrix();
    glTranslatef(-0.3, 0.0, 0.0);
    draw_eyes();
    glPopMatrix();
    glPopMatrix();
    glPopMatrix();
    glPushMatrix();
    glPushMatrix();
```

```

glTranslatef(0.8,2.2,0.0);
glScalef(0.1,0.5,0.1);
glRotatef(-50.0,0.0,0.0,1.0);
draw_antenna();
glPopMatrix();
glPushMatrix();
  glTranslatef(-0.8,2.2,0.0);
  glScalef(0.1,0.5,0.1);
  glRotatef(50.0,0.0,0.0,1.0);
  glColor3f(0.0,1.0,0.0);
  draw_antenna();
glPopMatrix();
glPopMatrix();
}

void Android :: draw_body()
{
  glPushMatrix();
  glScalef(2.0,2.5,2.0);
  glTranslatef(0.0,-0.5,0.0);
  android_body->draw(visualization);
  glPopMatrix();
  glPushMatrix();
    glScalef(0.8,0.8,0.8);
    glPushMatrix();
      glTranslatef(-1.5,0.5,0.0);
      draw_arm();
    glPopMatrix();
    glPushMatrix();
      glTranslatef(1.5,0.5,0.0);
      glColor3f(0.0,1.0,0.0);
      draw_arm();
    glPopMatrix();
  glPopMatrix();
  glPushMatrix();
    glTranslatef(0.0,-1.0,0.0);
    glScalef(1.1,0.5,1.1);
    glPushMatrix();
      glTranslatef(-0.5,0.0,0.0);
      draw_leg();
    glPopMatrix();
    glPushMatrix();
      glTranslatef(0.5,0.0,0.0);
      draw_leg();
    glPopMatrix();
  glPopMatrix();
}

void Android :: draw_robot()
{
  draw_head();
  draw_body();
}

void Android :: draw(visual_t visualization) {

  glPointSize(2.0);
  this->visualization = visualization;
  switch(visualization) {
    case POINT:
      glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
      break;
    case FILL:
      glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
      break;
  }
}

```

```
case LINE:  
case CHECKERED:  
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);  
    break;  
default:  
    glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);  
    break;  
  
}  
glColor3f(0.0,1.0,0.0);  
draw_robot();  
}
```

En la siguientes imágenes podemos visualizar el robot.

