

# CURSO: CC322 - 2017

## Practica Calificada.

### 1. Introducción

#### Objetivo general:

-Aplicar Pipeline.

### 2 . Recursos Informáticos

<https://www.opengl.org>

[https://www.khronos.org/opengl/wiki/Getting\\_Started](https://www.khronos.org/opengl/wiki/Getting_Started)

[https://www.khronos.org/opengl/wiki/Category:Core\\_API\\_Reference](https://www.khronos.org/opengl/wiki/Category:Core_API_Reference)

### 3. DESARROLLO

1. Verifique si tiene instalado un compilador de código en C (cpp, g++)

Sino, proceda a instalarlo en su PC

2. Verifique si tiene un editor de texto adecuado para editar programas en C

(p.ej. Geany, SublimeText, u otro con el que esté familiarizado) si no está - Proceda a Instalarlo.

3.Verifique si tiene instaladas las librerías OPENGLE Y GLUT. Si no están proceda a instalarlas.

sudo apt-get install freeglut3-dev

4. Al final entregara un archivo con el nombre CC322\_LabPC2\_<Nombre\_apellido>.zip con los archivos generados en la practica.

### 4 . Problemas

Copie el programa siguiente o bajelo del sitio web del curso. (Lab-pc-2.c).

```
#include <GL/glut.h>
float posicion;
int grados;
void reshape(int width, int height){
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-10, 10, -10, 10, 0.1f, 20);
    glTranslatef(-3.0f, 0.0f, -15.0f);
    glMatrixMode(GL_MODELVIEW);
}
void dibujar_rueda_y_tornillos_delante(){
    glPushMatrix();
    glColor3f(0.5f,0.5f,0.5f);
    glTranslatef(1.5f,0.0f, 0.0f);
    glutSolidSphere(1.0f,20,20);
    glPopMatrix();
}
void dibujar_rueda_y_tornillos_detras(){
    glPushMatrix();
    glColor3f(0.5f,0.5f,0.5f);
    glTranslatef(5.5f,0.0f, 3.0f);
    glutSolidSphere(1.0f,20,20);
}
```

```

    glPopMatrix();
}
void dibujar_ventanas(){
    glPushMatrix();
    glColor3f(0.0f,0.0f,1.0f);
    glTranslatef(1.2f,2.0f, 0.01f);
    glBegin(GL_TRIANGLES);
        glVertex3f(0.0f, 0.0f, 0.0f);
        glVertex3f(0.8f, 0.0f, 0.0f);
        glVertex3f(0.8f, 0.8f, 0.0f);
    glEnd();
    glBegin(GL_QUADS);
        glVertex3f(0.8f, 0.0f, 0.0f);
        glVertex3f(1.7f, 0.0f, 0.0f);
        glVertex3f(1.7f, 0.8f, 0.0f);
        glVertex3f(0.8f, 0.8f, 0.0f);
    glEnd();
    glTranslatef(1.9f,0.0f, 0.0f);
    glBegin(GL_QUADS);
        glVertex3f(0.0f, 0.0f, 0.0f);
        glVertex3f(0.9f, 0.0f, 0.0f);
        glVertex3f(0.9f, 0.8f, 0.0f);
        glVertex3f(0.0f, 0.8f, 0.0f);
    glEnd();
    glBegin(GL_TRIANGLES);
        glVertex3f(0.9f, 0.0f, 0.0f);
        glVertex3f(1.7f, 0.0f, 0.0f);
        glVertex3f(0.9f, 0.8f, 0.0f);
    glEnd();
    glPopMatrix();
}
void techo(){
    glPushMatrix();
    glTranslatef(1.0f,2.0f, 0.0f);
    glBegin(GL_TRIANGLES);
        glVertex3f(0.0f, 0.0f, 0.0f);
        glVertex3f(1.0f, 0.0f, 0.0f);
        glVertex3f(1.0f, 1.0f, 0.0f);
    glEnd();
    glBegin(GL_QUADS);
        glVertex3f(1.0f, 0.0f, 0.0f);
        glVertex3f(3.0f, 0.0f, 0.0f);
        glVertex3f(3.0f, 1.0f, 0.0f);
        glVertex3f(1.0f, 1.0f, 0.0f);
    glEnd();
    glBegin(GL_TRIANGLES);
        glVertex3f(3.0f, 0.0f, 0.0f);
        glVertex3f(4.0f, 0.0f, 0.0f);
        glVertex3f(3.0f, 1.0f, 0.0f);
    glEnd();
    glPopMatrix();
}
void dibujar_cuerpo_coche(){
    glColor3f(1,0,0);
    glTranslatef(0.0f,0.0f, 0.0f);
    glBegin(GL_QUADS);
        glVertex3f(0.0f, 0.0f, 0.0f);
        glVertex3f(7.0f, 0.0f, 0.0f);
        glVertex3f(7.0f, 2.0f, 0.0f);
        glVertex3f(0.0f, 2.0f, 0.0f);
    glEnd();
}
void dibujar_automovil(){
    dibujar_cuerpo_coche();
    techo();
    dibujar_ventanas();
    dibujar_rueda_y_tornillos_delante();
    dibujar_rueda_y_tornillos_detras();
}
void display(){
    GLfloat angulo;
    int i;
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    dibujar_automovil();
    glPopMatrix();
    glutSwapBuffers();
}
void init(){
    glClearColor(0,0,0,0);
    posicion=1.5;
    grados=0;
}

```

```

    glEnable(GL_DEPTH_TEST);
}

void keyboard(unsigned char key, int x, int y){
    switch(key)
    {
        case 'p':
        case 'P':
            display();
            break;

        case 'o':
        case 'O':
            display();
            break;
        case 27: // escape
            exit(0);
            break;
    }
}

int main(int argc, char **argv){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowPosition(50, 50);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Practica Calificada");
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```

Modifique el programa de tal manera que:

- a)(2 puntos) El automóvil cuente con ruedas que tengan 6 “tornillos”
- b)(2 puntos) El automóvil sea de modelo mas “aerodinámico” - Nuevo diseño de parachoques y parte trasera
- c)(2 puntos) El automovil se mueve a la izquierda y a la derecha presionando 2 teclas distintas
- d)(2 puntos) Las ruedas del automóvil giran en la dirección correcta, a medida que el automóvil avanza o retrocede.
- e)(4 puntos) Dibuje un segundo automóvil de color diferente y que se mueva de manera similar al primero, pero mas rapido. Este debe ser controlado por otras 2 teclas.
- f)(4 puntos) Dibuje un obstaculo que aparece y desaparece en forma aleatoria en el camino, y no permite que los automoviles se desplacen cuando chocan con el.
- g)(4 puntos) Modifique el programa anterior para que uno de los automoviles pase **sobre** el obstaculo.

Comente las modificaciones que haga en su código, Comprima los archivos resultantes y suba el archivo comprimido al repositorio del curso.