



3. Problema asignado para entrega. Escriba un programa en OpenGL ES, que genere un interesante escenario en dos dimensiones utilizando cuadrados, triangulos y circulos en multiples colores. Denomine a este proyecto OpenGLES-Escena. Puede dibujar un circulo con una secuencia de segmentos de linea.

La solución debe incluir en un documento pdf, con la siguiente información:

a. Código fuente.

```
public class Renderiza implements Renderer {
    /* Objeto */
    private List<Triangulo> triangulo_array = new ArrayList();
    private List<Triangulo> triangulo_array_circulo_blanco = new ArrayList();
    private List<Triangulo> triangulo_array_circulo_cafe1 = new ArrayList();
    private List<Triangulo> triangulo_array_circulo_cafe2 = new ArrayList();
    private List<Triangulo> triangulo_array_dientes = new ArrayList(); // para los dientes
    private List<Triangulo> triangulo_array_lengua = new ArrayList(); // para la lengua
    private List<Triangulo> triangulo_array_boca = new ArrayList(); // para la boca
    private List<Triangulo> triangulo_array_ojos = new ArrayList(); // para los ojos
    private List<Triangulo> triangulo_array_iris = new ArrayList(); // para la iris
    private List<Triangulo> triangulo_array_mano_d = new ArrayList(); // para la mano d
    private List<Triangulo> triangulo_array_mano_i = new ArrayList(); // para la mano i
    private List<Triangulo> triangulo_array_pies = new ArrayList(); // para los pies
    private List<Triangulo> triangulo_array_cuerpo = new ArrayList(); // para el cuerpo
    private List<Triangulo> triangulo_array_quino = new ArrayList(); // para el cuerpo
    @Override
    public void onSurfaceCreated(GL10 gl, EGLConfig arg1) {
        // COLORES:
        Colores colores = new Colores();

        // BOMBA POKEMON
        int nro_triangulos=360;
        CirculoVertices circuloVertices = new CirculoVertices(-140,-10,85);
        for(int i = 0; i < nro_triangulos; i++){
            triangulo_array_circulo_blanco.add(new Triangulo(
                circuloVertices.getVerticesCirculo(i),
                colores.getRgba(2)));
        }
        circuloVertices = new CirculoVertices(-140,-10,80);
        for(int i = 0; i < nro_triangulos; i++){
            triangulo_array_circulo_cafe1.add(new Triangulo(
                circuloVertices.getVerticesCirculo(i),
                colores.getRgba(3)));
        }
        circuloVertices = new CirculoVertices(-140,-10,40);
        for(int i = 0; i < nro_triangulos; i++){
            triangulo_array_circulo_cafe2.add(new Triangulo(
                circuloVertices.getVerticesCirculo(i),
                colores.getRgba(4)));
        }

        // RAYOS BOMBA POKEMON
        nro_triangulos = 38+12+15;
        TrianguloVertices trianguloVertices = new TrianguloVertices();
        for(int i = 0; i <= nro_triangulos; i++){
            triangulo_array.add(new Triangulo(
                trianguloVertices.getVerticesTriangulo(i),
                colores.getRgba(1)));
        }
    }
}
```

```

// DIENTES
nro_triangulos = 20;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_dientes.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_dientes(i),
        colores.getRgba(5)));
}

// LENGUA
nro_triangulos = 9;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_lengua.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_lengua(i),
        colores.getRgba(6)));
}

// BOCA
nro_triangulos = 16;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_boca.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_boca(i),
        colores.getRgba(7)));
}

// OJOS
nro_triangulos = 14;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_ojos.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_ojos(i),
        colores.getRgba(8)));
}

// IRIS
nro_triangulos = 2;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_iris.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_iris(i),
        colores.getRgba(9)));
}

// CUERPO MANO D
nro_triangulos = 17;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_mano_d.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_mano_d(i),
        colores.getRgba(10)));
}

// CUERPO MANO I
nro_triangulos = 14;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_mano_i.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_mano_i(i),
        colores.getRgba(10)));
}

// CUERPO PIES
nro_triangulos = 19+16;
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_pies.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_pies(i),
        colores.getRgba(10)));
}

// CUERPO
nro_triangulos = 23+16+17+12+25;
trianguloVertices = new TrianguloVertices();

```

```

for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_cuerpo.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_cuerpo(i),
        colores.getRgba(10)));
}

// QUINO
circuloVertices = new CirculoVertices(160,210,50);
for(int i = 0; i < 360; i++){
    triangulo_array_circulo_cafe2.add(new Triangulo(
        circuloVertices.getVerticesCirculo(i),
        colores.getRgba(8)));
}
nro_triangulos = 17;//17
trianguloVertices = new TrianguloVertices();
for(int i = 0; i < nro_triangulos; i++){
    triangulo_array_quino.add(new Triangulo(
        trianguloVertices.getVerticesTriangulo_quino(i),
        colores.getRgba(10)));
}

/* Color de fondo */
gl.glClearColor(0, 1, 1, 0);
//gl.glClearColor(224f/255f, 247f/255f, 250f/255f, 0); // claro
//gl.glClearColor(50f/255f, 33f/255f, 35f/255f, 0); // oscuro
}
@Override
public void onDrawFrame(GL10 gl) {
    /* Inicializa el buffer de color */
    gl.glClear(GL10.GL_COLOR_BUFFER_BIT);

    for (int i = 0; i < triangulo_array_circulo_blanco.size(); i++){
        triangulo_array_circulo_blanco.get(i).dibuja(gl);
    }
    for (int i = 0; i < triangulo_array_circulo_cafe1.size(); i++){
        triangulo_array_circulo_cafe1.get(i).dibuja(gl);
    }
    for (int i = 0; i < triangulo_array_circulo_cafe2.size(); i++){
        triangulo_array_circulo_cafe2.get(i).dibuja(gl);
    }

    for (int i = 0; i < triangulo_array.size(); i++){
        triangulo_array.get(i).dibuja(gl);
    }

    // DIENTES
    for (int i = 0; i < triangulo_array_dientes.size(); i++)
        triangulo_array_dientes.get(i).dibuja(gl);
    // LENGUA
    for (int i = 0; i < triangulo_array_lengua.size(); i++)
        triangulo_array_lengua.get(i).dibuja(gl);
    // BOCA
    for (int i = 0; i < triangulo_array_boca.size(); i++)
        triangulo_array_boca.get(i).dibuja(gl);
    // OJOS
    for (int i = 0; i < triangulo_array_ojos.size(); i++)
        triangulo_array_ojos.get(i).dibuja(gl);
    // IRIS
    for (int i = 0; i < triangulo_array_iris.size(); i++)
        triangulo_array_iris.get(i).dibuja(gl);
    // CUERPO MANO DERECHA
    for (int i = 0; i < triangulo_array_mano_d.size(); i++)
        triangulo_array_mano_d.get(i).dibuja(gl);
    // CUERPO MANO IZQUIERDA
    for (int i = 0; i < triangulo_array_mano_i.size(); i++)
        triangulo_array_mano_i.get(i).dibuja(gl);
    // CUERPO PIES
    for (int i = 0; i < triangulo_array_pies.size(); i++)
        triangulo_array_pies.get(i).dibuja(gl);
    // CUERPO
    for (int i = 0; i < triangulo_array_cuerpo.size(); i++)
        triangulo_array_cuerpo.get(i).dibuja(gl);
    // QUINO
    for (int i = 0; i < triangulo_array_quino.size(); i++)
        triangulo_array_quino.get(i).dibuja(gl);
}

```

```

    }
}
public class Colores {
    private float rgba[];

    /* Los colores x vértice (r,g,b,a) */
    private float lilas_tonos[] = new float [] {
        228f/255f, 210f/255f, 231f/255f, 1, // 0    lila
        215f/255f, 181f/255f, 216f/255f, 1, // 1    lila
        201f/255f, 155f/255f, 203f/255f, 1, // 2    lila
        189f/255f, 140f/255f, 195f/255f, 1, // 3    lila
        177f/255f, 114f/255f, 182f/255f, 1, // 4    lila
        164f/255f, 91f/255f, 170f/255f, 1, // 5    lila
    };
    private float azul_tonos[] = new float [] {
        178f/255f, 235f/255f, 242f/255f, 1, // 0    azul
        80f/255f, 222f/255f, 234f/255f, 1, // 1    azul
        26f/255f, 198f/255f, 218f/255f, 1, // 2    azul
        00f/255f, 172f/255f, 193f/255f, 1, // 3    azul
        00f/255f, 97f/255f, 167f/255f, 1, // 4    azul
        00f/255f, 83f/255f, 143f/255f, 1, // 5    azul
    };
    private float entero[] = new float [] {
        1, 0, 0, 1, // 0    rojo
        0, 1, 0, 1, // 1    verde
        0, 0, 1, 1, // 2    azul
        0, 0, 0, 1, // 3    negro
    };
    private float rosado_tonos[] = new float [] {
        253f/255f, 164f/255f, 186f/255f, 1, // 0    blanco
        252f/255f, 186f/255f, 203f/255f, 1, // 0    blanco
        254f/255f, 197f/255f, 229f/255f, 1, // 0    blanco
        247f/255f, 154f/255f, 192f/255f, 1, // 0    blanco
        253f/255f, 171f/255f, 159f/255f, 1, // 0    blanco
        242f/255f, 107f/255f, 138f/255f, 1, // 0    blanco
    };
    private float gris_tonos[] = new float [] {
        120f/255f, 120f/255f, 120f/255f, 1, // 0    blanco
        130f/255f, 130f/255f, 130f/255f, 1, // 0    blanco
        140f/255f, 140f/255f, 140f/255f, 1, // 0    blanco
        150f/255f, 150f/255f, 150f/255f, 1, // 0    blanco
        160f/255f, 160f/255f, 160f/255f, 1, // 0    blanco
        170f/255f, 170f/255f, 170f/255f, 1, // 0    blanco
    };
    private float blanco_tonos[] = new float [] {
        255f/255f, 255f/255f, 255f/255f, 1, // 0    blanco
        250f/255f, 250f/255f, 250f/255f, 1, // 0    blanco
        245f/255f, 245f/255f, 245f/255f, 1, // 0    blanco
        240f/255f, 240f/255f, 240f/255f, 1, // 0    blanco
        235f/255f, 235f/255f, 235f/255f, 1, // 0    blanco
        230f/255f, 230f/255f, 230f/255f, 1, // 0    blanco
    };
    private float cafe_tonos[] = new float [] {
        62f/255f, 27f/255f, 23f/255f, 1, // 0    lila
        78f/255f, 34f/255f, 46f/255f, 1, // 0    lila
        201f/255f, 155f/255f, 203f/255f, 1, // 2    lila
        189f/255f, 140f/255f, 195f/255f, 1, // 3    lila
        177f/255f, 114f/255f, 182f/255f, 1, // 4    lila
        164f/255f, 91f/255f, 170f/255f, 1, // 5    lila
    };
    float[] getRgba(int option){
        rgba = new float[4];
        switch (option){
            case 1:
                // colores RGBA tonos de lila
                int random = getRandomInt();
                rgba = new float[4];
                rgba[0] = lilas_tonos[(random*4) + 0];
                rgba[1] = lilas_tonos[(random*4) + 1];
                rgba[2] = lilas_tonos[(random*4) + 2];
                rgba[3] = lilas_tonos[(random*4) + 3];
                break;

```

```

case 2:
    rgba[0] = blanco_tonos[0];
    rgba[1] = blanco_tonos[1];
    rgba[2] = blanco_tonos[2];
    rgba[3] = blanco_tonos[3];
    break;
case 3:
    rgba[0] = cafe_tonos[0];
    rgba[1] = cafe_tonos[1];
    rgba[2] = cafe_tonos[2];
    rgba[3] = cafe_tonos[3];
    break;
case 4:
    rgba[0] = cafe_tonos[4];
    rgba[1] = cafe_tonos[5];
    rgba[2] = cafe_tonos[6];
    rgba[3] = cafe_tonos[7];
    break;
case 5:
    // colores RGBA para dientes
    random = getRandomInt();
    rgba = new float[4];
    rgba[0] = blanco_tonos[(random*4) + 0];
    rgba[1] = blanco_tonos[(random*4) + 1];
    rgba[2] = blanco_tonos[(random*4) + 2];
    rgba[3] = blanco_tonos[(random*4) + 3];
    break;
case 6:
    // colores RGBA para Lengua
    random = getRandomInt();
    rgba = new float[4];
    rgba[0] = rosado_tonos[(random*4) + 0];
    rgba[1] = rosado_tonos[(random*4) + 1];
    rgba[2] = rosado_tonos[(random*4) + 2];
    rgba[3] = rosado_tonos[(random*4) + 3];
    break;
case 7:
    // colores RGBA para boca
    random = getRandomInt();
    rgba = new float[4];
    rgba[0] = gris_tonos[(random*4) + 0];
    rgba[1] = gris_tonos[(random*4) + 1];
    rgba[2] = gris_tonos[(random*4) + 2];
    rgba[3] = gris_tonos[(random*4) + 3];
    break;
case 8:
    // colores RGBA para ojos ROJO
    rgba = new float[4];
    rgba[0] = entero[0];
    rgba[1] = entero[1];
    rgba[2] = entero[2];
    rgba[3] = entero[3];
    break;
case 9:
    // colores RGBA para ojos ROJO
    rgba = new float[4];
    rgba[0] = entero[12];
    rgba[1] = entero[13];
    rgba[2] = entero[14];
    rgba[3] = entero[15];
    break;
case 10:
    // colores RGBA tonos de Lila
    random = getRandomInt();
    rgba = new float[4];
    rgba[0] = azul_tonos[(random*4) + 0];
    rgba[1] = azul_tonos[(random*4) + 1];
    rgba[2] = azul_tonos[(random*4) + 2];
    rgba[3] = azul_tonos[(random*4) + 3];
    break;
default:
    break;
}
return rgba;

```

```

}
public int getRandomInt(){
    int numero = (int)(Math.random()*10);
    if (numero > 5)
        numero = numero - 4;
    return numero;
}
}

```

b. Salida de su código.

