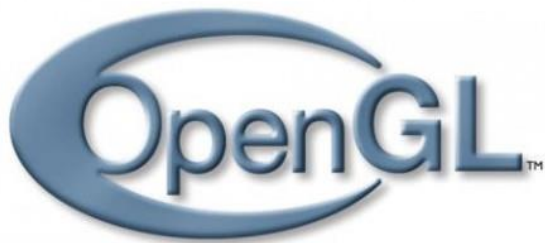


Primitivas.



OpenGL ES
Parte II

Jhonny Felípez Andrade



jrfelizamigo@yahoo.es



Contenido

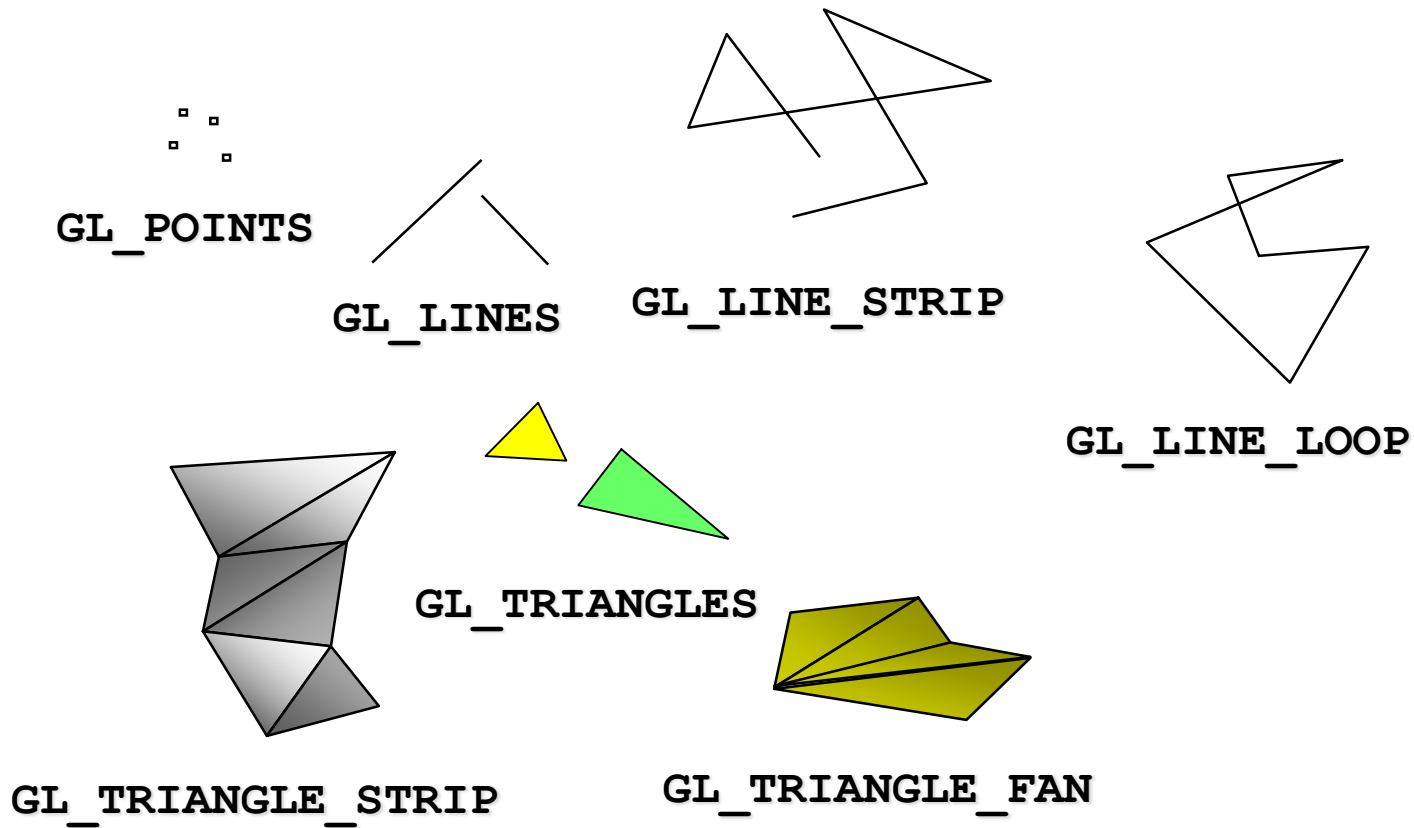
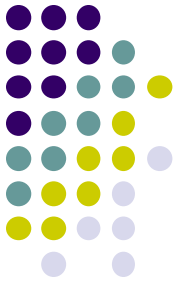
- Primitivas del OpenGL ES
- Volumen de visualización.
- Código en OpenGL ES 1.0



Primitivas

PRIMITIVAS DEL OPENGL ES

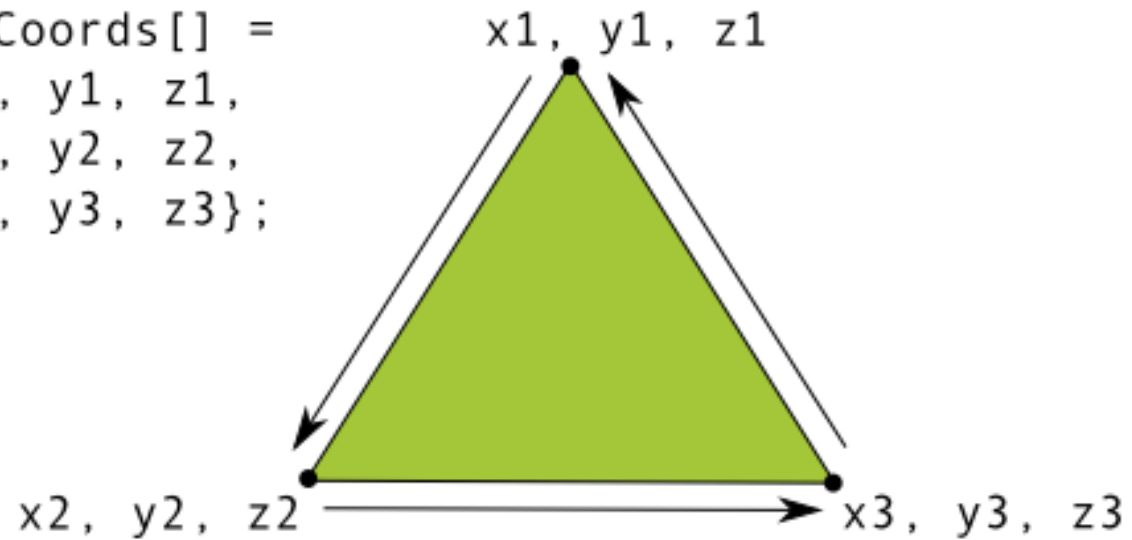
Primitivas del OpenGL



Triangulo

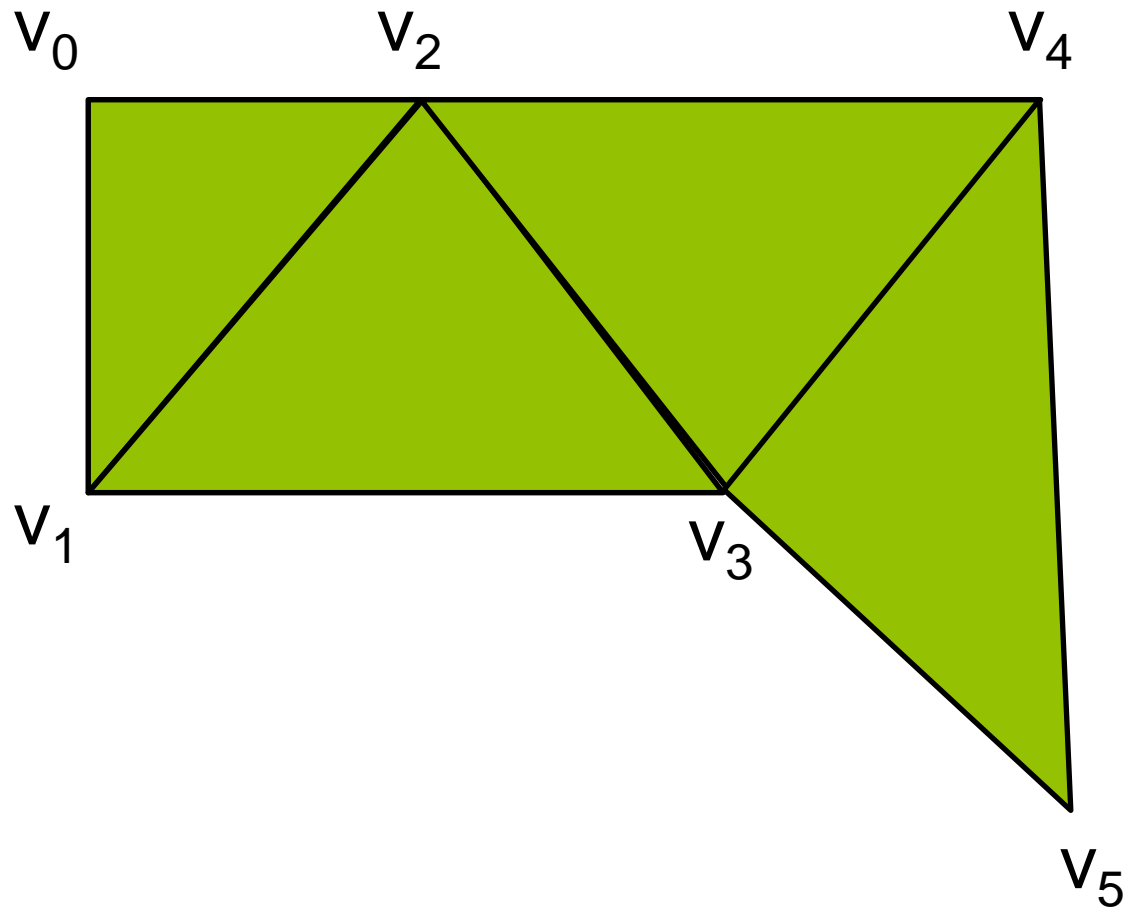
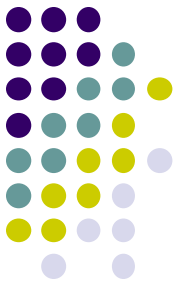


```
float triangleCoords[] =  
    {x1, y1, z1,  
     x2, y2, z2,  
     x3, y3, z3};
```



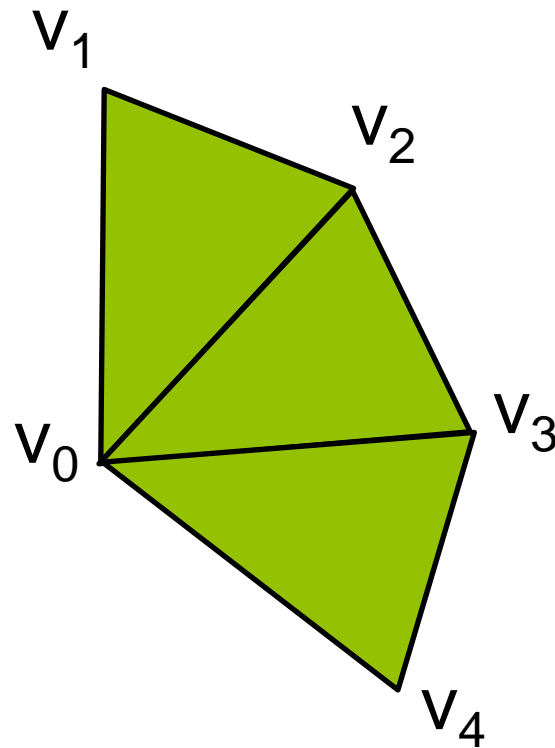
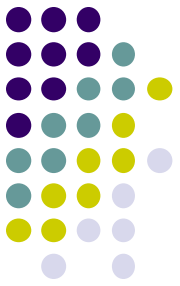
Triangulos Conectados

GL_TRIANGLE_STRIP

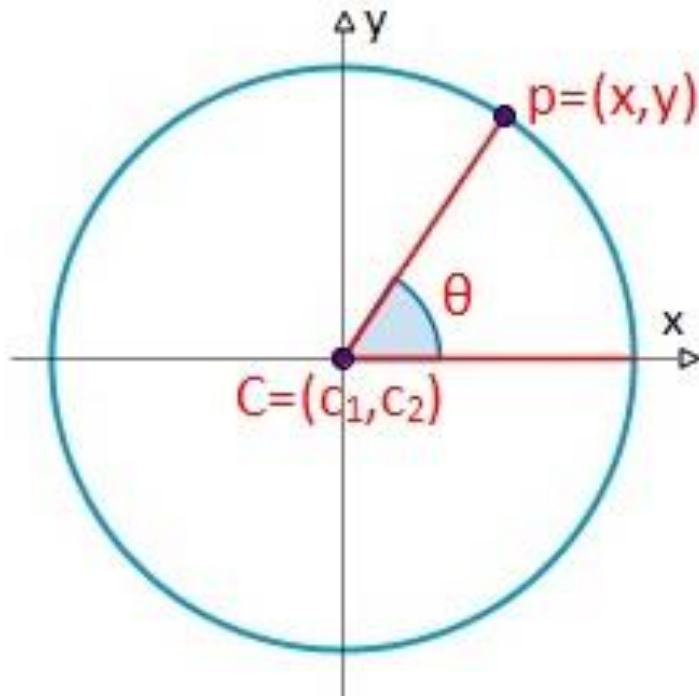


Triangulos Conectados Abanico

GL_TRIANGLE_FAN



Ecuación Paramétrica de una Circunferencia



$$x = c_1 + r \cos \theta$$

$$y = c_2 + r \sin \theta$$

Siendo $C = (c_1, c_2)$ el centro y θ el ángulo del punto.

El ángulo se puede expresar en:

- radianes ($0 \leq \theta \leq 2\pi$).
- grados sexagesimales ($0^\circ \leq \theta \leq 360^\circ$).



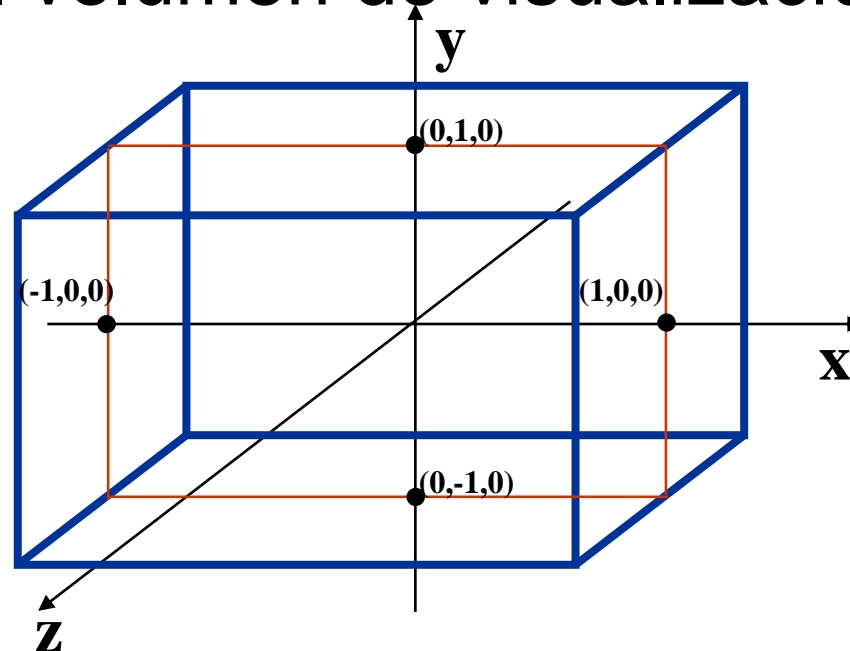
Primitivas

VOLUMEN DE VISUALIZACIÓN



Volumen de Visualización

- OpenGL no trabaja con coordenadas de pantalla, sino con coordenadas posicionales dentro del volumen de visualización



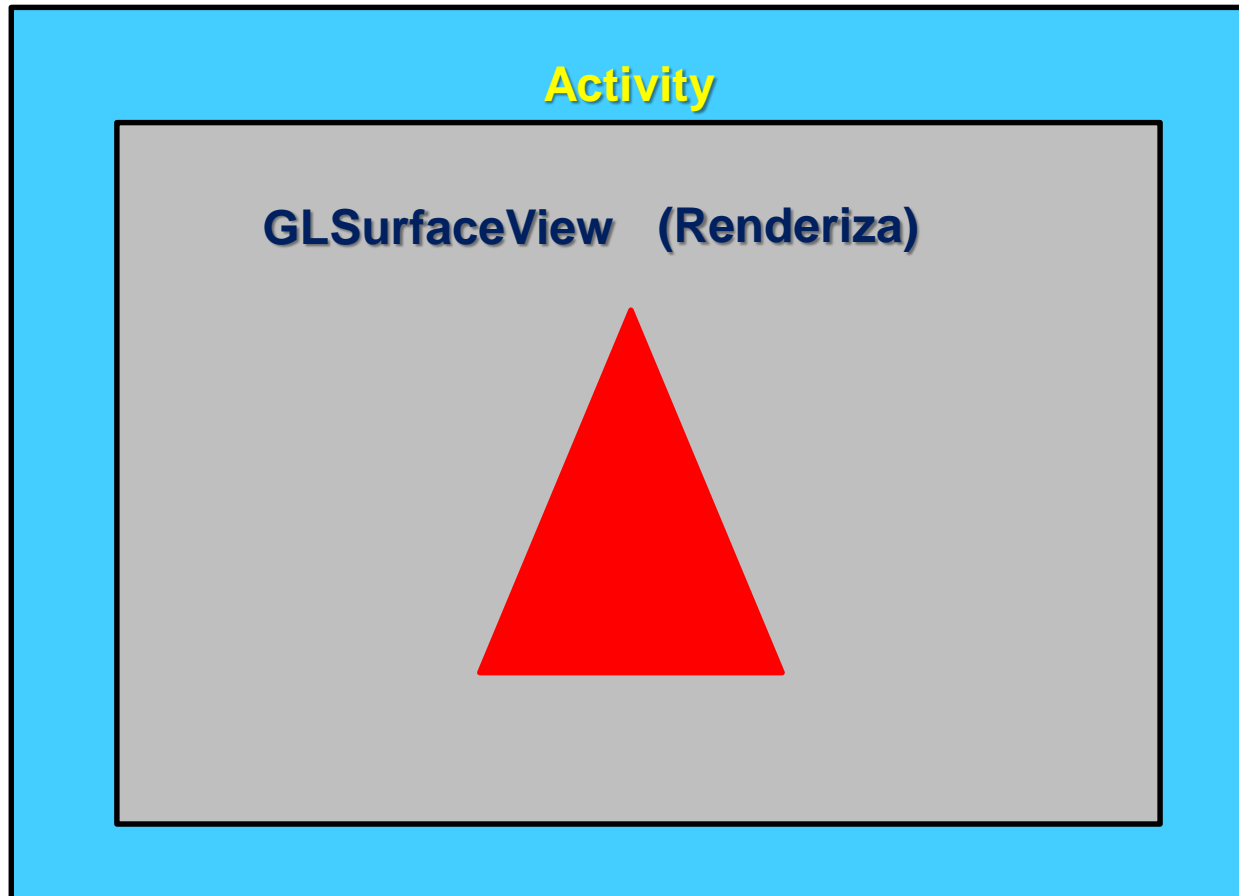
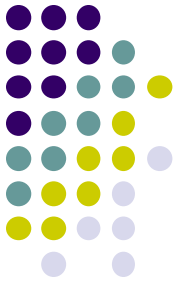
```
gluOrtho2D (gl, -1, 1, -1, 1)  
glOrthof (-1, 1, -1, 1, -1, 1);
```



Primitivas

CÓDIGO EN OPENGLES 1.0


Android





```
MainActivity {  
  
    onCreate() {  
  
        GLSurfaceView <- Renderiza  
  
        inicia_el_renderizado;  
  
        Activity <- GLSurfaceView  
    }  
}
```

```
Renderiza {  
  
    onSurfaceCreated()  
  
    onDrawFrame()  
  
    onSurfaceChanged()  
}
```



```
import android.opengl.GLSurfaceView;
import android.os.Bundle;
import android.app.Activity;
public class MainActivity extends Activity {
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

GLSurfaceView <- Renderiza



```
    GLSurfaceView superficie = new GLSurfaceView(this);
    Renderiza renderiza = new Renderiza();
    superficie.setRenderer(renderiza);
```

inicia_el_renderizado

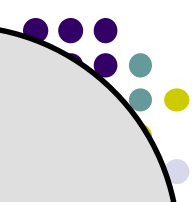


```
    setContentView(superficie);
```

Activity <- GLSurfaceView

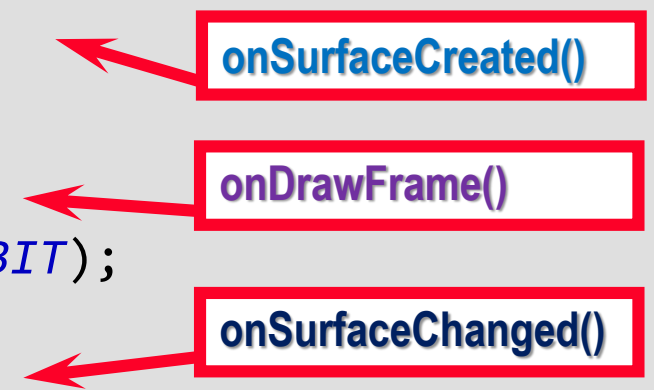


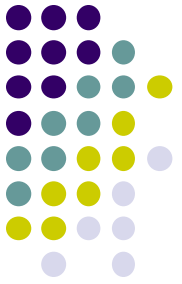
```
    }
}
```



```
import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;
import android.opengl.GLSurfaceView.Renderer;
```

```
public class Renderiza implements Renderer {
    @Override
    public void onSurfaceCreated(GL10 gl, EGLConfig arg1) {
        gl.glClearColor(0, 1, 1, 0);
    }
    @Override
    public void onDrawFrame(GL10 gl) {
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT);
    }
    @Override
    public void onSurfaceChanged(GL10 gl, int w, int h) {
        gl.glViewport(0, 0, w, h);
    }
}
```





FIN