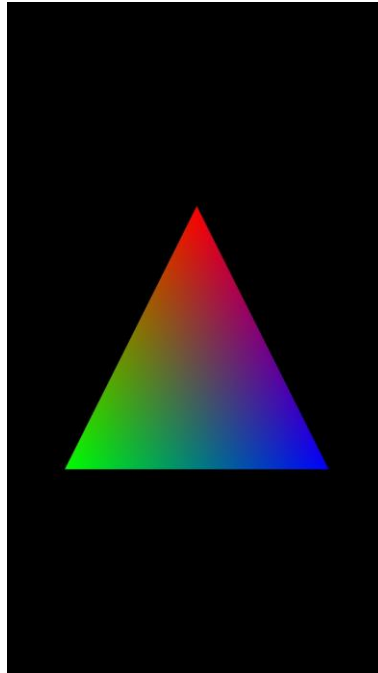


Android - OpenGL ES 2 - Tutorial 4

Vertici Colorati

In questo tutorial imposteremo singolarmente i colori dei vertici:



Modifichiamo il progetto precedente nel seguente modo:

- nella classe "ObjTriangle", aggiungiamo la variabile per il colore:

```
private FloatBuffer colorBuffer;
```

aggiungiamo le informazioni dei colori:

```
private final int COMPONENT_PER_COLOR = 4;  
private final int colorStride = COMPONENT_PER_COLOR * 4;  
private float[] colors = {  
    0.0f, 1.0f, 0.0f, 1.0f,  
    0.0f, 0.0f, 1.0f, 1.0f,  
    1.0f, 0.0f, 0.0f, 1.0f  
};
```

aggiungiamo la conversione dei colori in formato utile ad OpenGL:

```
ByteBuffer vcc = ByteBuffer.allocateDirect(this.colors.length * 4);  
vcc.order(ByteOrder.nativeOrder());  
this.colorBuffer = vcc.asFloatBuffer();  
this.colorBuffer.put(this.colors);  
this.colorBuffer.position(0);
```

modifichiamo lo shader:



```
private final String vertexShaderCode =
    "uniform mat4 uMVPMatrix;" +
    "attribute vec4 vPosition;" +
    "attribute vec4 iColor;" +
    "varying vec4 vColor;" +
    "void main() {" +
    "    vColor = iColor;" +
    "    gl_Position = uMVPMatrix * vPosition;" +
    "}";

private final String fragmentShaderCode =
    "precision mediump float;" +
    "varying vec4 vColor;" +
    "void main() {" +
    "    gl_FragColor = vColor;" +
    "}";
```

rinominiamo il metodo in "DrawTriangleRainbow()" e lo modifichiamo con:

```
public void DrawTriangleRainbow(float[] modelMatrix) {
    GLES20.glUseProgram(this.mProgram);

    this.positionHandle = GLES20.glGetAttribLocation(this.mProgram, "vPosition");
    GLES20.glEnableVertexAttribArray(this.positionHandle);
    GLES20.glVertexAttribPointer(this.positionHandle, COORDS_PER_VERTEX, GLES20.GL_FLOAT,
    false, this.vertexStride, this.vertexBuffer);

    this.colorHandle = GLES20.glGetAttribLocation(this.mProgram, "iColor");
    GLES20.glEnableVertexAttribArray(this.colorHandle);
    GLES20.glVertexAttribPointer(this.colorHandle, COMPONENT_PER_COLOR, GLES20.GL_FLOAT,
    false, this.colorStride, this.colorBuffer);

    this.modelHandle = GLES20.glGetUniformLocation(this.mProgram, "uMVPMatrix");
    GLES20.glUniformMatrix4fv(this.modelHandle, 1, false, modelMatrix, 0);

    GLES20.glDrawElements(GL10.GL_TRIANGLES, this.indices.length, GL10.GL_UNSIGNED_SHORT,
    this.indexBuffer);

    GLES20.glDisableVertexAttribArray(this.positionHandle);
    GLES20.glDisableVertexAttribArray(this.colorHandle);
}
```

- nel metodo "onDrawFrame()" usiamo:

```
this.ResetModelMatrix();
Matrix.translateM(this.modelMatrix, 0, this.width / 2.0f, this.height / 2.0f, 0.0f);
Matrix.scaleM(this.modelMatrix, 0, 500.0f, 500.0f, 0.0f);
this.objtriangle.DrawTriangleRainbow(this.modelMatrix);
```