Android - OpenGL ES 1 - Tutorial 6

Apply Texture

Now let's see how to draw a square on which to apply a texture:

Create the class "ObjTexture" with the following code:

```
package opengles1.tutorial6;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.opengl.GLUtils;

import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.FloatBuffer;
import java.nio.ShortBuffer;

import javax.microedition.khronos.opengles.GL10;

public class ObjTexture {
    private FloatBuffer vertexBuffer;
    private ShortBuffer indexBuffer;
    private FloatBuffer textureBuffer;

    private float[] vertices = {
            -0.5f, -0.5f,
            0.5f, -0.5f,
            -0.5f, 0.5f,
            0.5f, 0.5f
    };
```

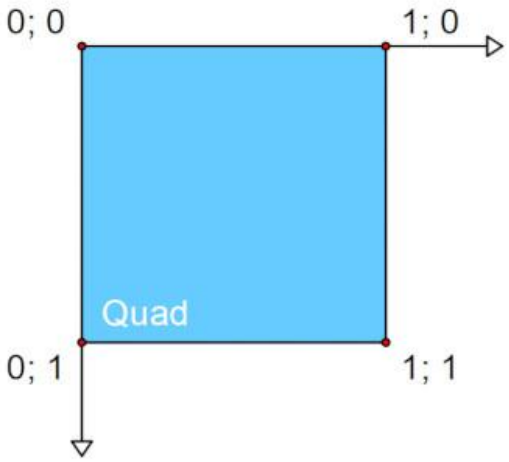
```
    private short[] indices = {
            0, 1, 2,
            2, 1, 3
    };

    private float[] texture = {
            0.0f, 1.0f,
            1.0f, 1.0f,
            0.0f, 0.0f,
            1.0f, 0.0f
    };

    public ObjTexture() {
        ByteBuffer vbb = ByteBuffer.allocateDirect(this.vertices.length * 4);
        vbb.order(ByteOrder.nativeOrder());
        this.vertexBuffer = vbb.asFloatBuffer();
        this.vertexBuffer.put(this.vertices);
        this.vertexBuffer.position(0);

        ByteBuffer ibb = ByteBuffer.allocateDirect(this.indices.length * 2);
        ibb.order(ByteOrder.nativeOrder());
        this.indexBuffer = ibb.asShortBuffer();
        this.indexBuffer.put(this.indices);
        this.indexBuffer.position(0);

        ByteBuffer tbb = ByteBuffer.allocateDirect(this.texture.length * 4);
        tbb.order(ByteOrder.nativeOrder());
        this.textureBuffer = tbb.asFloatBuffer();
        this.textureBuffer.put(this.texture);
        this.textureBuffer.position(0);
    }

    private final int text_count = 2; // Texture count.
    private int textures[] = new int[this.text_count]; // Texture pointer.

    public void LoadTexture(GL10 gl, Context context) {
        // Generate texture names.
        gl.glGenTextures(this.text_count, this.textures, 0);

        Bitmap bitmap;

        // Set bitmap from resources.
        bitmap = BitmapFactory.decodeResource(context.getResources(),
R.drawable.boxwood);
        // Bind a named texture to a texturing target.
        gl.glBindTexture(GL10.GL_TEXTURE_2D, this.textures[0]);
        // Set texture parameters.
        gl.glTexParameterf(GL10.GL_TEXTURE_2D, GL10.GL_TEXTURE_MIN_FILTER,
GL10.GL_LINEAR);
        // Set texture parameters.
        gl.glTexParameterf(GL10.GL_TEXTURE_2D, GL10.GL_TEXTURE_MAG_FILTER,
GL10.GL_LINEAR);
        // Load texture.
        GLUtils.texImage2D(GL10.GL_TEXTURE_2D, 0, bitmap, 0);

        bitmap = BitmapFactory.decodeResource(context.getResources(),
R.drawable.textwood);
        gl.glBindTexture(GL10.GL_TEXTURE_2D, this.textures[1]);
        gl.glTexParameterf(GL10.GL_TEXTURE_2D, GL10.GL_TEXTURE_MIN_FILTER,
```

```
GL10.GL_LINEAR);
        gl.glTexParameterf(GL10.GL_TEXTURE_2D, GL10.GL_TEXTURE_MAG_FILTER,
GL10.GL_LINEAR);
        GLUtils.texImage2D(GL10.GL_TEXTURE_2D, 0, bitmap, 0);

        // Clean up.
        bitmap.recycle();
    }

    public void DrawTexture(GL10 gl, int texture) {
        gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
        gl.glVertexPointer(2, GL10.GL_FLOAT, 0, this.vertexBuffer);
        gl.glEnableClientState(GL10.GL_TEXTURE_COORD_ARRAY);
        gl.glTexCoordPointer(2, GL10.GL_FLOAT, 0, this.textureBuffer);
        // Bind a named texture to a texturing target.
        gl.glBindTexture(GL10.GL_TEXTURE_2D, this.textures[texture]);
        gl.glDrawElements(GL10.GL_TRIANGLES, this.indices.length, GL10.GL_UNSIGNED_SHORT,
this.indexBuffer);
        gl.glDisableClientState(GL10.GL_VERTEX_ARRAY);
        gl.glDisableClientState(GL10.GL_TEXTURE_COORD_ARRAY);
    }

}
```

The square is that of tutorial 5, in addition we find the coordinates of the texture to be applied.

The textures (present in the project in "res\drawable") must have dimensions multiple of 2, in our case they are 512x512 pixels.

The texture coordinate system is:



OpenGL will draw the 2 triangles of the square by applying the texture using the coordinates.

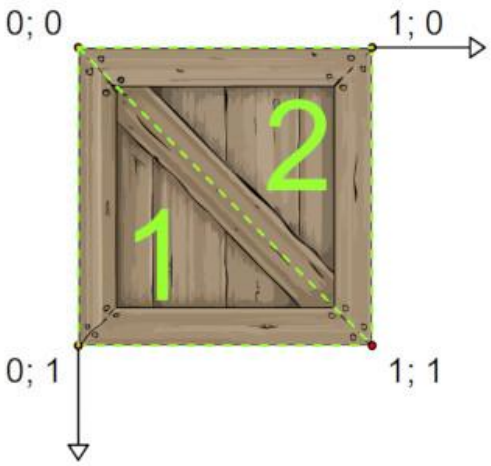
For the first triangle, indices 0, 1, 2:

- P1(-0.5;-0.5) P2(0.5;-0.5) P3(-0.5;0.5)
- TextureP1(0;1) TextureP2(1;1) TextureP3(0;0)

For the second triangle, indices 2, 1, 3:

- P1(-0.5;0.5) P2(0.5;-0.5) P3(0.5;0.5)

- TextureP1(0;0) TextureP2(1;1) TextureP3(1;0)



In the "MainRenderer" class we have:

```
package opengles1.tutorial6;

import android.content.Context;
import android.opengl.GLSurfaceView;

import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;

public class MainRenderer implements GLSurfaceView.Renderer {
    MainActivity activity;
    int width, height;

    ObjTexture objtexture;

    public MainRenderer(Context context) {
        // Set context.
        this.activity = (MainActivity) context;

        this.objtexture = new ObjTexture();
    }

    @Override
    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        gl.glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
        // Enable texture.
        gl.glEnable(GL10.GL_TEXTURE_2D);

        // Load texture.
        this.objtexture.LoadTexture(gl, this.activity);
    }

    @Override
    public void onSurfaceChanged(GL10 gl, int width, int height) {
        this.width = width;
        this.height = height;

        gl.glViewport(0, 0, this.width, this.height);
        gl.glMatrixMode(GL10.GL_PROJECTION);
```

```
        gl.glLoadIdentity();
        gl.glOrthof(0.0f, this.width - 1.0f, 0.0f, this.height - 1.0f, 1.0f, -1.0f);

        gl.glMatrixMode(GL10.GL_MODELVIEW);
    }

    @Override
    public void onDrawFrame(GL10 gl) {
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT);

        gl.glLoadIdentity();
        gl.glTranslatef(this.width / 2.0f, this.height / 4.0f * 3.0f, 0.0f);
        gl.glScalef(300.0f, 300.0f, 0.0f);
        this.objtexture.DrawTexture(gl, 0);

        gl.glLoadIdentity();
        gl.glTranslatef(this.width / 2.0f, this.height / 4.0f, 0.0f);
        gl.glScalef(300.0f, 300.0f, 0.0f);
        this.objtexture.DrawTexture(gl, 1);
    }

}
```

In the "onSurfaceCreated()" method is enabled the use of textures and the textures are loaded with the "LoadTexture()" method.

With "DrawTexture()" we draw the square specifying which texture to use through an index that coincides with the loading order.