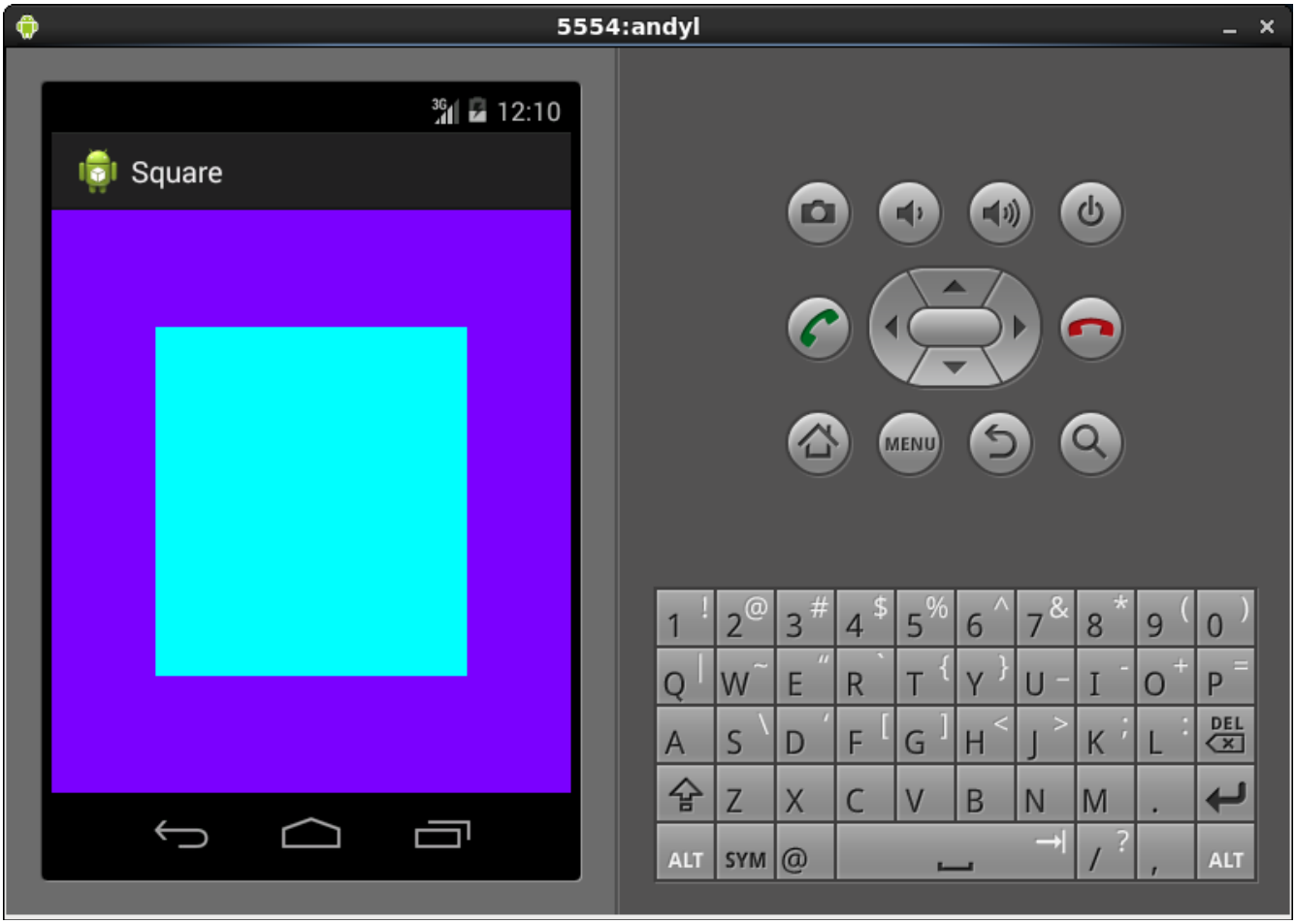
Yazhuo Liu
Lab 2

Write an Android graphics program that draws a yellow square using OpenGL ES 1.X.



Code:

```
package cse520.square;

import java.nio.ByteBuffer;
import java.nio.ByteOrder;

import java.nio.FloatBuffer;

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

import android.opengl.GLSurfaceView;

public class HelloESRenderer implements GLSurfaceView.Renderer {

      private FloatBuffer triangle;
      private FloatBuffer triangle2;

    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        // Set the background frame color to blue
```

```
        gl.glClearColor(0.5f, 0.0f, 1.0f, 1.0f);
        // initialize the triangle vertex array
        initShapes();
        // Enable use of vertex arrays
        gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
    }

    public void onDrawFrame(GL10 gl) {
        // Redraw background color
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT | GL10.GL_DEPTH_BUFFER_BIT);
        // Draw the triangle
        gl.glColor4f(0.0f, 1.0f, 1.0f, 0.0f);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, triangle);
        gl.glDrawArrays(GL10.GL_TRIANGLES, 0, 3);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, triangle2);
        gl.glDrawArrays(GL10.GL_TRIANGLES, 0, 3);

    }

    public void onSurfaceChanged(GL10 gl, int width, int height) {
        gl.glViewport(0, 0, width, height);
    }

    private void initShapes(){
        float vertices_1[] = {
            -0.6f, -0.6f, 0,
             0.6f, -0.6f, 0,
            -0.6f,  0.6f, 0
        };

        float vertices_2[] = {
            0.6f, -0.6f, 0,
            0.6f, 0.6f, 0,
            -0.6f, 0.6f, 0
        };

        // initialize vertex Buffer for triangle
        ByteBuffer vbb = ByteBuffer.allocateDirect(
                // (# of coordinate values * 4 bytes per float)
                vertices_1.length * 4);
        vbb.order(ByteOrder.nativeOrder());// use the device hardware's native
byte order
        triangle = vbb.asFloatBuffer();  // create a floating point buffer from
the ByteBuffer
        triangle.put(vertices_1);    // add the coordinates to the FloatBuffer
        triangle.position(0);            // set the buffer to read the first
coordinate

        ByteBuffer vbbb = ByteBuffer.allocateDirect(vertices_2.length * 4);
        vbbb.order(ByteOrder.nativeOrder());// use the device hardware's native
byte order
        triangle2 = vbbb.asFloatBuffer();  // create a floating point buffer from
the ByteBuffer
        triangle2.put(vertices_2);    // add the coordinates to the FloatBuffer
        triangle2.position(0);            // set the buffer to read the first
coordinate
    }
```

Report:

I completed the lab successfully.