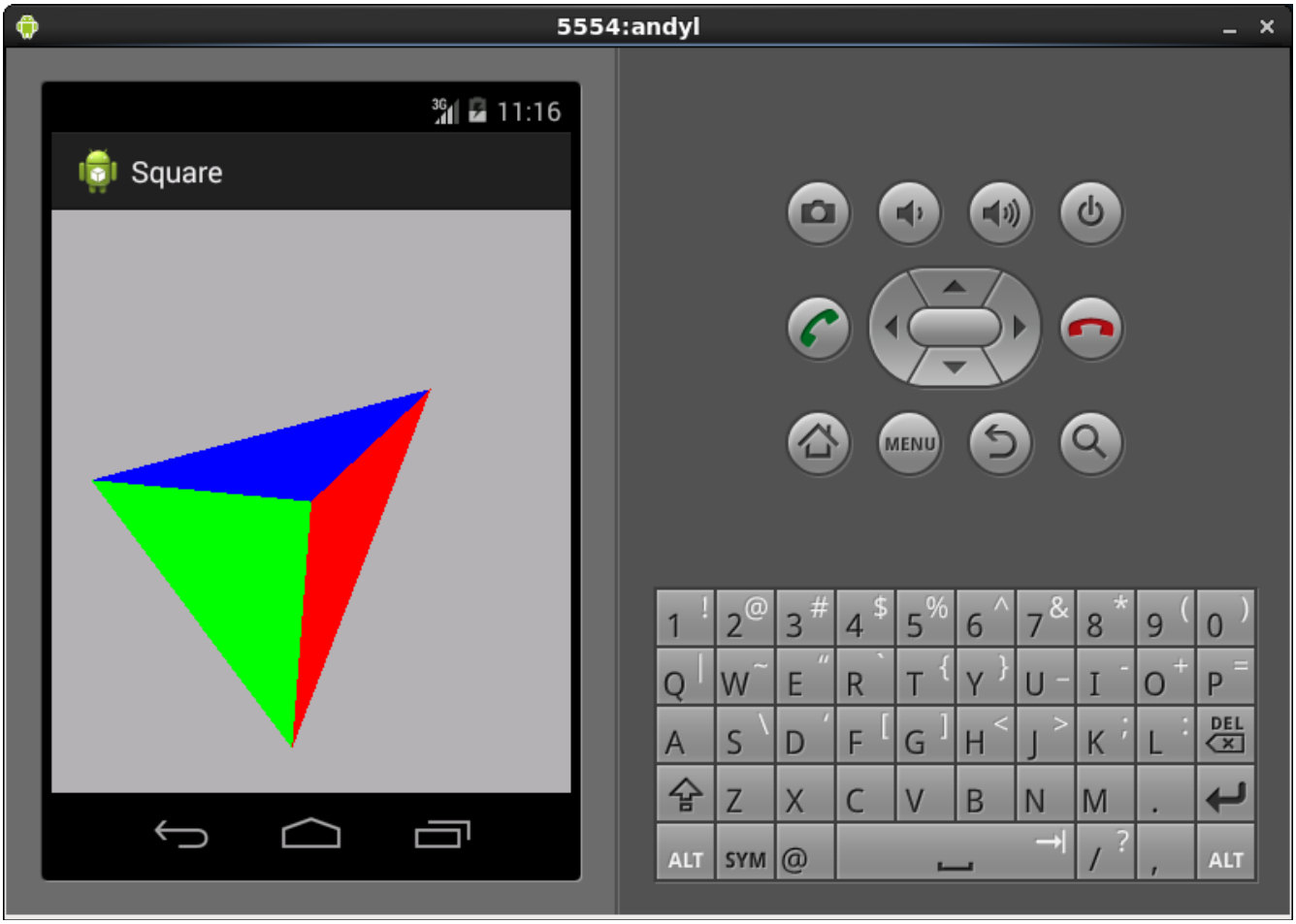
Yazhuo Liu
Lab 3

Write an Android graphics program using OpenGL ES 1X .that renders a colored tetrahedron.



Code:

...

```
public class HelloESRenderer implements GLSurfaceView.Renderer {

      private FloatBuffer triangle;
      private FloatBuffer triangle2;
      private FloatBuffer triangle3;


    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        // Set the background frame color to blue
        gl.glClearColor(0.7f, 0.7f, 0.7f, 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);
        gl.glMatrixMode(GL10.GL_MODELVIEW);
        gl.glLoadIdentity();
        GLU.gluLookAt(gl, 0, 0, 5, 0f, 0f, 0f, 0f, 1.0f, 0.0f);
        gl.glRotatef(-50, 0, 0, 1);

        // Draw the triangle
        gl.glColor4f(0.0f, 1.0f, 0.0f, 0.0f);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, triangle);
        gl.glDrawArrays(GL10.GL_TRIANGLES, 0, 3);
        gl.glColor4f(1.0f, 0.0f, 0.0f, 0.0f);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, triangle2);
        gl.glDrawArrays(GL10.GL_TRIANGLES, 0, 3);
        gl.glColor4f(0.0f, 0.0f, 1.0f, 0.0f);
        gl.glVertexPointer(3, GL10.GL_FLOAT, 0, triangle3);
        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, 0.85f,
                  0.6f, -0.6f, 0.85f,
                  0.0f, 0.6f, 0
      };

      float vertices_2[] = {
                  0.0f, 0.6f, 0,
                  0.6f, -0.6f, 0.85f,
                  0, 0, -0.6f,
      };

      float vertices_3[] = {
                  0, 0.6f, 0,
                  0, 0, -0.6f,
                  -0.6f, -0.6f, 0.85f
      };

        // initialize vertex Buffer for triangle
        ByteBuffer vbb = ByteBuffer.allocateDirect(vertices_1.length * 4);
        vbb.order(ByteOrder.nativeOrder());
        triangle = vbb.asFloatBuffer();
        triangle.put(vertices_1);
        triangle.position(0);

        ByteBuffer vbbb = ByteBuffer.allocateDirect(vertices_2.length * 4);
        vbbb.order(ByteOrder.nativeOrder());
        triangle2 = vbbb.asFloatBuffer();
        triangle2.put(vertices_2);
        triangle2.position(0);

        ByteBuffer vb = ByteBuffer.allocateDirect(vertices_3.length * 4);
        vb.order(ByteOrder.nativeOrder());
```

```
        triangle3 = vb.asFloatBuffer();
        triangle3.put(vertices_3);
        triangle3.position(0);
    }
}
```

Report:

I believe I have completed this lab successfully.