Samuel Marrujo
Saxon Loster
Professor Yu
June 05, 2014
Final Project Report
ObjectPlay

Abstract

The goal of this project is to create manipulative objects through a rotation environment using the Android platform. By creating and using triangles, being able to create and play with multiple objects using a button. Experimenting through the button uses

Introduction

The final project begins as an Android application project, using the tools learned from the classroom setting, and the various labs that were worked with. The idea is to build a cube or more objects through the use of many triangles, and be able to incorporate buttons to interact with the object. The foundation was consisted of using different external methods to create the object, however it proved to be much easier to use triangles. This way, the application can be opened through a variety of different Android versions. Although there were complications in creating the actual application due to the inexperienced use of Eclipse, the project was still a success and a great learning experience in the process.

Software Package

Upon building the application, many different ideas came to thought. At first the concept of creating a game came about, however not knowing the limitations of the Android platform caused a pause in that direction. Therefore, we wanted to implement simply an application that interacts with an object directly. This way, instead of worrying about whether the code actually works or not, the idea presented was to use a cube, and being able to rotate it with the use of buttons. The build introduces the use of buttons that can be pushed to interact with what the user would like to do. While the button’s functions are only 1-dimensional, they prove to be of good use for the usage of the application. The structure of the cube and buttons will be described and expanded upon later on.

Because of our limited knowledge of the Android platform, this task actually proved to be quite difficult. With the transition to Java and Eclipse from a C++ setting with OpenGL, there were many functions that were possible, and also that were not possible. For example, the use of GL_POLYGON and GL_QUADS simply does not exist for understandable reasons of finite memory in Android. Due to this, the use of GL_TRIANGLES and creating a cube based off of triangles proved useful into our creation of the cube used in this Android application. Furthermore, the adjustment to passing variables through different activities was quite confusing until we got the hang of it.

```
            public void draw(GL10 gl) {
            gl.glFrontFace(GL10.GL_CCW);
            gl.glEnable(GL10.GL_CULL_FACE);
            gl.glCullFace(GL10.GL_BACK);

            gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
            gl.glVertexPointer(3, GL10.GL_FLOAT, 0, vertexBuffer);
            for (int face = 0; face < numFaces; face++) {
                gl.glColor4f(colors[face][0], colors[face][1], colors[face][2],
colors[face][3]);
                gl.glDrawArrays(GL10.GL_TRIANGLE_STRIP, face*4, 4);
            }
            gl.glDisableClientState(GL10.GL_VERTEX_ARRAY);
            gl.glDisable(GL10.GL_CULL_FACE);
        }
```

One of the other struggles that were experienced in the process of this application was the utilization of the buttons in Android. While the concept is simple to implement, the foundation that was built upon when creating the use for the buttons proved to be quite difficult. Once this problem was sorted out though, the progress of the program was a much smoother transition. However, another instance arose when trying to use the buttons in a way we desired. This was simply the use of a rotation mechanic, and was the main focus of the interaction with the object.

Instead of approaching the matrix way, we wanted to utilize the glRotatef function as a method of rotating the object in the direction we desire. Fortunately, by the usage of a passing of variables, this was able to be successfully implemented. The use of a “speed” variable is also created to show the user that the rate of rotation can be increased or decreased without much trouble at all. To our advantage, we increased the speed on all variables to check for crashes and did not experience such a problem. Because of this, the way the code was built upon brings signs of relief for our group.

```
gl.glRotatef( angleCubeX , 1.0f , 0.0f , 0.0f );
gl.glRotatef( angleCubeY , 0.0f , 1.0f , 0.0f );
gl.glRotatef( angleCubeZ , 0.0f , 0.0f , 1.0f );
```

The main utilities of this program consist of buttons that interact with the cube (object). By using these buttons, it enables the ability of rotating the cube in the desired direction. For example, pushing the “x +” button will increase the speed of rotation by 1 in the x-direction. Upon pushing the button again, it increases the speed and allows the cube to rotate at a higher frequency. These buttons exist for each axis direction for three dimensions, which are x, y, and z. Furthermore, there exists a button to slow down the rate of rotation. That is, pushing the “x -” button will slow down the rate of rotation until at which it reaches the 0 point, at which it will then begin to rotate in the other direction. Now, because there are buttons for multiple directions, pushing “x +” then “y +” will cause the cube to rotate in both axes. Furthermore, there is a “Reset” button for each direction that will put the speed of rotation for each direction to be 0. Because of this, it can become addicting to play around with if the user is in a “bored” state.

```
private static float angleCubeX = 0;
private static float angleCubeY = 0;
private static float angleCubeZ = 0;
private static float speedCubeX = 0.0f;
private static float speedCubeY = 0.0f;
private static float speedCubeZ = 0.0f;

public MyRenderer( Context context , View parent ) {
    myShape = new Shape();
    this.parent = parent;
    addListenerOnButton();
    }

public void addListenerOnButton(){
    buttonXn = ( Button ) parent.findViewById( R.id.button1 );
    buttonXp = ( Button ) parent.findViewById( R.id.button2 );
    buttonYn = ( Button ) parent.findViewById( R.id.button3 );
    buttonYp = ( Button ) parent.findViewById( R.id.button4 );
    buttonZn = ( Button ) parent.findViewById( R.id.button5 );
    buttonZp = ( Button ) parent.findViewById( R.id.button6 );

    buttonXn.setOnClickListener( new OnClickListener(){
        @Override public void onClick( View view ){
            speedCubeX = speedCubeX - 1.0f;
            }
        } );
    buttonXp.setOnClickListener( new OnClickListener(){
        @Override public void onClick( View view ){
            speedCubeX = speedCubeX + 1.0f;
            }
        } );
    buttonYn.setOnClickListener( new OnClickListener(){
        @Override public void onClick( View view ){
            speedCubeY = speedCubeY - 1.0f;
            }
        } );
    buttonYp.setOnClickListener( new OnClickListener(){
        @Override public void onClick( View view ){
            speedCubeY = speedCubeY + 1.0f;
            }
        } );
    buttonZn.setOnClickListener( new OnClickListener(){
        @Override public void onClick( View view ){
            speedCubeZ = speedCubeZ - 1.0f;
            }
        } );
    buttonZp.setOnClickListener( new OnClickListener(){
        @Override public void onClick( View view ){
            speedCubeZ = speedCubeZ + 1.0f;
            }
        } );
    }
```

The idea of the application is that when people become bored during their daily activities, they want something mindless to do. So the idea for an application that uses pointless mechanics, that creates a time-wasting application seemed fitting for the job. Due to this, the focus on the project was to create an object that can be endlessly played with until the desired results are gained. While the application is not one to be played addictively like “Flappy Bird,” it still holds it’s own by being an application that can be interacted with continuously.

The other idea for this project was to be able to create manipulative objects in the Android environment. By creating this project, this is successfully achieved and we are now able to move forward with some of our future aspirations in the project.

For future implementation of this project, being able to interact with more objects would definitely be desirable. Furthermore, the inclusion of more buttons that do different things would be more fun. For example, moving the object without causing the object to remove from the screen would be a desirable approach. Allowing the ability to play with multiple colors is also a possible way to have the user use a more creation-aspect to the application. However, with our small knowledge in Android, this proved to be extremely difficult. Despite the lesser knowledge, there were other interesting and fun ideas to be used for this project.

Another idea that can be used towards this project is to create a game that uses the rotation mechanic as a way of swinging a bat to hit a ball. Originally this Baseball concept was the plan, but in Android this proved to be a much more difficult task due to the creation and use of many models, and the programming portion of the game. Furthermore, the interaction of the polygons and how the ball would react with the bat is a task that is not fitting for the knowledge known of the Eclipse platform. Nevertheless, the future implementations using this rotation concept can be further expanded into other future projects.

In conclusion, this project has definitely caused this group some frustration from errors in Eclipse, but overall the experience gained by implementing an Android application is nothing short of an amazing learning adventure. The amount of information that was gained by doing the project was a very large amount. However, being able to implement the future ideas would be a much more desirable of a task. The idea of the project is to become more familiar with the Eclipse platform, and be able to familiarize ourselves with different ways of approaching an Android application. Because of this project, we feel we were able to accomplish a successful learning experience entirely.

References

Used everything from prior knowledge of C++ and the use of the Lecture Notes from the CSE520 class website. Further knowledge is used from the OpenGL programming guide on the official OpenGL website, although the content is similar anyways. The idea for the project came from the idea of an interactive game that can waste considerable amounts of time.