Exploring Animation and OpenGL In Android

Satya Komatineni
Active Intellect, Inc.
(http://www.satyakomatineni.com)

In Pursuit of Education and Entertainment on hand held computers

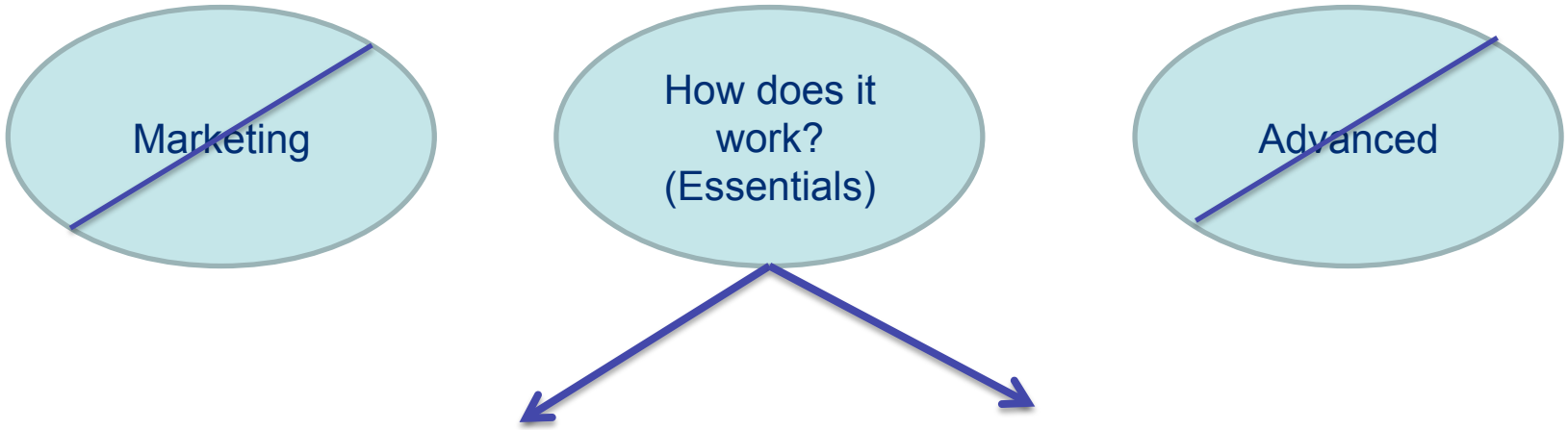
About me

•Aspire Web - A little known Java/Web fluent platform (my toil)

•KnowledgeFolders - Research, Documentation, Productivity, Education, My Public Presence, Book Writing

•Pro Android – A Mobile book that I have co-authored with Sayed Hashimi for Apress.

Why this session?



- Tell me how it works (Animation and OpenGL)
- Let me get a sense of how easy or difficult is this to do
- What is my learning curve?
- How can you help me to get up to speed?
- Does this give me a lay of the land?
- Do I have a reason to get excited about it?

My High Level Plan

- Say a little bit about Android (really little)
- Talk about Animation
- Talk about OpenGL
- Resources

In Animation

- Frame Animation
- List Animation
- View Animation

In OpenGL…

- Android OpenGL architecture
- Drawing Primitives
- Setting the Camera and Understanding Coordinates
- Draw a Triangle and Play with Camera
- Draw any Polygon
- Apply Textures
- Draw Multiple times
- What helped me!

Intro To Android

What is Android SDK?

- Linux based OS under the hood
- Java based Custom Virtual Machine called Dalvik
- At release level 1.5 and aiming for 2.0 in December
- Supports a device emulator well integrated into Eclipse
- Integrated browser similar to Chrome (Webkit)
- 2D and 3D (through OpenGL ES 1.0)
- Full media support including recording (audio and video)
- GSM Telephony
- Bluetooth, EDGE, 3G, and WiFi
- Camera, GPS, compass, and accelerometer
- Animation based on frames and also tweening

Notable Pieces of Android

- A Modern Declarative UI (through XML and Java)
- REST/Content Providers and SQLite
- Well integrated, simple Eclipse based IDE
- Lot of built-in support for Animation and OpenGL ES
- A comprehensive “ONE” package to deal with (Very programmer friendly)
- Very highly multi-threaded

Hello World

```
public class HelloAndroid extends Activity
{
public void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://
schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:text="Hello, Android"/>
```

```
public final class R {
    public static final class attr {
    };
    public static final class drawable {
        public static final int icon=0x7f020000;
    };
    public static final class layout {
        public static final int main=0x7f030000;
    };
    public static final class string {
        public static final int app_name=0x7f040000;
    };
};
```

Layout XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingLeft="6dip"
    android:paddingRight="6dip"
    android:paddingBottom="3dip">

    <EditText android:id="@+id/title"
        android:maxLines="1"
        android:layout_marginTop="2dip"
        android:layout_width="wrap_content"
      android:ems="25"
        android:layout_height="wrap_content"
        android:autoText="true"
        android:capitalize="sentences"
        android:scrollHorizontally="true" />

    <Button android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/button_ok" />

</LinearLayout>
```

Manifest.xml

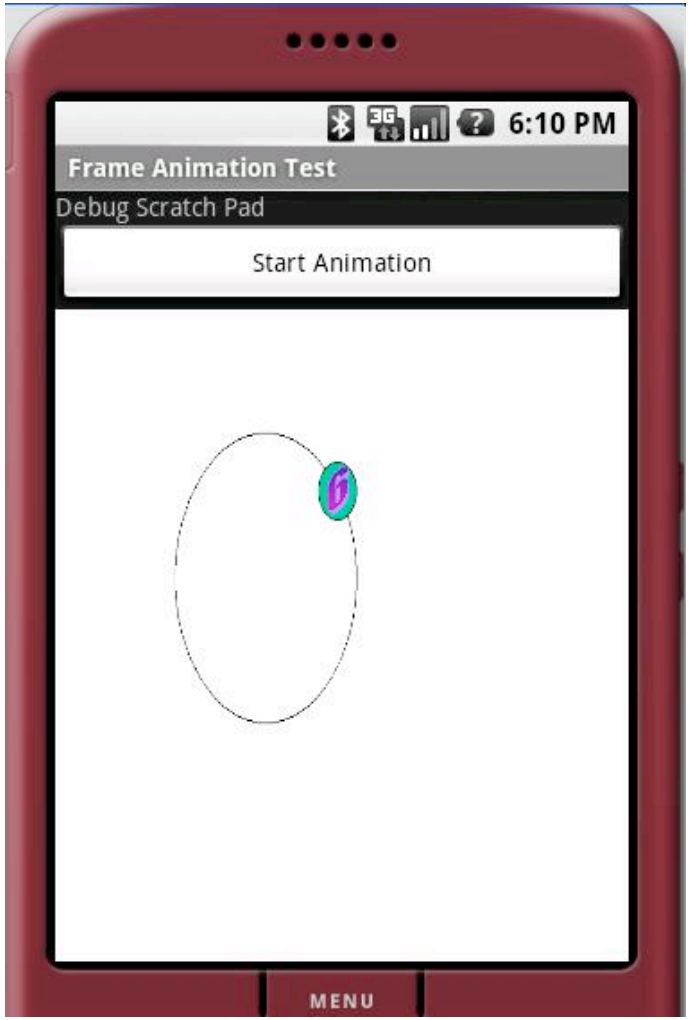
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.notepad"
>
    <application android:icon="@drawable/app_notes"
        android:label="@string/app_name"
    >
        <provider android:name="NotePadProvider"
            android:authorities="com.google.provider.NotePad"
        />

        <activity android:name="NotesList" android:label="@string/title_notes_list">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <action android:name="android.intent.action.EDIT" />
                <action android:name="android.intent.action.PICK" />
                <category android:name="android.intent.category.DEFAULT" />
                <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.GET_CONTENT" />
                <category android:name="android.intent.category.DEFAULT" />
                <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
            </intent-filter>
        </activity>

        <activity android:name="NoteEditor"
            android:theme="@android:style/Theme.Light"
            android:label="@string/title_note"
        >
```

Packages

- App
- Content
- Content.pm
- Content.res
- Database
- Database.sqlite
- Graphics
- Graphics.drawable
- Graphics.drawable.shapes
- Hardware
- Location
- Media
- Net
- Net.http
- Net.wifi
- Opengl
- Os
- Preference
- Provider
- Sax
- Telephony
- Telephony.gsm
- Text
- Text.method
- Text.style
- Text.util
- View
- View.animation
- Webkit
- Widget
- Com.google.android.maps
- Java.*
- Javax.*
- Junit.*
- Org.apache.http.*
- Org.json
- Org.xml.*

Animation

Frame Animation



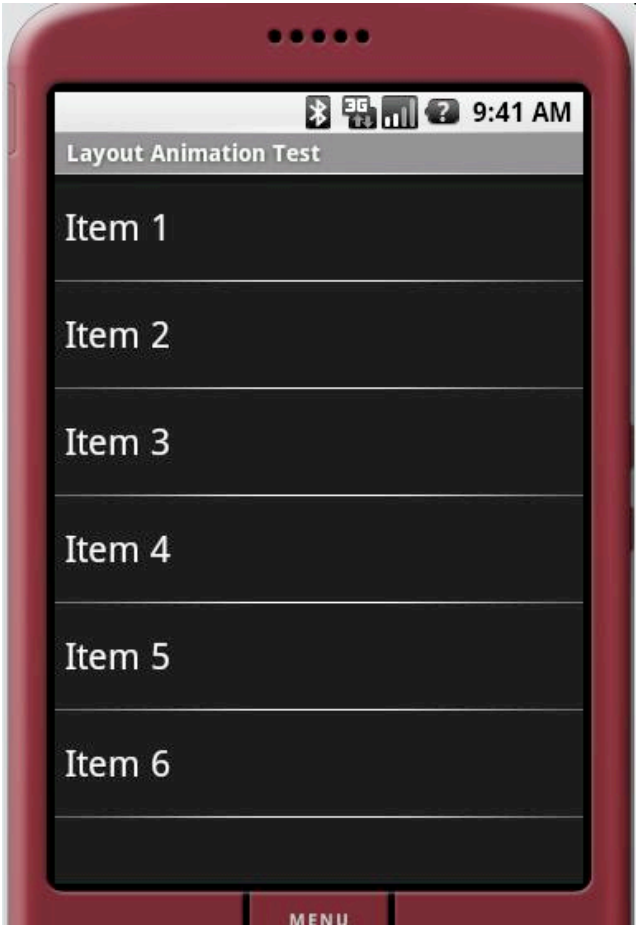
Frame Animation: XML Definition

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
        android:oneshot="false">
   <item android:drawable="@drawable/numbers11" android:duration="50" />
   <item android:drawable="@drawable/numbers12" android:duration="50" />
   <item android:drawable="@drawable/numbers13" android:duration="50" />
   <item android:drawable="@drawable/numbers14" android:duration="50" />
   <item android:drawable="@drawable/numbers15" android:duration="50" />
   <item android:drawable="@drawable/numbers16" android:duration="50" />
   <item android:drawable="@drawable/numbers17" android:duration="50" />
   <item android:drawable="@drawable/numbers18" android:duration="50" />
   <item android:drawable="@drawable/numbers19" android:duration="50" />
</animation-list>
```

Frame Animation: Java code

```
private void animate()    {
    ImageView imgView = (ImageView)findViewById(R.id.imageView);
    imgView.setVisibility(ImageView.VISIBLE);
    imgView.setBackgroundResource(R.drawable.frame_animation);

    AnimationDrawable frameAnimation =
            (AnimationDrawable) imgView.getBackground();

    frameAnimation.running() ?
            frameAnimation.stop():
            frameAnimation.start();
}
```

A ViewGroup or List Animation (Also called Layout Animation)



List Animation: XML Definition

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
……
>
    <ListView
        android:id="@+id/list_view_id"
        android:persistentDrawingCache="animation|scrolling"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layoutAnimation="@anim/list_layout_controller" />
        />
</LinearLayout>
```

List Animation: adapter

```
<layoutAnimation xmlns:android="http://schemas.android.com/apk/res/android"
        android:delay="30%"
        android:animationOrder="reverse"
        android:animation="@anim/alpha_translate" />
```

Alpha and Translate Animation: An Example

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
            android:interpolator="@android:anim/accelerate_interpolator">

   <translate android:fromYDelta="-100%"
              android:toYDelta="0"
              android:duration="500" />

   <alpha android:fromAlpha="0.0"
          android:toAlpha="1.0"
          android:duration="500" />
</set>
```

Rotate and Scale Animation: An Example

```
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
      android:interpolator="@android:anim/accelerate_interpolator"
      android:fromDegrees="0.0"
      android:toDegrees="360"
      android:pivotX="50%"
      android:pivotY="50%"
      android:duration="500" />

<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/accelerate_interpolator">
  <scale
       android:fromXScale="1”       android:toXScale="1"
       android:fromYScale="0.1”     android:toYScale="1.0"
       android:duration="500”       android:pivotX="50%"
       android:pivotY="50%”         android:startOffset="100" />
</set>
```

Interpolators: Definition and Java code

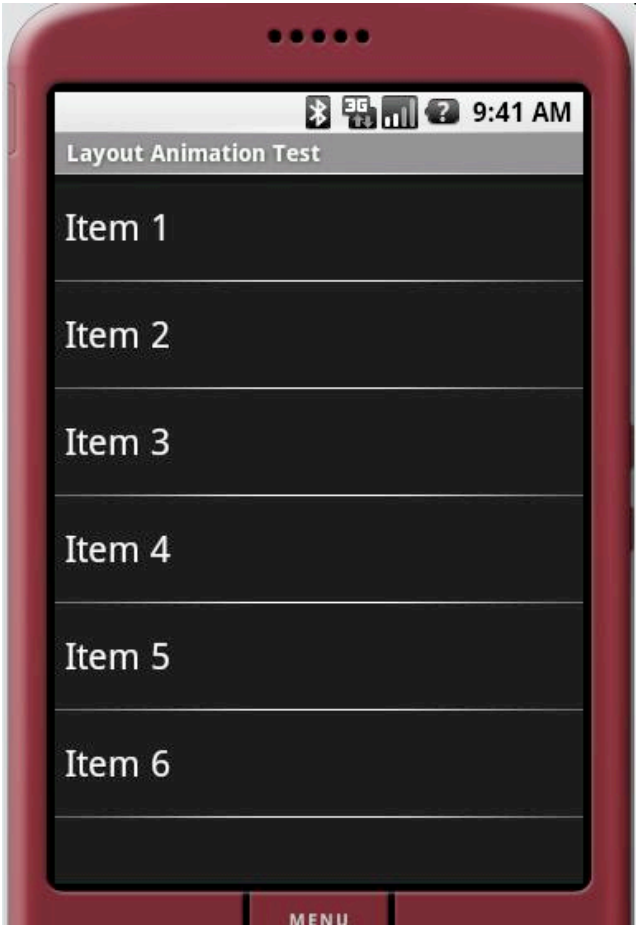
```
<accelerateInterpolator
  xmlns:android="http://schemas.android.com/apk/res/android"
  factor="1" />
```

```
public float getInterpolation(float input)
{
   if (mFactor == 1.0f)    {
      return (float)(input * input);
   }
   else    {
      return (float)Math.pow(input, 2 * mFactor);
   }
}
```

Available Interpolators

AccelerateDecelerateInterpolator
AccelerateInterpolator
CycleInterpolator
DecelerateInterpolator
LinearInterpolator

View Animation: A different take



Attach Animation to a View

```
View  someView;
someView.startAnimation(new ViewAnimation());
```

View Animation

```
public class ViewAnimation extends Animation
{
  public void initialize(int width, int height, int parentWidth, int parentHeight) {
        super.initialize(width, height, parentWidth, parentHeight);
        setDuration(2500);
        setFillAfter(true);
        setInterpolator(new LinearInterpolator());
    }
  protected void applyTransformation(float interpolatedTime, Transformation t)
   {
        final Matrix matrix = t.getMatrix();
        matrix.setScale(interpolatedTime, interpolatedTime);
        matrix.preTranslate(-centerX, -centerY);
        matrix.postTranslate(centerX, centerY);    }
}
```

Applying a 2D Camera

```
protected void applyTransformation(float interpolatedTime, Transformation t)
{
        final Matrix matrix = t.getMatrix();
        camera.save();
        camera.translate(0.0f, 0.0f, (1300 - 1300.0f * interpolatedTime));
        camera.rotateY(360 * interpolatedTime);
        camera.getMatrix(matrix);

        matrix.preTranslate(-centerX, -centerY);
        matrix.postTranslate(centerX, centerY);
        camera.restore();
}
```

OpenGL ES 1.0

How I Justify My Frustration with OpenGL

When the Goal of Each Step is Fun,
the Journey in the End is Profitable.

Key APIs to understand

glVertexPointer
glDrawElements

gluLookAt
glFrustum
glViewPort

glTexturePointer

Key Observations

1. There is a lot of OpenGL documentation
2. There is very little GOOD OpenGL documentation
3. Most books overlook the basic concepts
4. The flat “c” api makes it very difficult to understand the OpenGL state machine

Beginner Suggestions

1. Get a good list of resources, including OpenGL books
2. Understand the OpenGL Camera paradigm really well
3. You cannot use Java buffers use native “nio” buffers
4. Nio buffers use a part of the memory that is outside the JVM but they do clean themselves when gc is run
5. Know that you can call “glDrawElements” multiple times by changing transformation matrices to draw multiple figures
6. Typically Textures and Surfaces are drawn together so you will have to use item 4

An attempt at a triangle

```
//Clear the surface of any color
gl.glClear(gl.GL_COLOR_BUFFER_BIT);

//Set the current color
gl.glColor4f(1.0f, 0, 0, 0.5f);

//There are three points (floats) in the specificed buffer pointer
gl.glVertexPointer(3, GL10.GL_FLOAT, 0, mFVertexBuffer);

//Draw them assuming they form triangles
//there are 3 points
gl.glDrawElements(GL10.GL_TRIANGLE_STRIP, 3,
    GL10.GL_UNSIGNED_SHORT, mIndexBuffer);
```

Primitive Shapes

GL_TRIANGLE_STRIP
GL_POINTS
GL_LINE_STRIP
GL_LINES
GL_LINE_LOOP
GL_TRIANGLES
GL_TRIANGLE_FAN

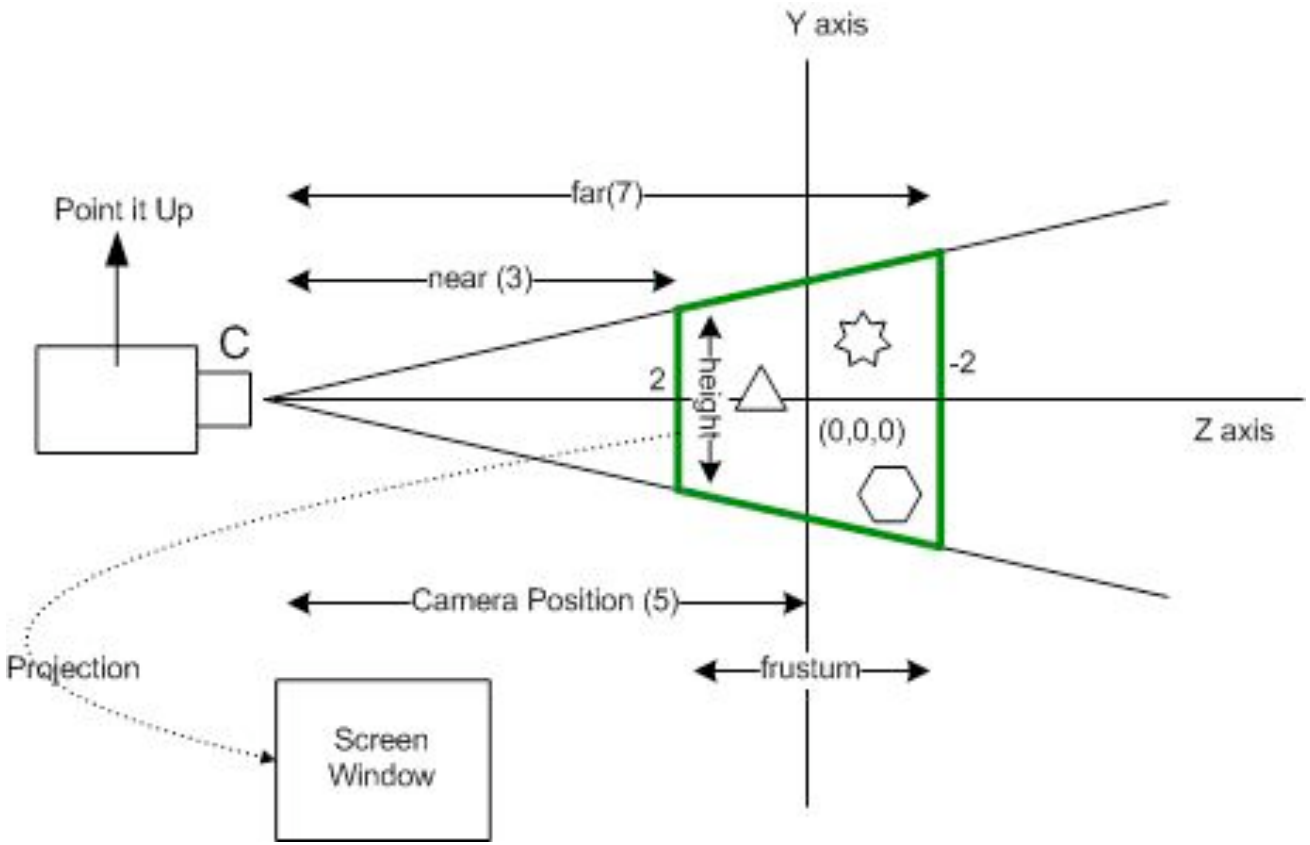
How you use NIO buffers: A vertex Example

```
java.nio.ByteBuffer vbb = ByteBuffer.allocateDirect(VERTS * 3 * 4);
vbb.order(ByteOrder.nativeOrder());
java.nio.FloatButffer mFVertexBuffer = vbb.asFloatBuffer();

float[] coords = {
        -0.5f, -0.5f, 0, // (x1,y1,z1)
         0.5f, -0.5f, 0,
         0.0f,  0.5f, 0 };

for (int i = 0; i < VERTS; i++) {
    for(int j = 0; j < 3; j++) {
        mFVertexBuffer.put(coords[i*3+j]);
}}
mFVertexBuffer.position(0);
```

How you use NIO buffers: An Index Example

```
java.nio.ByteBuffer ibb = ByteBuffer.allocateDirect(VERTS * 2);
ibb.order(ByteOrder.nativeOrder());
java.nio.ShortBuffer mIndexBuffer = ibb.asShortBuffer();

short[] myIndecesArray = {0,1,2};
for (int i=0;i<3;i++)
{
   mIndexBuffer.put(myIndecesArray[i]);
}
mIndexBuffer.position(0);
```

Camera and Coordinates



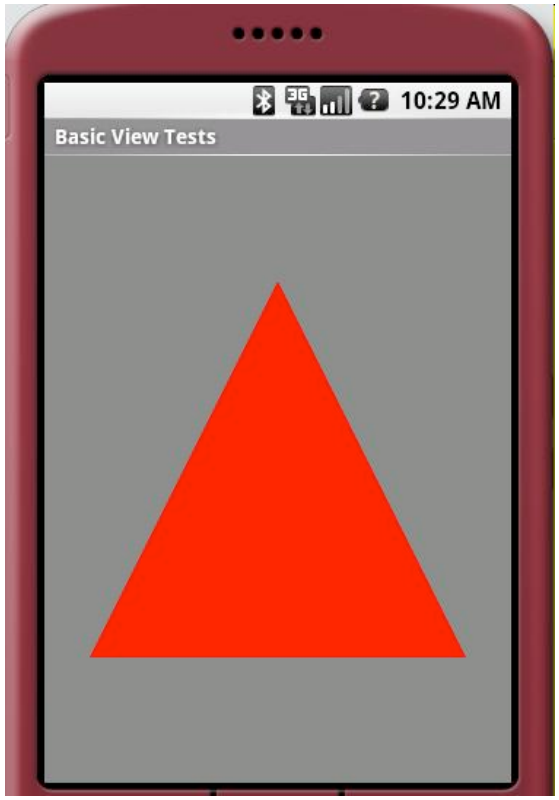
glFrustum
gluLookAt
glViewPort

How you use NIO buffers: An Index Example

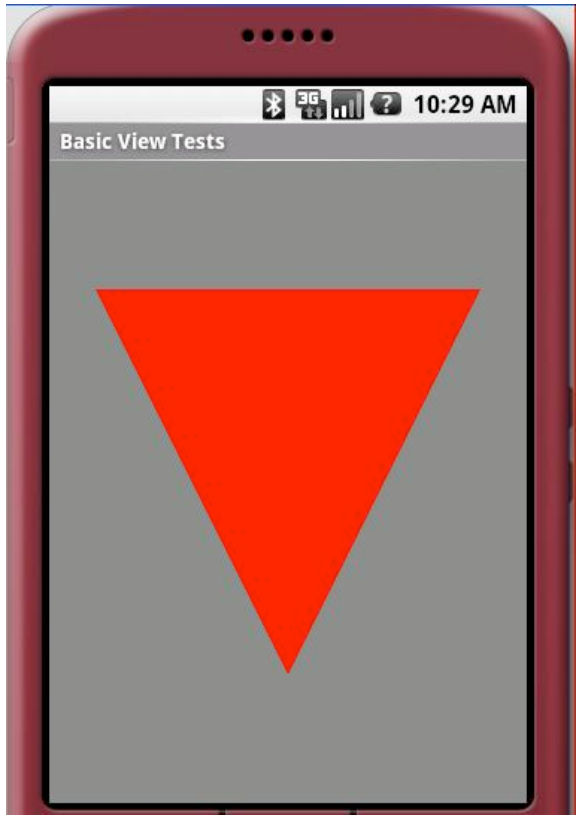
```
//Look at the screen (origin) from 5 units away from the front of the screen
GLU.gluLookAt(gl, 0,0,5,    0,0,0,   0,1,0);

//Set the height to 2 units and depth to 4 units
gl.glFrustumf(-ratio, ratio, -1, 1, 3, 7);

//normal window stuff
gl.glViewport(0, 0, w, h);
```

Changing Your Scene Settings

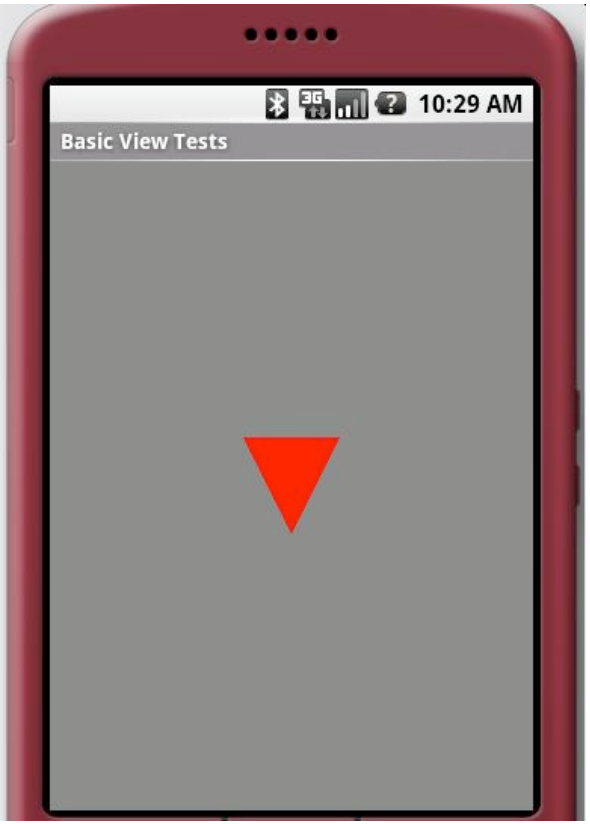
Up right gluLookAt



Look down gluLookAt



4 times frustum



Putting it All Together

```
public class OpenGL15TestHarnessActivity extends Activity
{
   private GLSurfaceView mTestHarness;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      mTestHarness = new GLSurfaceView(this);
      mTestHarness.setEGLConfigChooser(false);
      mTestHarness.setRenderer(new SimpleTriangleRenderer(this));
      mTestHarness.setRenderMode(
            GLSurfaceView.RENDERMODE_WHEN_DIRTY);
      setContentView(mTestHarness);
   }
```

What is a Renderer

```
public static interface GLSurfaceView.Renderer
{
   void onDrawFrame(GL10 gl);
   void onSuraceChanged(GL10 gl, int width, int height);
   void onSurfaceCreated(GL10 gl, EGLConfig config);
}
```

What goes into onSurfaceCreated

```
public abstract class AbstractRenderer
   implements GLSurfaceView.Renderer
{
     public void onSurfaceCreated(GL10 gl, EGLConfig eglConfig) {
      gl.glDisable(GL10.GL_DITHER);
      gl.glHint(GL10.GL_PERSPECTIVE_CORRECTION_HINT,
                              GL10.GL_FASTEST);
      gl.glClearColor(.5f, .5f, .5f, 1);
      gl.glShadeModel(GL10.GL_SMOOTH);
      gl.glEnable(GL10.GL_DEPTH_TEST);
   }
```

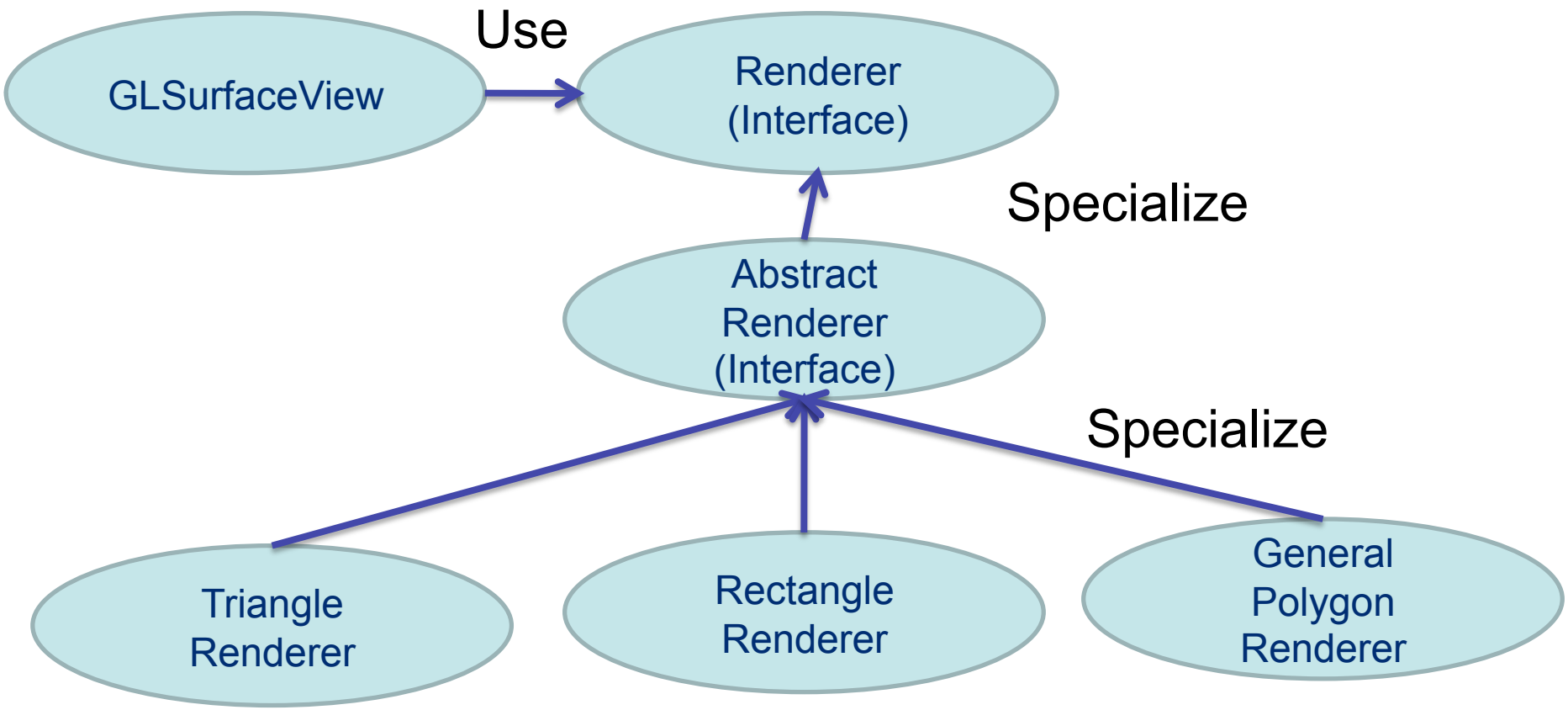
What goes into onSurfaceChanged

```
public void onSurfaceChanged(GL10 gl, int w, int h) {
    gl.glViewport(0, 0, w, h);
    float ratio = (float) w / h;
    gl.glMatrixMode(GL10.GL_PROJECTION);
    gl.glLoadIdentity();
    gl.glFrustumf(-ratio, ratio, -1, 1, 3, 7);
 }
```

What goes into onDrawFrame

```
public void onDrawFrame(GL10 gl)
  {
    gl.glDisable(GL10.GL_DITHER);
    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.glEnableClientState(GL10.GL_VERTEX_ARRAY);

    //Actual drawing can happen in a concrete class
    draw(gl);
  }
```

Brief Architecture of GLSurfaceView



Drawing Any Polygon

```
RegularPolygon t = new RegularPolygon(0,0,0,1,sides);
this.mFVertexBuffer = t.getVertexBuffer();
this.mIndexBuffer = t.getIndexBuffer();
this.numOfIndecies = t.getNumberOfIndecies();
this.mFVertexBuffer.position(0);
this.mIndexBuffer.position(0);

gl.glColor4f(1.0f, 0, 0, 0.5f);
gl.glVertexPointer(3, GL10.GL_FLOAT, 0, mFVertexBuffer);
gl.glDrawElements(GL10.GL_TRIANGLES, this.numOfIndecies,
            GL10.GL_UNSIGNED_SHORT, mIndexBuffer);
```

I can animate that polygon

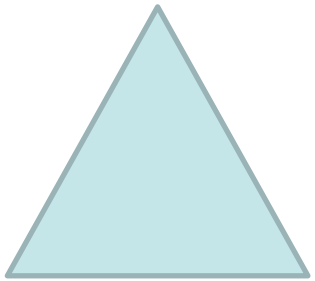
```
long curtime = SystemClock.uptimeMillis();
if ((curtime - prevtime) > 2000) {
    prevtime = curtime;
    sides += 1;
     if (sides > 20) { sides = 3; }
     this.prepareBuffers(sides);
}

gl.glColor4f(1.0f, 0, 0, 0.5f);
gl.glVertexPointer(3, GL10.GL_FLOAT, 0, mFVertexBuffer);
gl.glDrawElements(GL10.GL_TRIANGLES, this.numOfIndecies,
          GL10.GL_UNSIGNED_SHORT, mIndexBuffer);
```

Adding Texture

(1,1)

(0,0)

bitmap



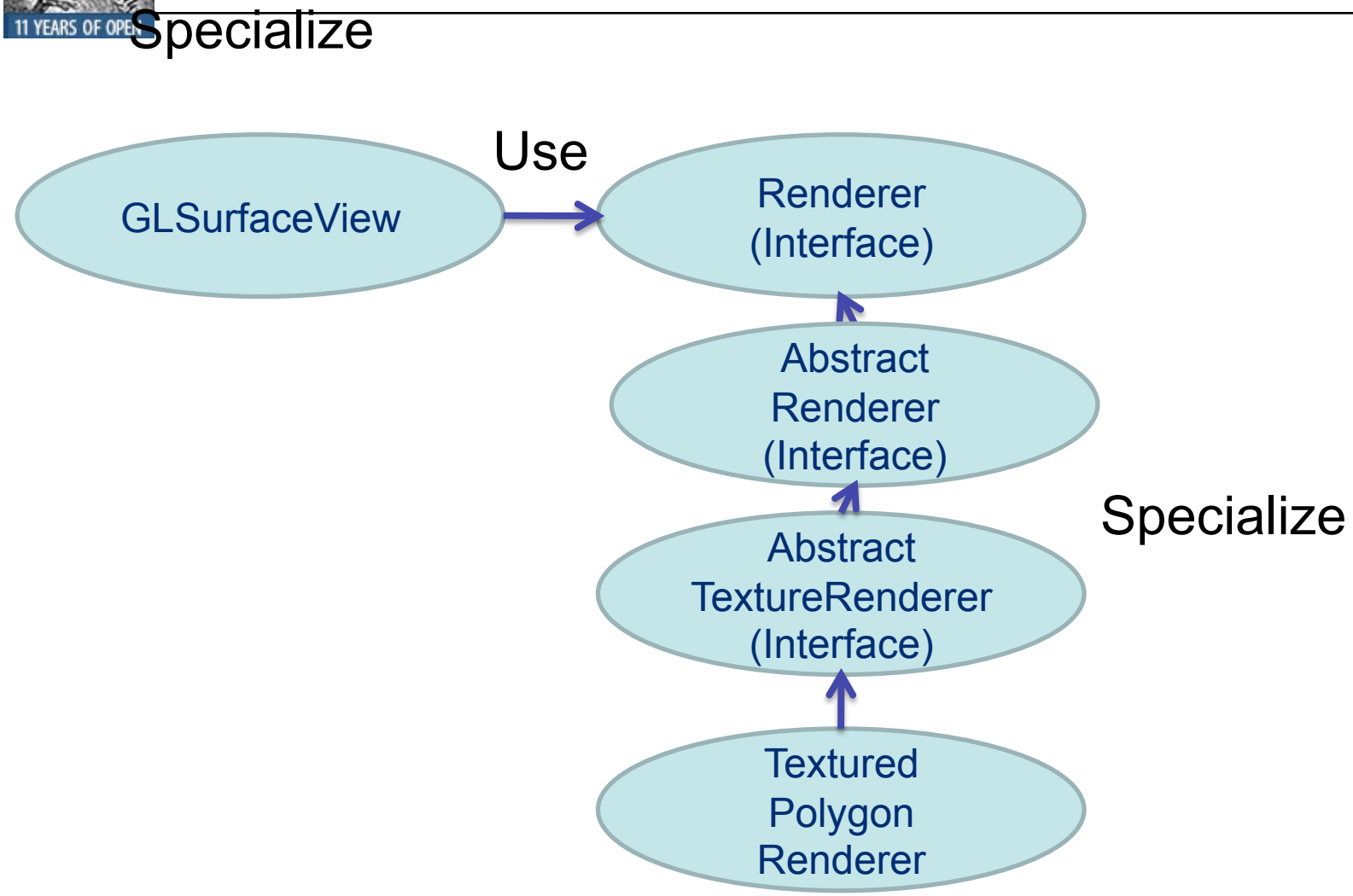
Object

Figure out texture coordinates

(1,1)

(0,0)

Abstract Textured Renderer



Setup Your Texture in OnSurfaceCreated

```
int[] textures = new int[1];        gl.glGenTextures(1, textures, 0);
mTextureID = textures[0];
gl.glBindTexture(GL10.GL_TEXTURE_2D, mTextureID);
gl.glTexParameterf(GL10.GL_TEXTURE_2D, GL10.GL_TEXTURE_MIN_FILTER,
            GL10.GL_NEAREST);
gl.glTexParameterf(GL10.GL_TEXTURE_2D,
            GL10.GL_TEXTURE_MAG_FILTER,
            GL10.GL_LINEAR);

gl.glTexParameterf(GL10.GL_TEXTURE_2D, GL10.GL_TEXTURE_WRAP_S,
            GL10.GL_CLAMP_TO_EDGE);
gl.glTexParameterf(GL10.GL_TEXTURE_2D, GL10.GL_TEXTURE_WRAP_T,
            GL10.GL_CLAMP_TO_EDGE);

gl.glTexEnvf(GL10.GL_TEXTURE_ENV, GL10.GL_TEXTURE_ENV_MODE,
            GL10.GL_REPLACE);
```

Setup Your Texture in OnSurfaceCreated

```
InputStream is = mContext.getResources()
            .openRawResource(this.mImageResourceId);
    Bitmap bitmap;
    try {
        bitmap = BitmapFactory.decodeStream(is);
    } finally {
        try {
            is.close();
        } catch(IOException e) {
            // Ignore.
        }
    }

    GLUtils.texImage2D(GL10.GL_TEXTURE_2D, 0, bitmap, 0);
```

A Textured Polygon

```
RegularPolygon t = new RegularPolygon(0,0,0,0.5f,sides);
this.mFVertexBuffer = t.getVertexBuffer();
this.mFTextureBuffer = t.getTextureBuffer();
this.mIndexBuffer = t.getIndexBuffer();
this.mFVertexBuffer.position(0);
this.mIndexBuffer.position(0);
this.mFTextureBuffer.position(0);

gl.glEnable(GL10.GL_TEXTURE_2D);
gl.glVertexPointer(3, GL10.GL_FLOAT, 0, mFVertexBuffer);
gl.glTexCoordPointer(2, GL10.GL_FLOAT, 0, mFTextureBuffer);

glDrawElements(….);
```

Drawing Two Textured Polygons

```
gl.glEnable(GL10.GL_TEXTURE_2D);
gl.glVertexPointer(3, GL10.GL_FLOAT, 0, mFVertexBuffer);
gl.glTexCoordPointer(2, GL10.GL_FLOAT, 0, mFTextureBuffer);

gl.glPushMatrix();
gl.glScalef(0.5f, 0.5f, 1.0f);
gl.glTranslatef(0.5f,0, 0);
gl.glDrawElements(GL10.GL_TRIANGLES, this.numOfIndecies,
        GL10.GL_UNSIGNED_SHORT, mIndexBuffer);

gl.glPopMatrix();gl.glPushMatrix();
gl.glScalef(0.5f, 0.5f, 1.0f);   gl.glTranslatef(-0.5f,0, 0);
gl.glDrawElements(GL10.GL_TRIANGLES, this.numOfIndecies,
          GL10.GL_UNSIGNED_SHORT, mIndexBuffer);
gl.glPopMatrix();
```

Drawing Two Textured Polygons

Resources 1/2

My Home Page where I am documenting Android as I explore it
http://www.satyakomatineni.com

This following book Pro Android talks about Android and also some of the content covered here.
http://www.satyakomatineni.com/akc/display?url=DisplayNoteIMPURL&reportId=3186&ownerUserId=satya

Khrons OpenGL ES api (very handy)
http://www.khronos.org/opengles/documentation/opengles1_0/html/index.html

OpenGL Red Book
http://www.glprogramming.com/red/

OpenGL Course Material from Wayne Cochran, WSU, Vancouver
http://ezekiel.vancouver.wsu.edu/~cs442/

Resources 2/2

My Notes on OpenGL
http://www.knowledgefolders.com/akc/display?url=NotesIMPTitlesURL&ownerUserId=satya&folderName=OpenGL&order_by_format=news

My Notes on Android 1.5
http://www.knowledgefolders.com/akc/servlet/DisplayServlet?url=NotesIMPTitlesURL&ownerUserId=satya&folderName=android+1.5&order_by_format=news

Mobile Sessions from Google I/O Conference
http://code.google.com/events/io/sessions.html#mobile

Writing Real Time Games for Android – Chris Pruett
http://code.google.com/events/io/sessions/WritingRealTimeGamesAndroid.html

Thank You