Android Graphics

- Custom 2D graphics library
- OpenGL ES 1.0 for high performance 3D graphcs

- The design of an application and the APIs used depend on the graphical demands:
  - static graphical application
  - dynamic interactive 2D and 3D rendering for games

AVD320480

- For development purposes, create an AVD out of the 3.2” QVGA using Android 4.2.2

- This will give us a small screen for initial development that makes it quicker to check for things like intersecting a window edge with a sprite

Adding Graphics

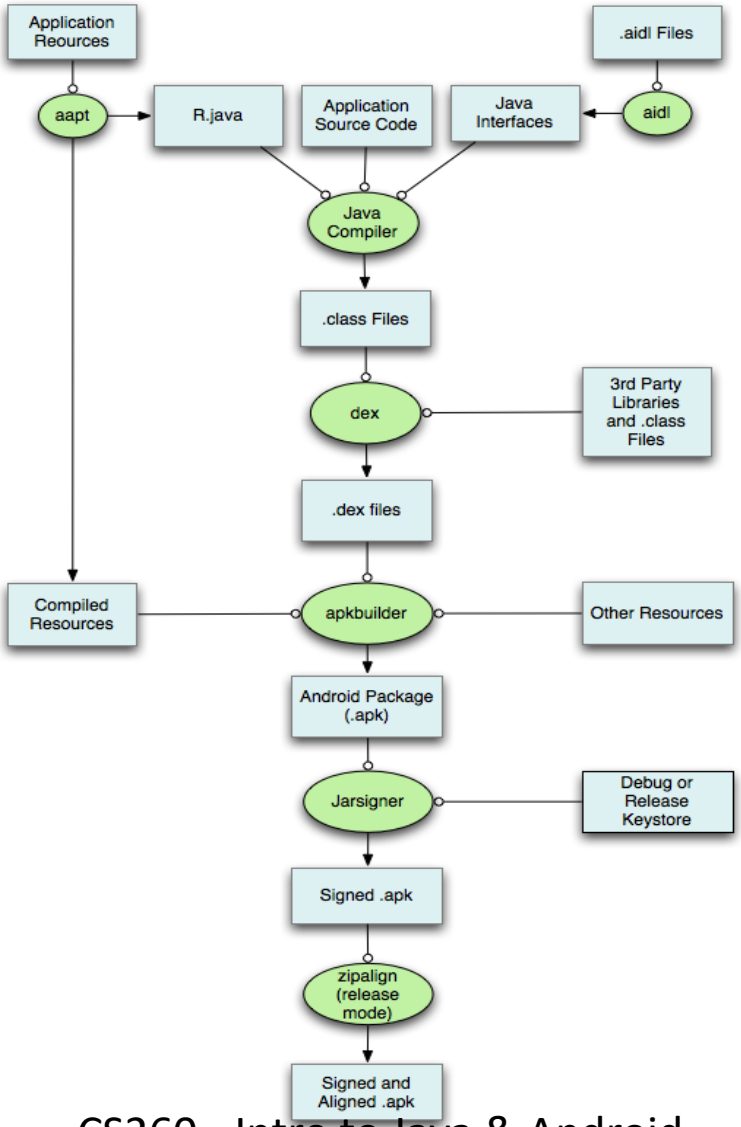
- Always externalize application resources such as images & strings for easier program maintenance
- Depending on the app, device specific resources are maintained in specific folders
- At runtime, Android uses the appropriate resource
- Referencing an image (PNG (preferred), JPG (acceptable), GIF (discouraged)) is the easiest way to add graphics

Adding Graphics

- IMPORTANT
  - Images placed in res/drawable may be optimized with lossless compression by the aapt (Android Asset Packaging Tool)
  - Images placed in the res/raw folder are not optimized

```
MyProject/
    src/
        MyActivity.java
    res/
        drawable/
            graphic.png
        layout/
            main.xml
            info.xml
        mipmap/
            icon.png
        values/
            strings.xml
```

Notice aapt


Application Reources
.aidl Files
aapt
R.java
Application Source Code
Java Interfaces
aidl
Java Compiler
.class Files
dex
3rd Party Libraries and .class Files
.dex files
Compiled Resources
apkbuilder
Other Resources
Android Package (.apk)
Jarsigner
Debug or Release Keystore
Signed .apk
zipalign (release mode)
Signed and Aligned .apk

2D Graphics

- Drawing 2D graphics is done in one of two ways:
  1. Draw the graphics/animations into a View and let Android's View hierarchy take care of the drawing process
  2. Draw the graphics/animation directly to the Canvas by calling the appropriate class's draw() method passing a Canvas

Draw Graphics into a View

```
1  package cs.pacificu.edu.drawintoview;
2
3  import android.graphics.Color;
4  import android.support.v7.app.AppCompatActivity;
5  import android.os.Bundle;
6  import android.view.Display;
7  import android.view.Window;
8  import android.view.WindowManager;
9
10 public class MainActivity extends AppCompatActivity
11 {
12   private Display mDisplay;
13   private GraphicsView mGraphicsView;
14
15   @Override
16   protected void onCreate (Bundle savedInstanceState)
17   {
18     super.onCreate (savedInstanceState);
19
20     mGraphicsView = new GraphicsView ( context: this);
21     mGraphicsView.setBackgroundColor (Color.BLACK);
22     setContentView (mGraphicsView);
23   }
24 }
```

Draw Graphics into a View

```
12  public class GraphicsView extends View
13  {
14
15    private float mRectangleWidth, mRectangleHeight;
16    private final Paint mBackground = new Paint (), mDarklines = new Paint ();
17    private static final int mNUMBER_OF_RECTANGLES = 3;
18    private final Rect mSelectedRectangle = new Rect ();
19    private final Paint mSelectedRectanglePaint = new Paint ();
20    private final Paint mText = new Paint (Paint.ANTI_ALIAS_FLAG);
21
22    public GraphicsView (Context context)
23    {
24      super (context);
25      setFocusable (true);
26      setFocusableInTouchMode (true);
27      mSelectedRectanglePaint.setColor (Color.BLUE);
28      mDarklines.setColor (Color.BLACK);
29      mBackground.setColor (Color.LTGRAY);
30
31
32    }
```

2D Graphics

- Draw the graphics/animation directly to the Canvas by calling the appropriate class's draw() method passing a Canvas

- In the CS260-01Public\2018\Animation folder, you will find:
  - GraphicsView.java
  - Sprite.java
  - Three png files

Problem

- You are to create an Android project called Animation that has three classes:
  - GraphicsView
  - Sprite
  - MainActivity
- Place these three classes in Animation in a package called cs.pacificu.edu.animation
- Copy the code from GraphicsView and Sprite into their respective classes in Animation
- Place the three png files in res\drawable

Problem

- Here is the main activity code

```
private Display mDisplay;
private GraphicsView mGraphicsView;

@Override
protected void onCreate (Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);

   this.requestWindowFeature(Window.FEATURE_NO_TITLE);
   this.getWindow ().setFlags
            (WindowManager.LayoutParams.FLAG_FULLSCREEN,
                    WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

Problem

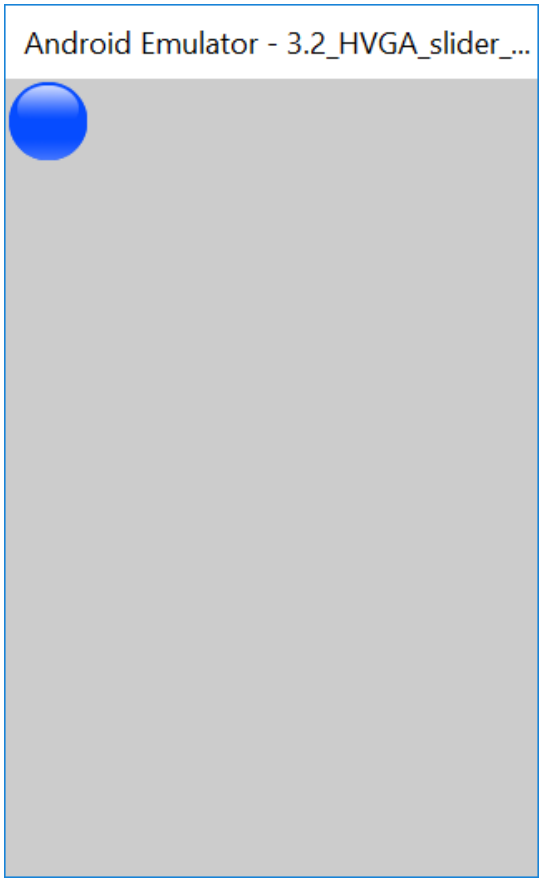
- Here is the main activity code

```
        WindowManager window = getWindowManager ();
        mDisplay = window.getDefaultDisplay ();

        mGraphicsView = new GraphicsView ( context: this, mDisplay);
        mGraphicsView.setBackgroundColor (Color.LTGRAY);
        setContentView (mGraphicsView);
    }
```

Result

- A correctly running Animation project produces



Animation Problem

- Create a new class MovingSprite that subclasses Sprite
- Create a MovingSprite and have the sprite move on the screen

Animation Problem

- Capture an onTouch event
- When the user touches the screen, add another ball on the screen at the position touched. The ball color is to be random and the starting direction is to be random.
- Use polymorphism to move blue, green, and yellow balls in different ways