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

Adding Graphics

- Referencing an image (PNG (preferred), JPG (acceptable), GIF (discouraged)) is the easiest way to add graphics
- IMPORTANT
  - Images placed in res/drawable may be optimized with lossless compression by the aapt tool
  - Images placed in the res/raw folder are not optimized

2D Graphics

- Drawing 2D graphics is done in one of two ways:
- Draw the graphics/animations into a View and let Android’s View hierarchy take care of the drawing process

Draw Graphics into a View

```
protected void onCreate(Bundle savedInstanceState)
 {
     super.onCreate(savedInstanceState);

     // Add ImageView to the LinearLayout
     mLinearLayout = new LinearLayout (this);

     // Instantiate an ImageView
     ImageView i = new ImageView(this);
     i.setImageResource(R.drawable.ball_blue);

     // Add the ImageView to the LinearLayout
     mLinearLayout.addView(i);

     setContentView(mLinearLayout);
  }
```

2D Graphics

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

- Grab the BallAnimation application from CS260-01 Public

Animation Problem

1) Get the ball to bounce off of the sides of the window.
2) Using the ArrayList or Vector class, get three different colored balls bouncing off the side of the window.
3) Add the paddle as a fourth object to the bottom of the window.
4) Move the paddle left and right based on the D-Pad arrow. Left is left and right is right.