

## Graphics

# Supporting multiple Screen Densities

- Android devices come with varying screen densities. To support multiple screen densities multiple resources need to be prepared and kept in separate folder.
- Android system recognizes four screen densities namely ldpi, mdpi, hdpi and xhdpi
- Images need to be prepared and kept in respective folders namely drawable-ldpi, drawable-mdpi, drawable-hdpi and drawable-xhdpi
- Layout resource files need to be prepared and kept in layout-ldpi, layout-mdpi, layout-hdpi and layout-xhdpi

- `android.graphics.Canvas` can be used to draw graphics in android. It provides methods to draw oval, rectangle, picture, text, line etc.
- `android.graphics.Paint` is used to draw objects. It holds the information of color and style.

# MainActivity.java

```
package com.example.simplegraphics;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;

public class MainActivity extends Activity {
```

```
DemoView demoview;  
/** Called when the activity is first created. */  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    demoview = new DemoView(this);  
    setContentView(demoview);  
}  
private class DemoView extends View{  
    public DemoView(Context context){  
        super(context);  
    }  
}
```

```
@Override protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
  
    // custom drawing code here  
    Paint paint = new Paint();  
    paint.setStyle(Paint.Style.FILL);  
  
    // make the entire canvas white  
    paint.setColor(Color.WHITE);  
    canvas.drawPaint(paint);  
  
    // draw blue circle with anti aliasing  
    // turned off  
    paint.setAntiAlias(false);  
    paint.setColor(Color.BLUE);  
    canvas.drawCircle(20, 20, 15, paint);  
}
```

```
// draw green circle with anti aliasing  
turned on  
paint.setAntiAlias(true);  
paint.setColor(Color.GREEN);  
canvas.drawCircle(60, 20, 15, paint);  
  
// draw red rectangle with anti aliasing  
turned off  
paint.setAntiAlias(false);  
paint.setColor(Color.RED);  
canvas.drawRect(100, 5, 200, 30, paint);  
// draw the rotated text  
canvas.rotate(-45);  
  
paint.setStyle(Paint.Style.FILL);  
canvas.drawText("Graphics Rotation", 40,  
    180, paint);  
  
//undo the rotate  
//canvas.restore();
```

# Output





# Two ways to drawing Graphics

- Draw the graphics or animations into a View object. Useful when simple graphics.
- Draw your graphics directly to a Canvas. e.g. Video game

# Drawing Directly to a Canvas

- ❶ In the same thread as the UI activity, you create a custom View component, call `invalidate()` and then handle the `onDraw()` callback
- ❷ In a separate thread, wherein you manage a `SurfaceView` and perform draws to the Canvas

# Creating a Canvas from a Bitmap

```
Bitmap b = Bitmap.createBitmap(100, 100, Bitmap.  
    Config.ARGB_8888);  
Canvas c = new Canvas(b);
```