

CS 696 Emerging Web and Mobile Technologies
Spring Semester, 2011
Doc 28 Android Some 2D Graphics
May 5, 2011

Copyright ©, All rights reserved. 2011 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

2D Graphic Options

Draw graphics/animation on Views

- Simple graphics

- Simple animations

Draw on a canvas

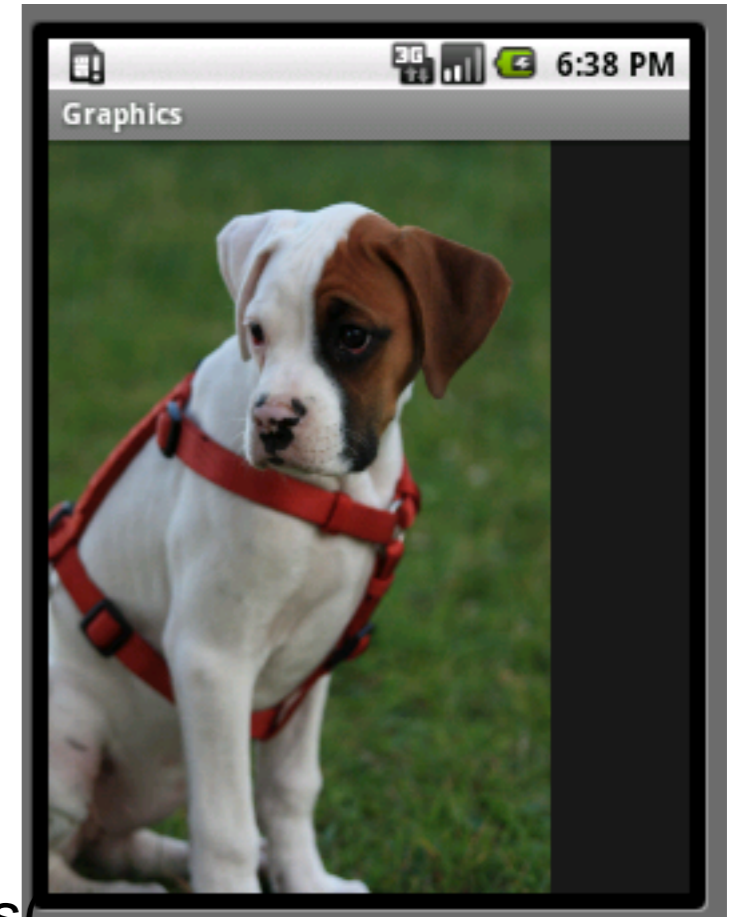
- Graphics need to be redrawn

- Games

Draw graphics/animation on Views

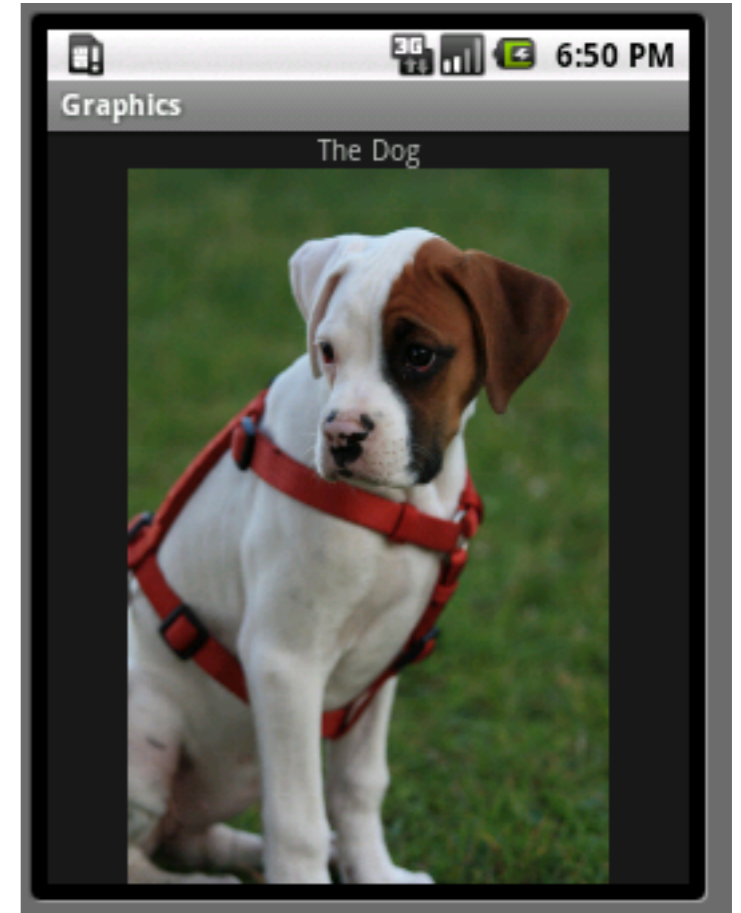
Display a Drawable

```
public class GraphicsExamples extends Activity {  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        LinearLayout linear = new LinearLayout(this);  
  
        ImageView image = new ImageView(this);  
        image.setImageResource(R.drawable.dog);  
        image.setAdjustViewBounds(true);  
        image.setLayoutParams(new Gallery.LayoutParams(  
            LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT));  
  
        linear.addView(image);  
        setContentView(linear);  
    }  
}
```



Display a Drawable - using XML

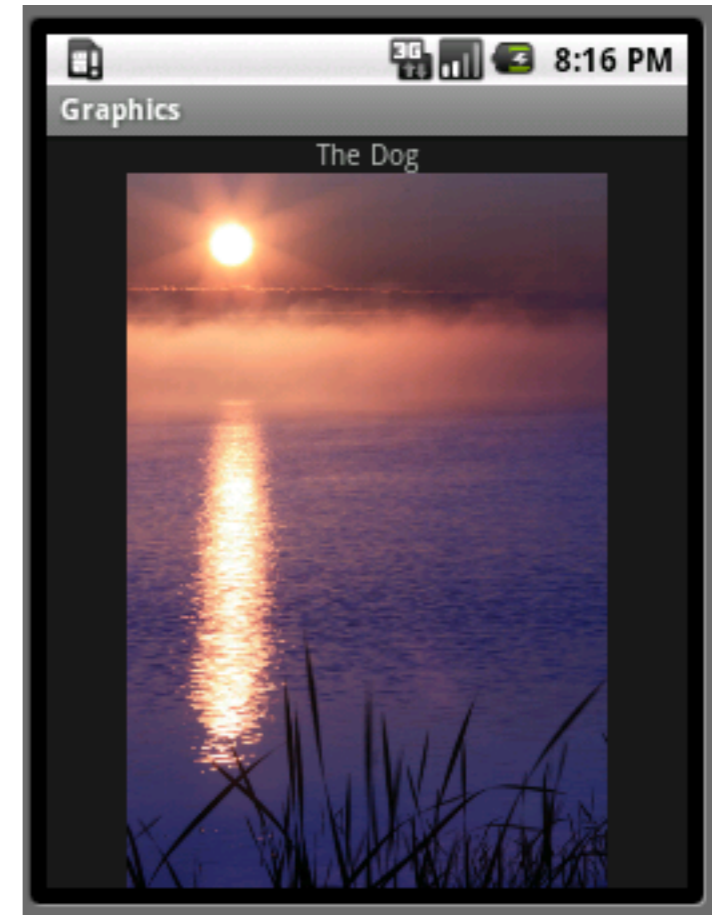
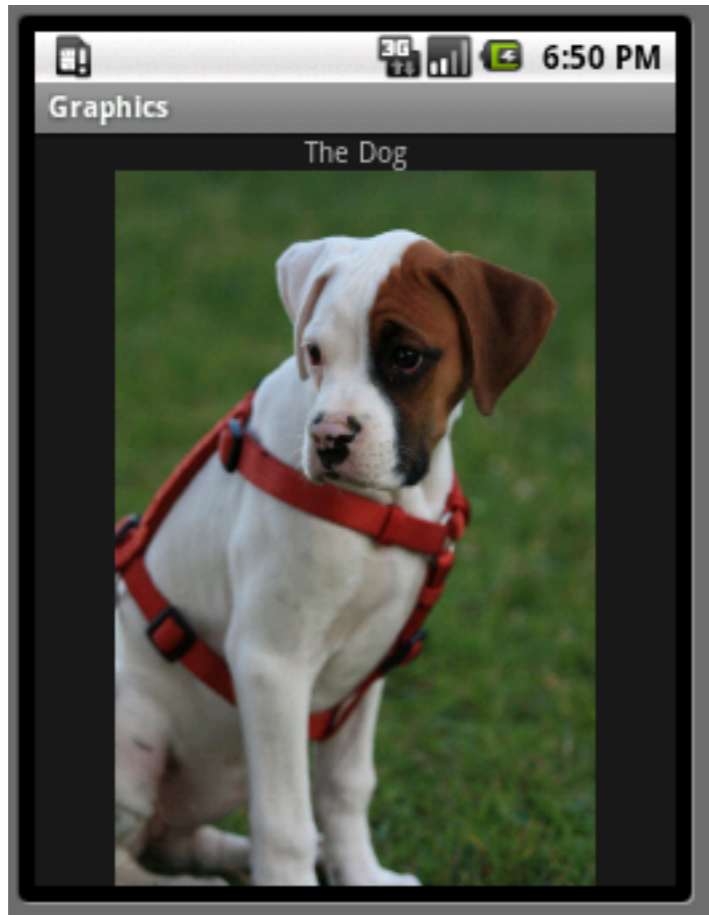
```
public class GraphicsExamples extends Activity {  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```



main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="The Dog"
    android:layout_gravity="center_horizontal"/>
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/dog"
    android:layout_gravity="center_horizontal"/>
</LinearLayout>
```

Transitions



Activity

```
public class GraphicsExamples extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        Resources resources = this.getResources();  
        TransitionDrawable transition = (TransitionDrawable) resources  
            .getDrawable(R.layout.transition);  
        ImageView image = (ImageView) findViewById(R.id.image);  
        image.setImageDrawable(transition);  
        transition.startTransition(5000);  
    }  
}
```


layout/transition.xml

```
<transition xmlns:android="http://schemas.android.com/apk/res/android">  
  <item android:drawable="@drawable/dog"/>  
  <item android:drawable="@drawable/fog"/>  
</transition>
```

Background of a View

```
public class GraphicsExamples extends Activity {
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.main);
```

```
        Resources resources = this.getResources();
```

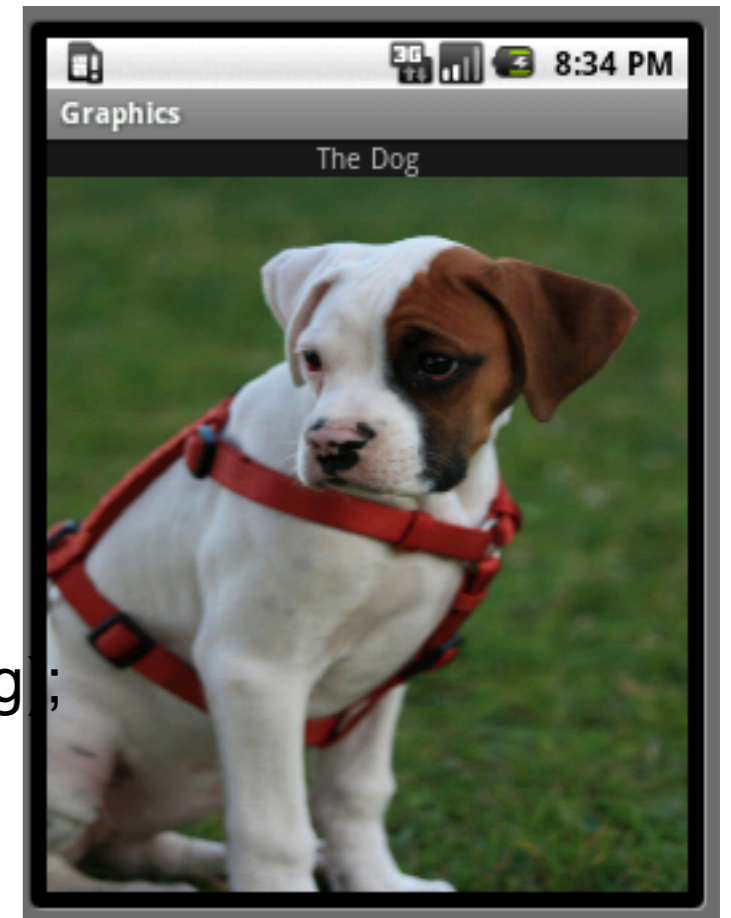
```
        Drawable dog = resources.getDrawable(R.drawable.dog);
```

```
        View image = findViewById(R.id.simpleView);
```

```
        image.setBackgroundDrawable(dog);
```

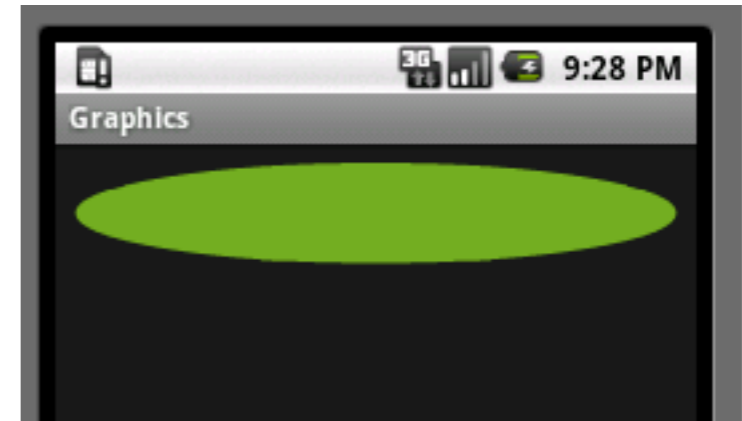
```
    }
```

```
}
```



Drawing an Oval

```
public class GraphicsExamples extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        View shapes = new SimpleDrawing(this);  
        setContentView(shapes);  
    }  
}
```



SimpleDrawing

```
public class SimpleDrawing extends View {  
    private ShapeDrawable oval;  
  
    public SimpleDrawing(Context context) {  
        super(context);  
        int x = 10;  
        int y = 10;  
        int width = 300;  
        int height = 50;  
  
        this.oval = new ShapeDrawable(new OvalShape());  
        this.oval.getPaint().setColor(0xff74AC23);  
        this.oval.setBounds(x, y, x + width, y + height);  
    }  
}
```

SimpleDrawing

```
public SimpleDrawing(Context context, AttributeSet attrs) {  
    super(context, attrs);  
}  
  
@Override  
protected void onDraw(Canvas canvas) {  
    this.oval.draw(canvas);  
}  
}
```

Colors

$(\text{alpha} \ll 24) \mid (\text{red} \ll 16) \mid (\text{green} \ll 8) \mid \text{blue}$

Color value range 0..255

0xff74AC23

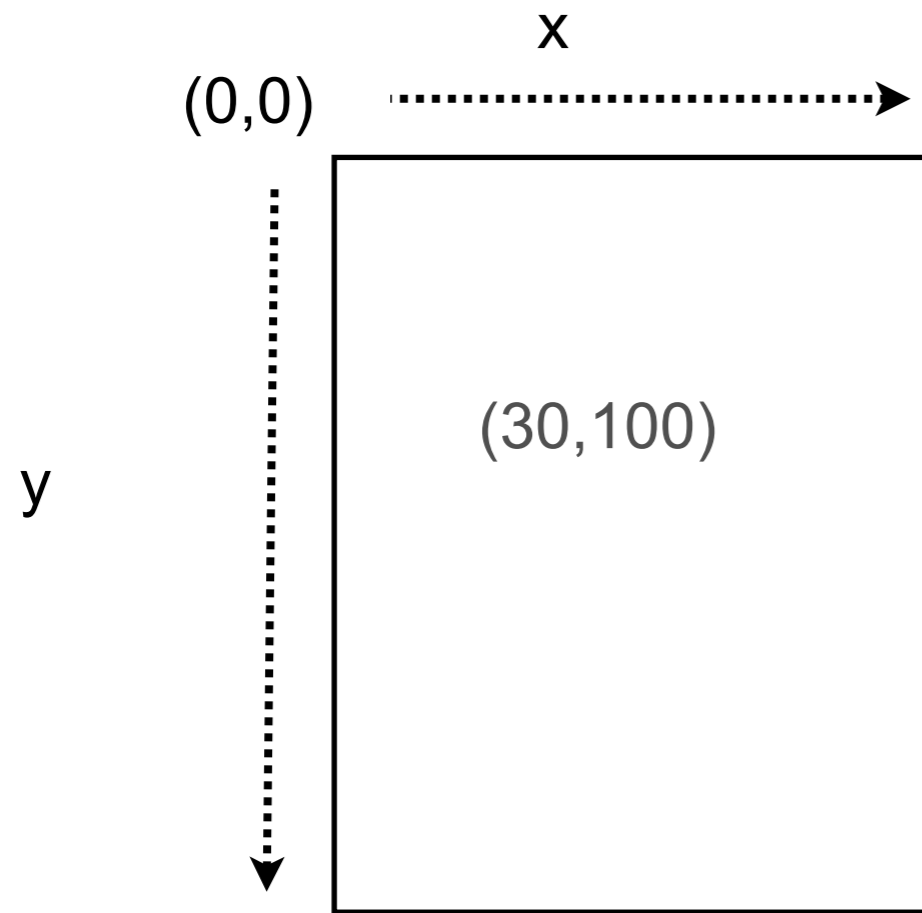
red = 0x74 (116)

blue = 0x23 (35)

alpha = 0xff (255)

green = 0xAC (172)

Coordinates



Shapes

PathShape

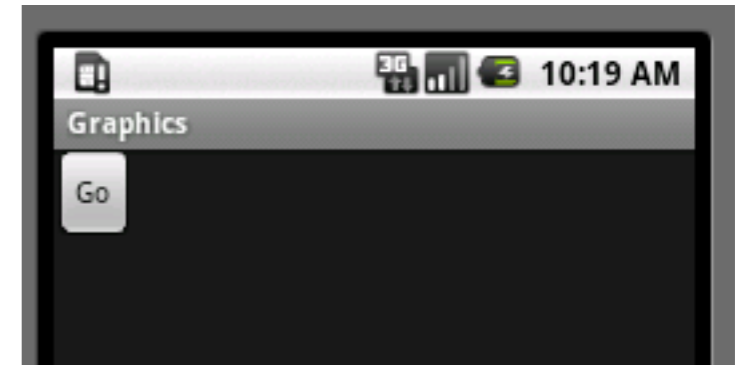
RectShape

ArcShape

OvalShape

RoundRectShape

Tween Animation



res/anim/tween_example.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
  <translate
    android:fromYDelta="0"
    android:toXDelta="800%"
    android:duration="1000" />
  <alpha
    android:fromAlpha="1.0"
    android:toAlpha="0.0"
    android:duration="1000" />
</set>
```

Activity

```
public class GraphicsExamples extends Activity implements View.OnClickListener {  
  
    public void onClick(View v) {  
        Animation shake = AnimationUtils.loadAnimation(this,  
            R.anim.tween_example);  
        findViewById(R.id.go).startAnimation(shake);  
    }  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        View go = findViewById(R.id.go);  
        go.setOnClickListener(this);  
    }  
}
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    >  
  
    <Button  
        android:text="Go"  
        android:id="@+id/go"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"/>  
  
</LinearLayout>
```

android.view.animation.Interpolator

Defines the rate of change of an animation

AccelerateDecelerateInterpolator

AccelerateInterpolator

AnticipateInterpolator

AnticipateOvershootInterpolator

BounceInterpolator

CycleInterpolator

DecelerateInterpolator

LinearInterpolator

OvershootInterpolator

Tween XML Attributes

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
    <translate
        android:fromYDelta="0"
        android:toXDelta="800%"
        android:duration="1000" />
    <alpha
        android:fromAlpha="1.0"
        android:toAlpha="0.0"
        android:duration="1000" />
</set>
```

android.view.animation.Animation

Defined in Animation Class

android:duration
android:fillAfter
android:fillBefore
android:fillEnabled
android:interpolator
android:repeatCount
setRepeatCount(int)
setRepeatMode(int)
android:startOffset
android:zAdjustment

Constructors as XML

AlphaAnimation(float fromAlpha, float toAlpha)

```
<alpha  
  android:fromAlpha="1.0"  
  android:toAlpha="0.0"  
  android:duration="1000" />
```