Basic Graphics

Introduction

- The android platform includes specific classes for drawing 2D graphics, including shapes and images. These classes are packed within the `android.grahpics.drawable` and the `android.view.animation` packages.

The `Drawable` Abstract Class

- The `Drawable` type describes something that can be drawn. You can either use one of the many predefined classes that extend `Drawable` or define your own.
- The simplest way to use a `Drawable` object is to use an `ImageView` object that its resource was set to be the `Drawable` object we want to draw.

Sample

```
package com.abelski.samples;

import android.app.Activity;
import android.content.res.Resources;
import android.graphics.drawable.Drawable;
import android.graphics.drawable.LayerDrawable;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.LinearLayout;

public class LayerDrawableActivity extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        LinearLayout layout = new LinearLayout(this);
        ImageView img = new ImageView(this);
        Resources resources = getResources();
        Drawable layer;
```

Sample

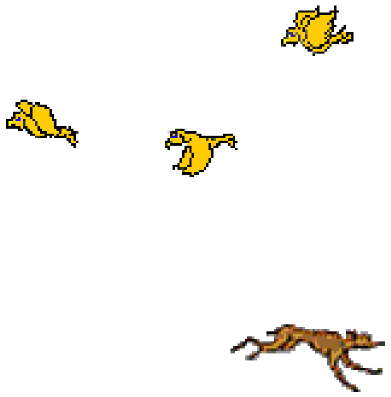
```
        Drawable[] layers = new Drawable[3];
        layer = resources.getDrawable(R.drawable.a);
        layers[0] = layer;
        layer = resources.getDrawable(R.drawable.b);
        layers[1] = layer;
        layer = resources.getDrawable(R.drawable.c);
        layers[2] = layer;
        LayerDrawable layerDrawable = new LayerDrawable(layers);
        img.setImageDrawable(layerDrawable);
        layout.addView(img);
        setContentView(layout);
    }
}
```

Sample

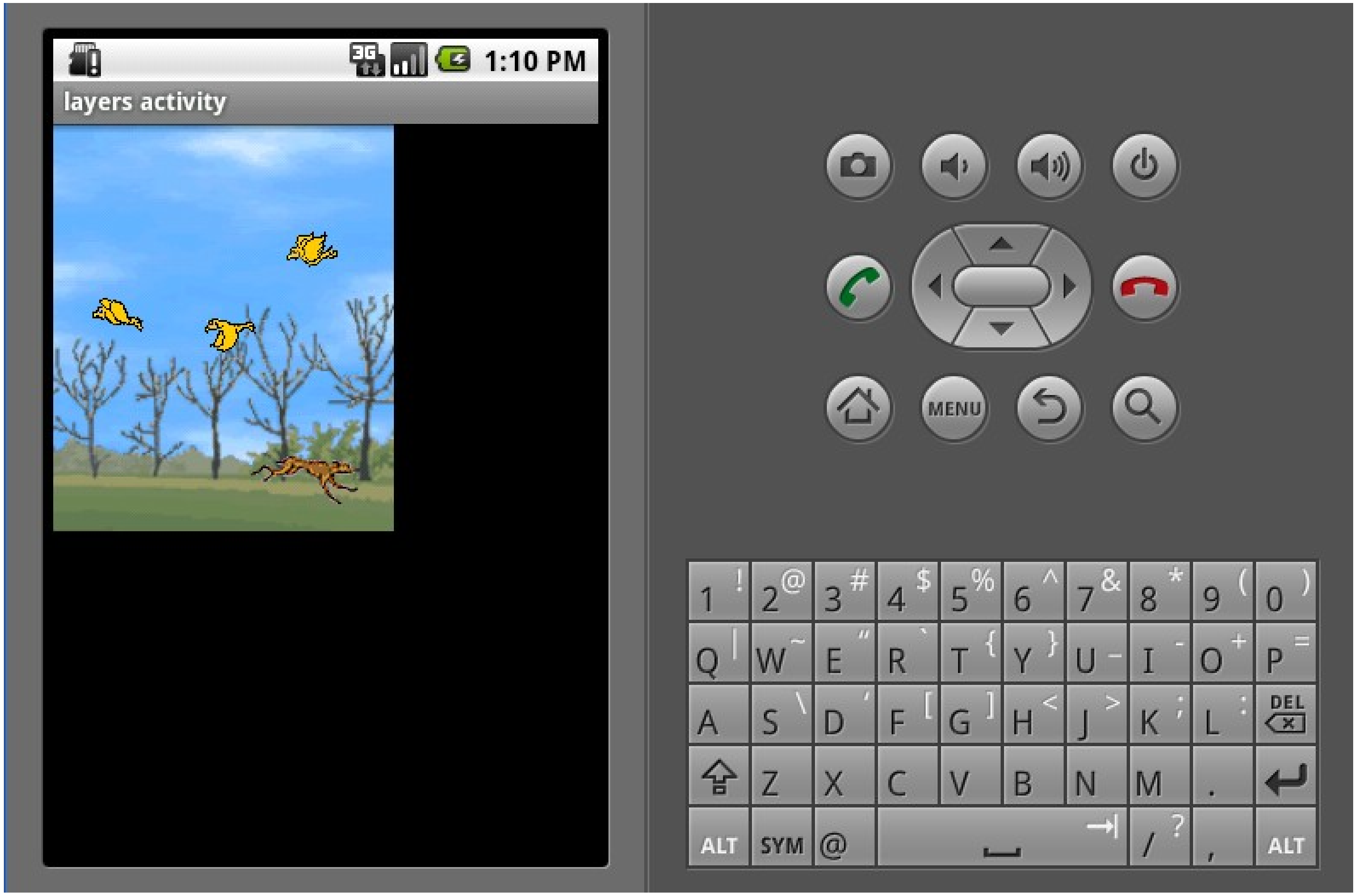
a.png

b.png

c.png



Sample



The `ShapeDrawable` Abstract Class

- The `ShapeDrawable` class extends `Drawable` and allows us to draw primitive shapes.
- Usually, the recommended practice for using this class is to define a new class that extends `View` and override its `onDraw` method in order to draw the shapes we want to draw.

Sample

```
package com.abelski.samples;

import android.app.Activity;
import android.os.Bundle;

public class ShapesActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(new MyView(this));
    }
}
```

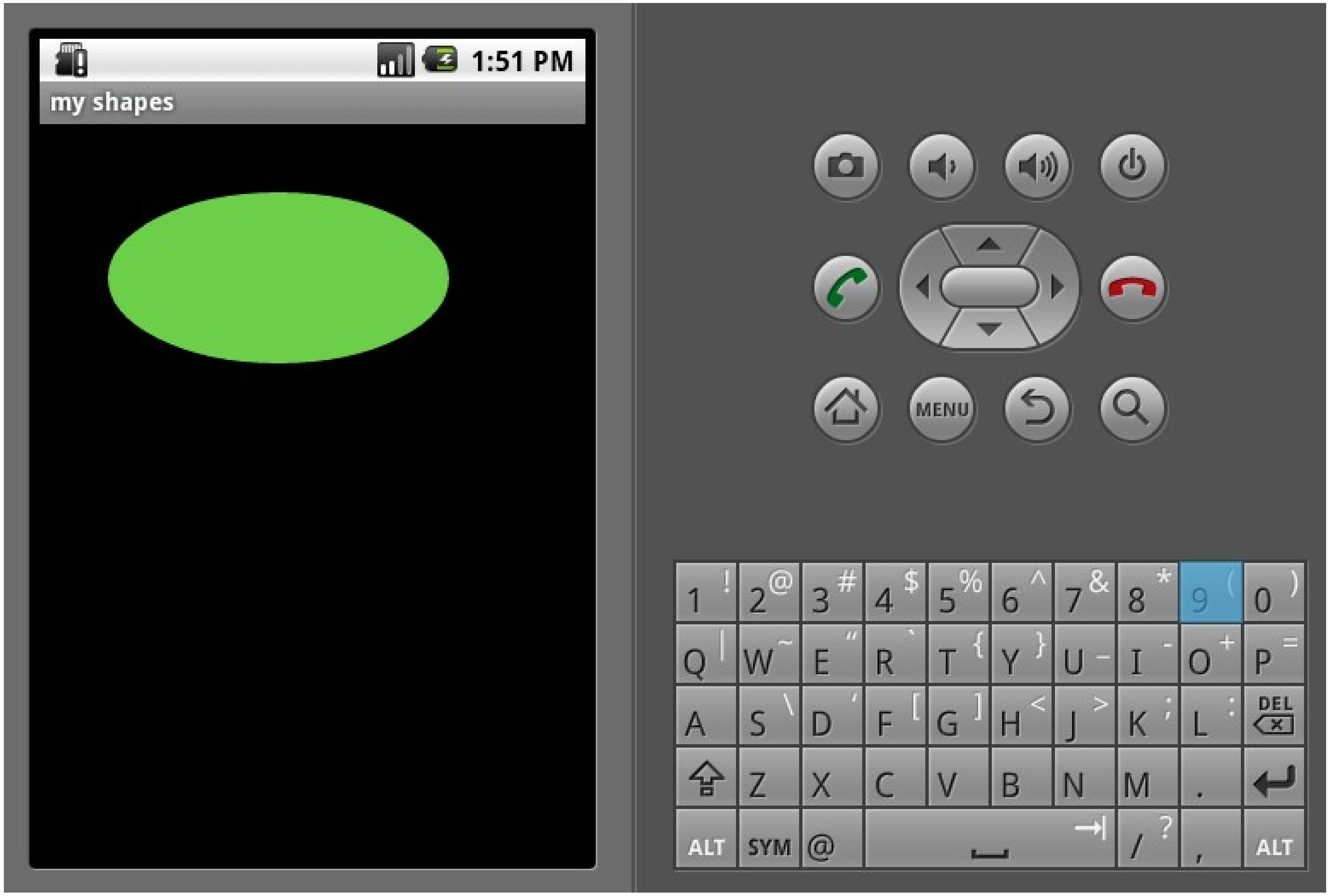
Sample

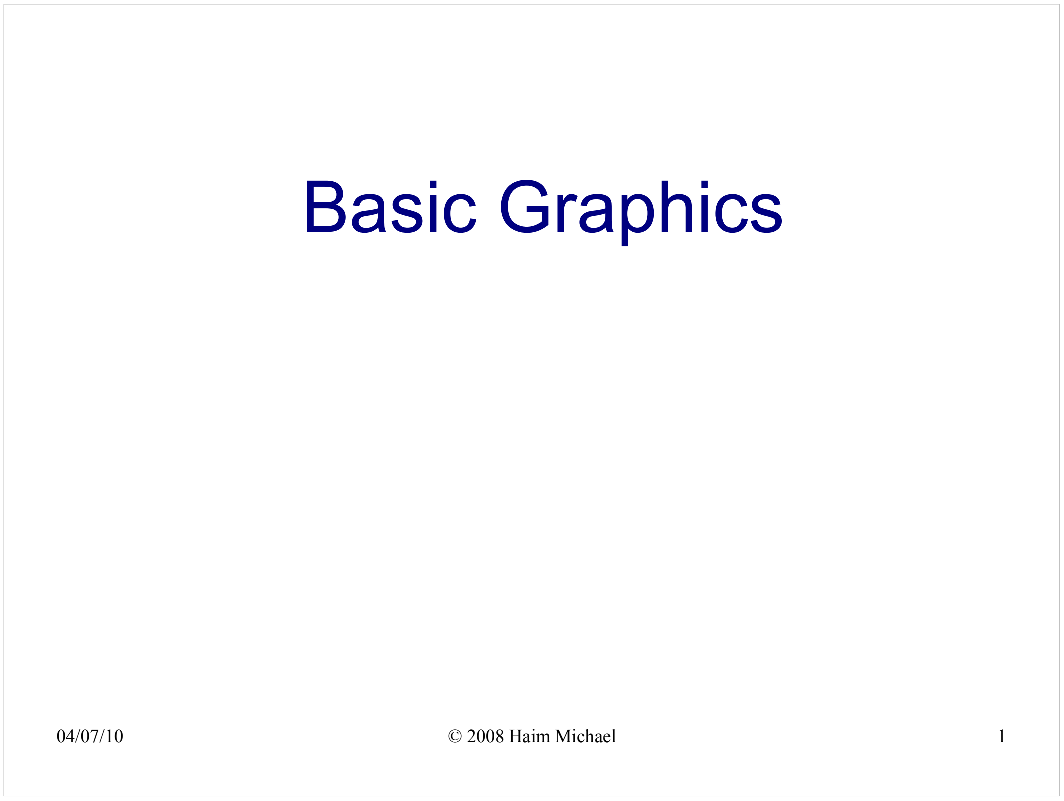
```
package com.abelski.samples;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.drawable.ShapeDrawable;
import android.graphics.drawable.shapes.OvalShape;
import android.view.View;

public class MyView extends View
{
    private ShapeDrawable mDrawable;

    public MyView(Context context)
    {
        super(context);
        int x = 40;
        int y = 40;
        int width = 200;
        int height = 100;
```

Sample

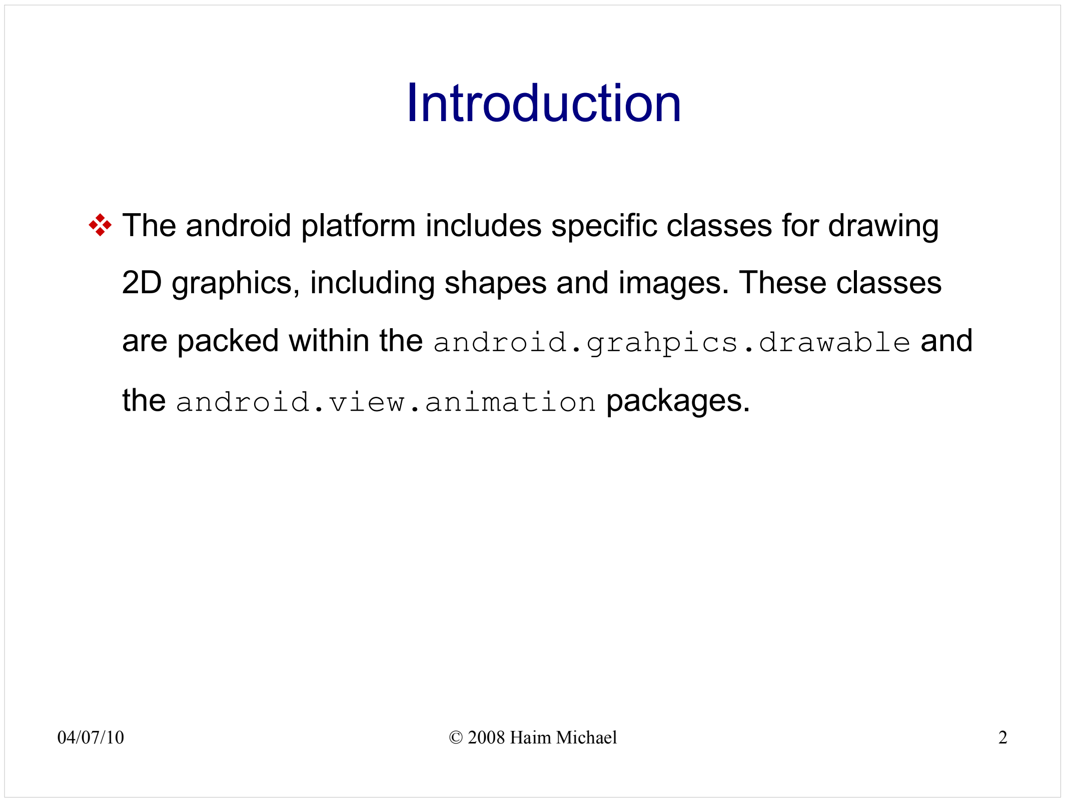
```
        mDrawable = new ShapeDrawable(new OvalShape());
        mDrawable.getPaint().setColor(0xf574DD54);
        mDrawable.setBounds(x, y, x + width, y + height);
    }

    protected void onDraw(Canvas canvas)
    {
        mDrawable.draw(canvas);
    }
}
```

Sample



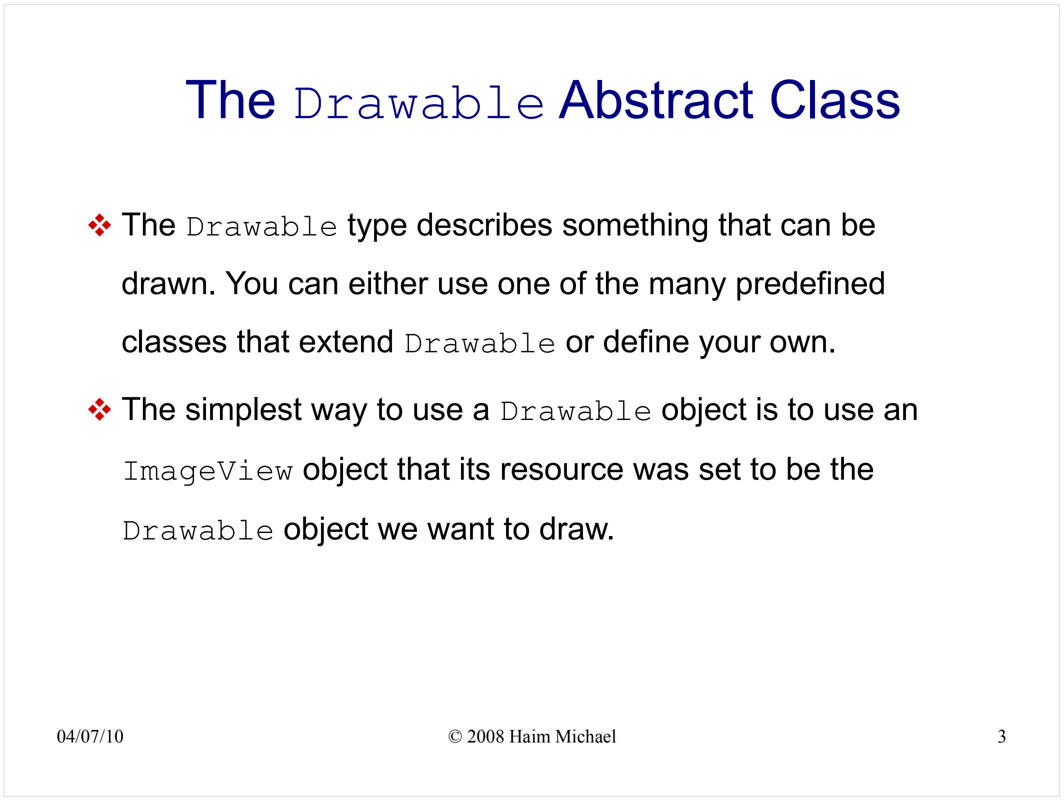


Basic Graphics



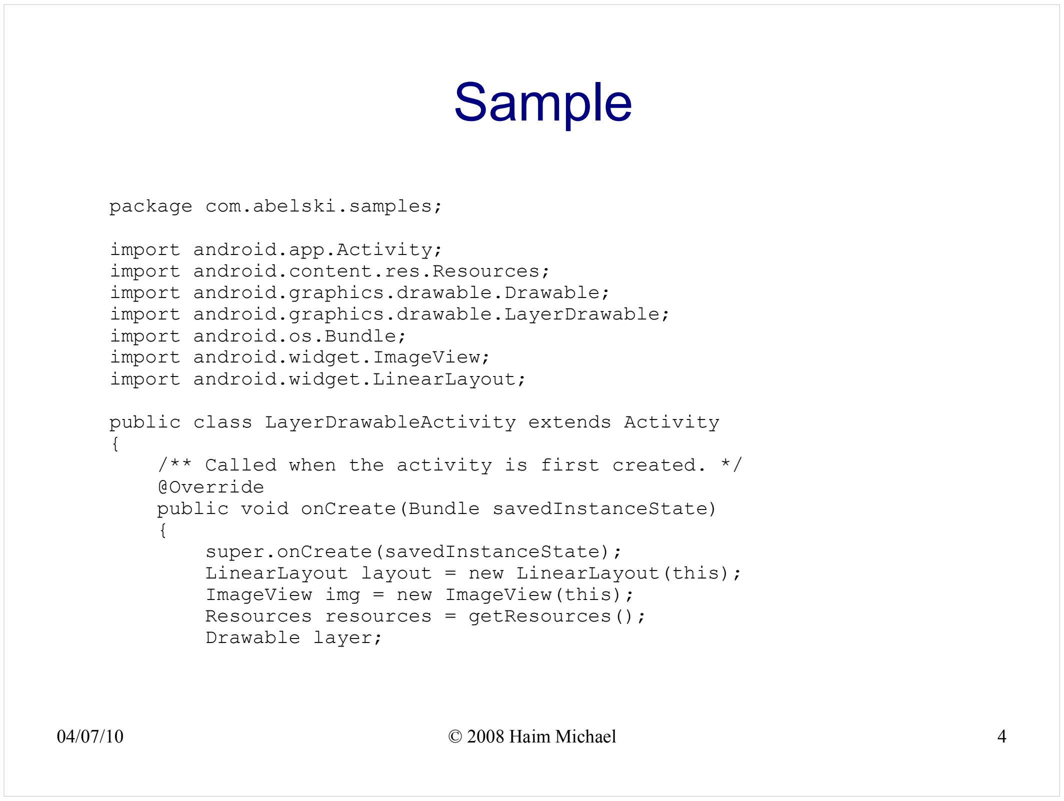
Introduction

- The android platform includes specific classes for drawing 2D graphics, including shapes and images. These classes are packed within the `android.grahpics.drawable` and the `android.view.animation` packages.



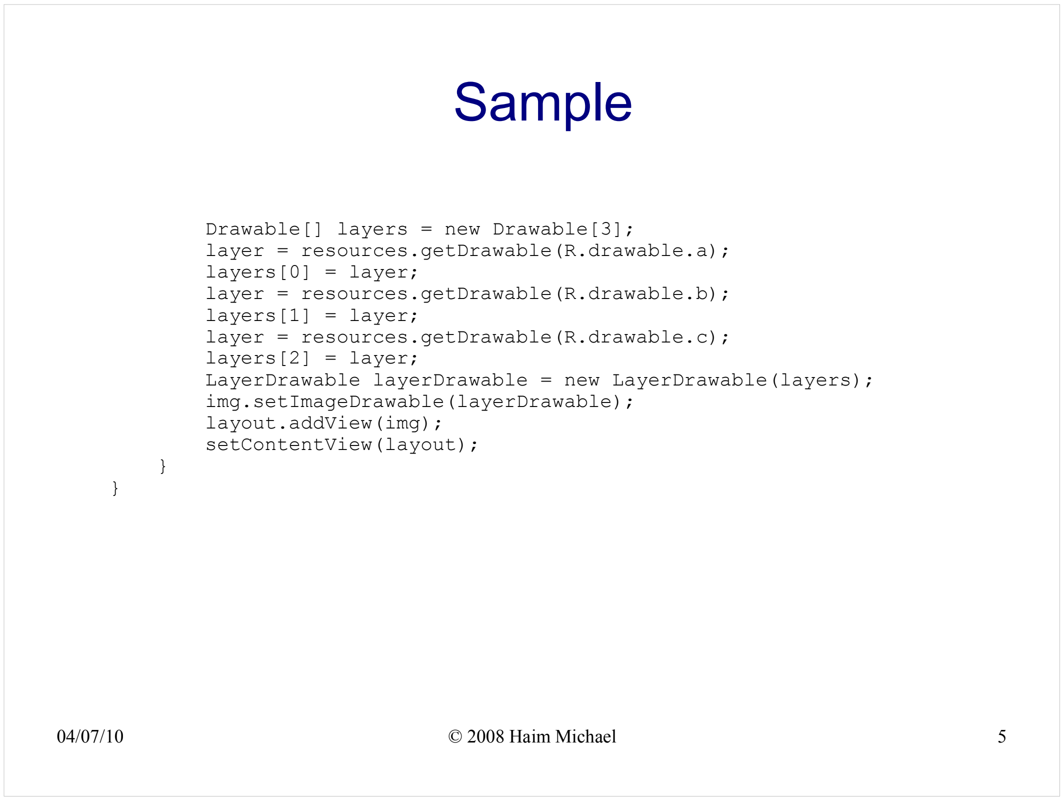
The `Drawable` Abstract Class

- The `Drawable` type describes something that can be drawn. You can either use one of the many predefined classes that extend `Drawable` or define your own.
- The simplest way to use a `Drawable` object is to use an `ImageView` object that its resource was set to be the `Drawable` object we want to draw.



Sample

```
package com.abelski.samples;

import android.app.Activity;
import android.content.res.Resources;
import android.graphics.drawable.Drawable;
import android.graphics.drawable.LayerDrawable;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.LinearLayout;

public class LayerDrawableActivity extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        LinearLayout layout = new LinearLayout(this);
        ImageView img = new ImageView(this);
        Resources resources = getResources();
        Drawable layer;
```



Sample

```
            Drawable[] layers = new Drawable[3];
            layer = resources.getDrawable(R.drawable.a);
            layers[0] = layer;
            layer = resources.getDrawable(R.drawable.b);
            layers[1] = layer;
            layer = resources.getDrawable(R.drawable.c);
            layers[2] = layer;
            LayerDrawable layerDrawable = new LayerDrawable(layers);
            img.setImageDrawable(layerDrawable);
            layout.addView(img);
            setContentView(layout);
        }
    }
```

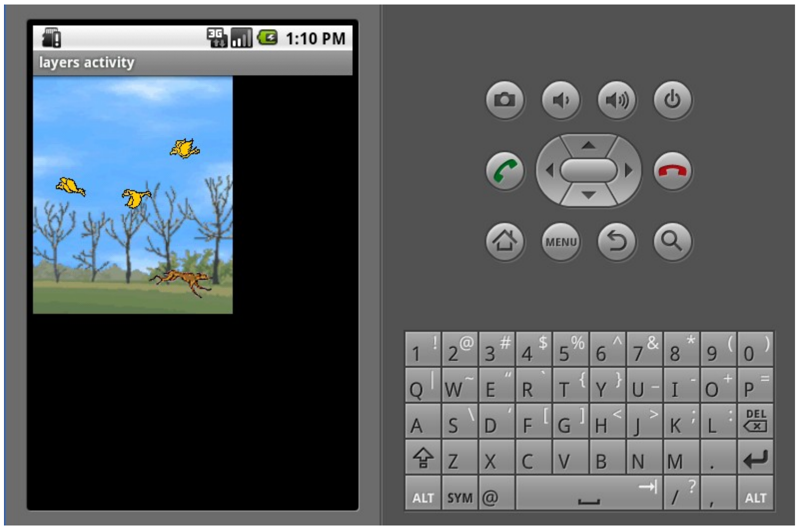
Sample

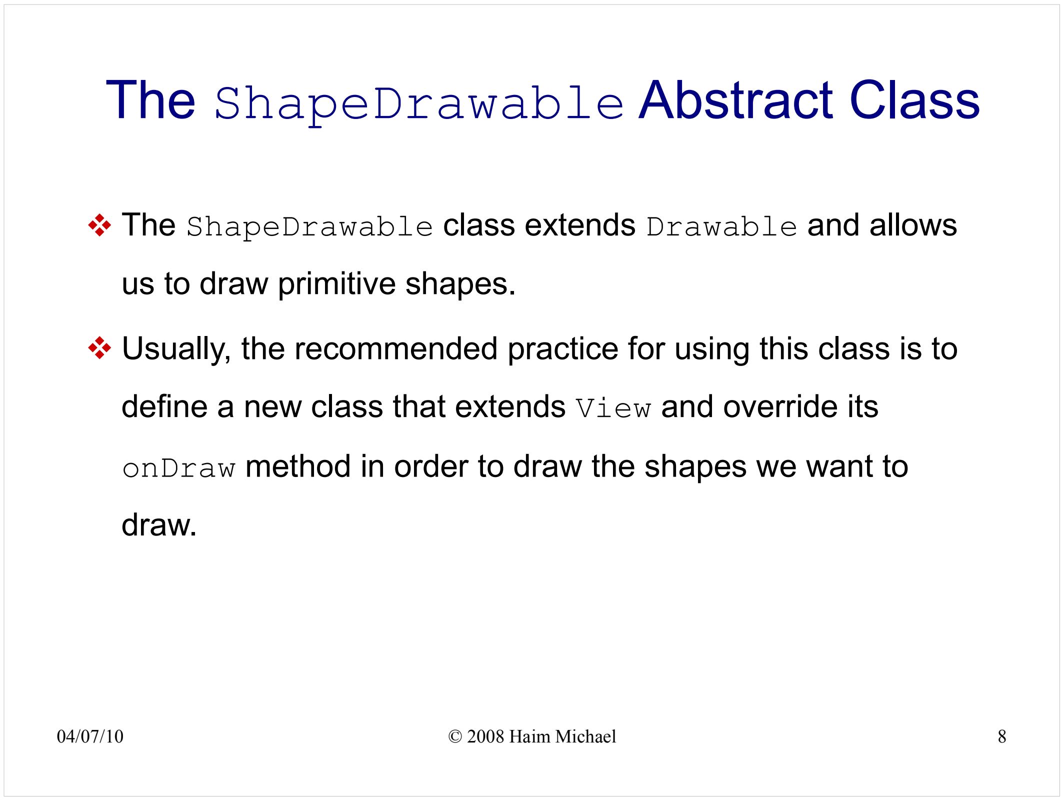
a.png

b.png

c.png

Sample

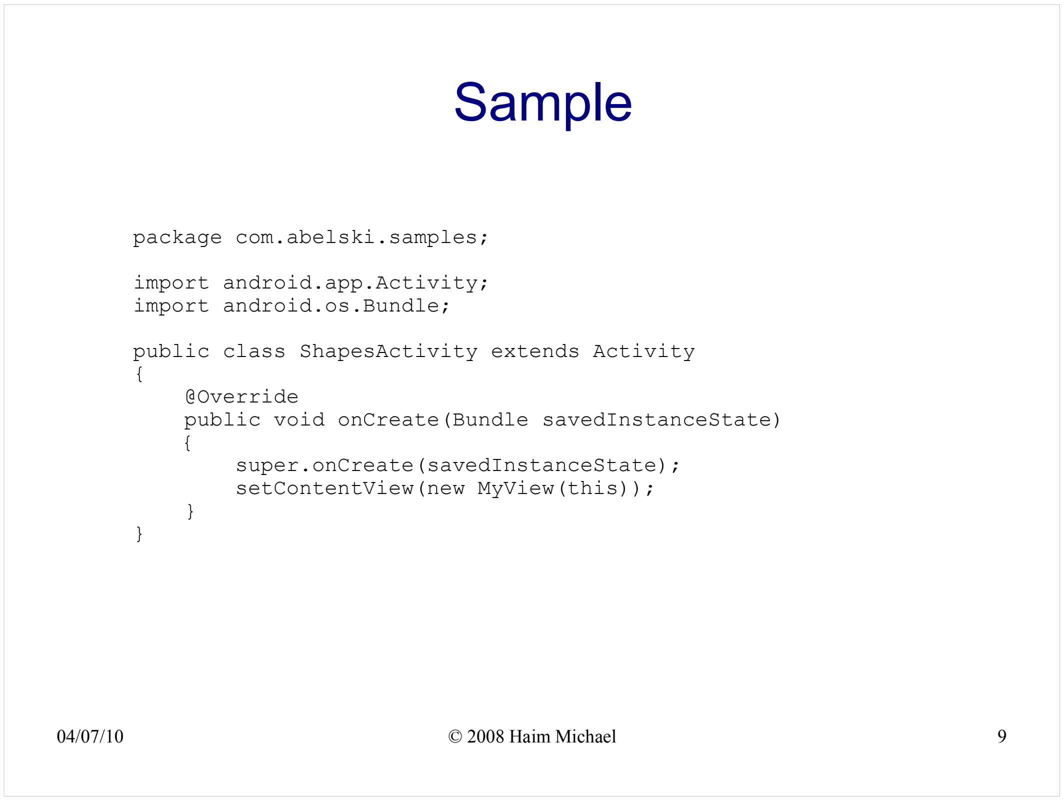




The `ShapeDrawable` Abstract Class

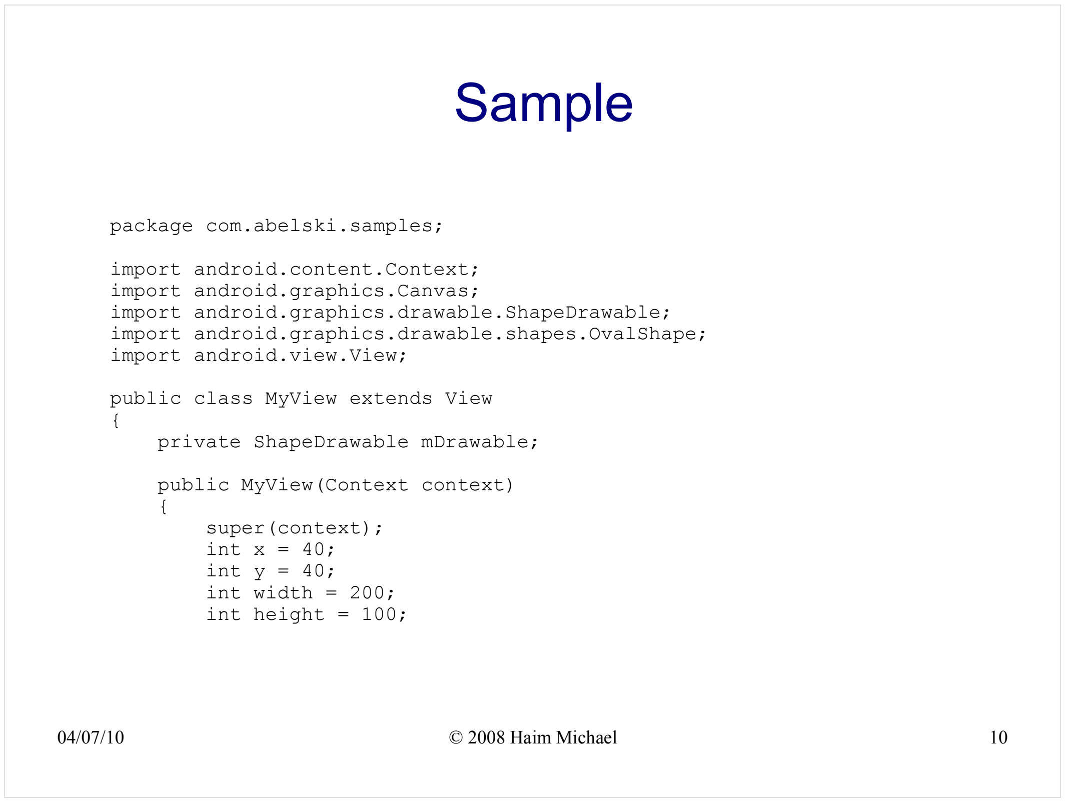
- The `ShapeDrawable` class extends `Drawable` and allows us to draw primitive shapes.
- Usually, the recommended practice for using this class is to define a new class that extends `View` and override its `onDraw` method in order to draw the shapes we want to draw.

04/07/10 © 2008 Haim Michael 8



Sample

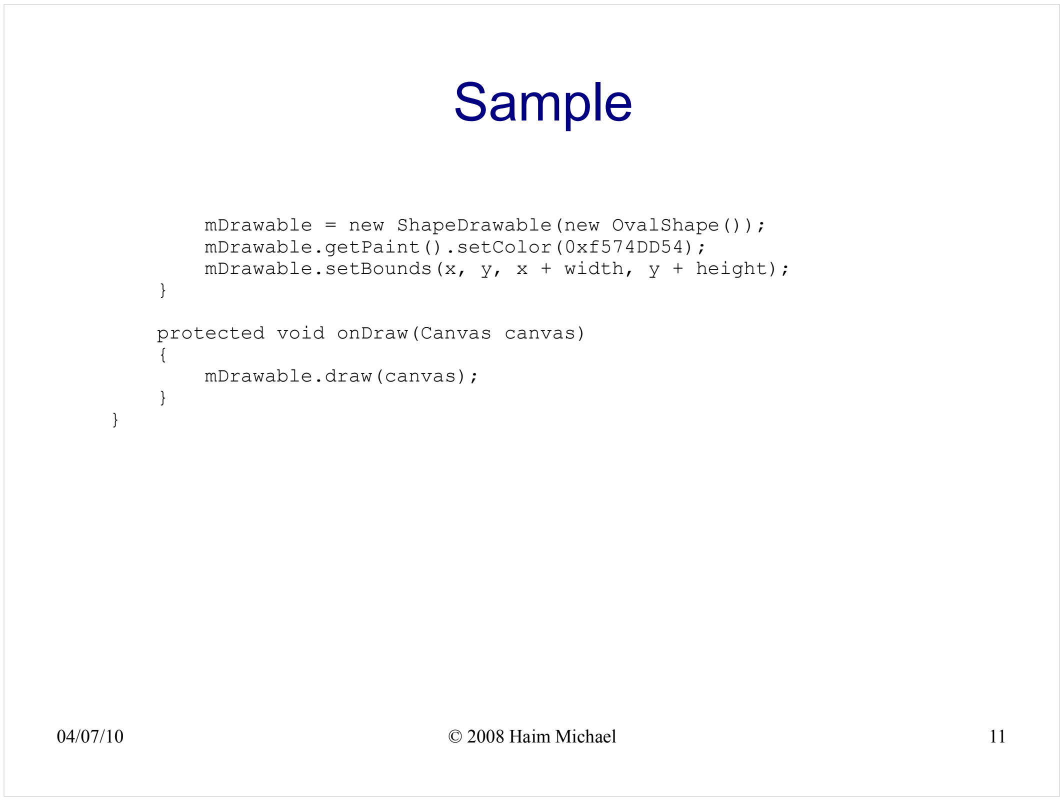
```
package com.abelski.samples;

import android.app.Activity;
import android.os.Bundle;

public class ShapesActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(new MyView(this));
    }
}
```



Sample

```
package com.abelski.samples;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.drawable.ShapeDrawable;
import android.graphics.drawable.shapes.OvalShape;
import android.view.View;

public class MyView extends View
{
    private ShapeDrawable mDrawable;

    public MyView(Context context)
    {
        super(context);
        int x = 40;
        int y = 40;
        int width = 200;
        int height = 100;
```

04/07/10 © 2008 Haim Michael 10

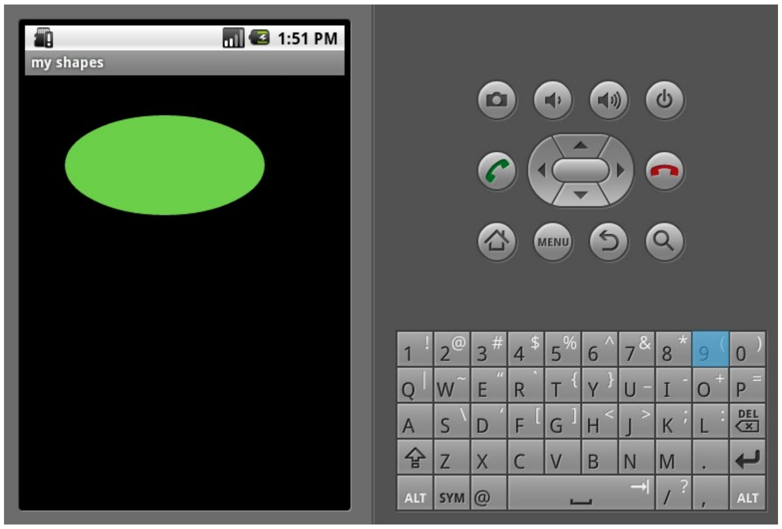


Sample

```
        mDrawable = new ShapeDrawable(new OvalShape());
        mDrawable.getPaint().setColor(0xf574DD54);
        mDrawable.setBounds(x, y, x + width, y + height);
    }

    protected void onDraw(Canvas canvas)
    {
        mDrawable.draw(canvas);
    }
}
```

04/07/10 © 2008 Haim Michael 11

Sample


1:51 PM
my shapes
MENU
DEL
ALT
SYM
ALT