CS 646 Android Mobile Application Development
Spring Semester, 2015
Doc 14 Touch, Gestures, Drawing
March 24, 2015

Chapter in Big Nerd Ranch

32

Touch Events

Touch

A view can generate touch events

Each touch event contains

- Type of event (down, up, move, etc)
- Number of touches
- Location each touch

Depending on device touch event may contain

- Pressure
- Size

Touch Events

To receive touch events from a view

Implement OnTouchListener interface

```
public boolean onTouch(View v, MotionEvent event)
```

Register as listener for that view

Example

```
public class TouchExampleActivity extends Activity implements OnTouchListener{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        View touchView = findViewById(R.id.touch);
        touchView.setOnTouchListener(this);
    }
      @Override
      public boolean onTouch(View v, MotionEvent event) {
            Log.i("rew", event.toString());
            logTouchType(event);
            Log.i("rew", "number of touches; " + event.getPointerCount());
            Log.i("rew", "x; " + event.getX() + " y: " + event.getY());
            for (int k = 1; k < event.getPointerCount();k++ )
                  Log.i("rew", "x; " + event.getX(k) + " y: " + event.getY(k));
            return true;
      }
}
```

TouchExampleActivity Continued

```
private void logTouchType(MotionEvent event) {
        switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
                Log.i("rew", "down");
                break;
        case MotionEvent.ACTION_MOVE:
                Log.i("rew", "move " + event.getHistorySize() );
                break;
        case MotionEvent.ACTION_UP:
                Log.i("rew", "UP");
                break;
        default:
                Log.i("rew","other action " + event.getAction());
        }
    }
}
```

Layout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
 >
    <View
        android:id="@+id/touch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```


Could have become listener to the linearLayout

Output on Samsung Galaxy IIS

MotionEvent{40514110 action=0 x=406.0 y=358.0 pressure=0.20392159 size=0.20000002}
down
number of touches; 1
x; 406.0 y: 358.0
MotionEvent{40514110 action=261 x=160.0 y=429.0 pressure=0.21960786 size=0.23333335}
other action 261
number of touches; 2
x; 406.0 y: 358.0
x; 160.0 y: 429.0
MotionEvent{40514110 action=6 x=406.0 y=358.0 pressure=0.20392159 size=0.20000002}
other action 6
number of touches; 2
x; 406.0 y: 358.0
x; 160.0 y: 429.0
MotionEvent{40514110 action=1 x=160.0 y=429.0 pressure=0.21960786 size=0.23333335}
UP
number of touches; 1
x; 160.0 y: 429.0

Nexus 5

MotionEvent { action=ACTION_DOWN, id[0]=0, x[0]=538.0, y[0]=708.0, toolType[0]=TOOL_TYPE_FINGER, buttonState=0, metaState=0, flags=0x0, edgeFlags=0x0, pointerCount=1, historySize=0, eventTime=2705539276, downTime=2705539276, deviceId=4, source=0x1002 }

down

number of touches; 1

x; 538.0 y: 708.0

MotionEvent { action=ACTION_MOVE, id[0]=0, x[0]=536.0, y[0]=708.0, toolType[0]=TOOL_TYPE_FINGER, buttonState=0, metaState=0, flags=0x0, edgeFlags=0x0, pointerCount=1, historySize=1, eventTime=2705539343, downTime=2705539276, deviceId=4, source=0x1002 }

move 1

number of touches; 1

x; 536.0 y: 708.0

MotionEvent { action=ACTION_MOVE, id[0]=0, x[0]=534.0, y[0]=708.0, toolType[0]=TOOL_TYPE_FINGER, buttonState=0, metaState=0, flags=0x0, edgeFlags=0x0, pointerCount=1, historySize=1, eventTime=2705539360, downTime=2705539276, deviceId=4, source=0x1002 }

move 1

number of touches; 1

x; 534.0 y: 708.0

Multiple Events

Don't get just one touch event

Get stream of events!

So how to tell what user is doing?

MotionEvent Actions

DOWN

First finger/touch device touches screen

MOVE

finger/touch device moevs on screen

POINTER_DOWN

Second, Third, etc finger/touch device touches screen

POINTER_UP

Second, Third, etc finger/touch device stops touching screen

UP

Last finger/touch device stops touching screen

CANCEL

Touch event cancelled

Parent View takes over touch event

OUTSIDE

MotionEvent Actions

Each motion event has an action

UP, DOWN, CANCEL, MOVE, OUTSIDE, POINTER_DOWN, POINTER_UP

Stream of motion events

starts with DOWN

ends with UP or CANCEL

Getting the Motion Action

```
public boolean onTouch(View v, MotionEvent event) {
        int action = event.getAction();
        int actionCode = action & MotionEvent.ACTION_MASK;
```

Why the ACTION_MASK?

POINTER_DOWN events

event.getAction();

Returns different value depending on how many fingers are on screen

event.getAction() & MotionEvent.ACTION_MASK;

Returns action code disregarding number of fingers on the screen

Example - Single Finger Swipe

Track single finger swipe

Print out change in direction at end of swipe

Monday, March 23, 15

Android does have gesture recognizers which are easier to use than detecting an swipe from raw events.

The Activity

```
public class TouchExampleActivity extends Activity implements OnTouchListener{
    private float startX;
      private float startY;
      private boolean swipeInProgress = false;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        View touchView = findViewById(R.id.touch);
        touchView.setOnTouchListener(this);
    }
```

Activity Continued - onTouch

```
public boolean onTouch(View v, MotionEvent event) {
    int action = event.getAction();
    int actionCode = action & MotionEvent.ACTION_MASK;
    switch (actionCode) {
        case MotionEvent.ACTION_DOWN:
            return handleActionDown(event);
        case MotionEvent.ACTION_UP:
            return handleActionUp(event);
        case MotionEvent.ACTION_CANCEL:
        case MotionEvent.ACTION_POINTER_DOWN:
        case MotionEvent.ACTION_POINTER_UP:
            swipeInProgress = false;
            return false;
    }
    return false;
}
```

Handling the Events

```
private boolean handleActionDown(MotionEvent event) {
        swipeInProgress = true;
        startX = event.getX();
        startY = event.getY();
        return true;
}

private boolean handleActionUp(MotionEvent event) {
        if (!swipeInProgress) return false;
        float endX = event.getX();
        float endY = event.getY();
        Log.i("rew", "x swipe distance " + (endX - startX));
swipeInProgress = false;
        return true;
 }
```

Fine Point - Batching

Android may combine multiple move events into one event

Event will then have history size greater than 0

Can get events in history

Fine Point - Multiple Touches

Multiple fingers touching across multiple events

Each event has list of touch points

K'th touch point in different events may represent different finger

Finger A touches screen
Then Finger B touches screen
Finger A - index = 0
Finger B - index = 1
Finger A lifted off screen
Finger B - index = 0

But each finger is given an id which does not change

So use ids rather than location to track individual fingers

Tracking Movement

VelocityTracker

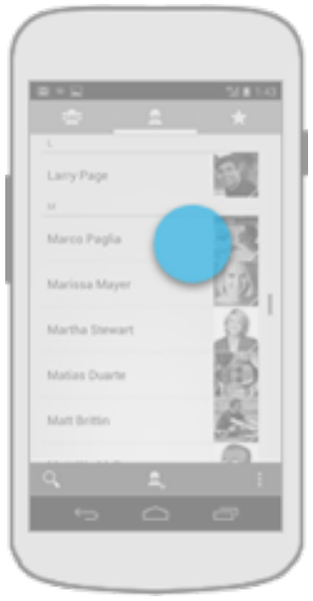
Tracks velocity of touch events

VelocityTracker.obtain()	to get tracker
addMovement(MotionEvent)	to track event
computeCurrentVelocity(int)	to compute velocity
getXVelocity()	
getYVelocity()	returns last computed velocity

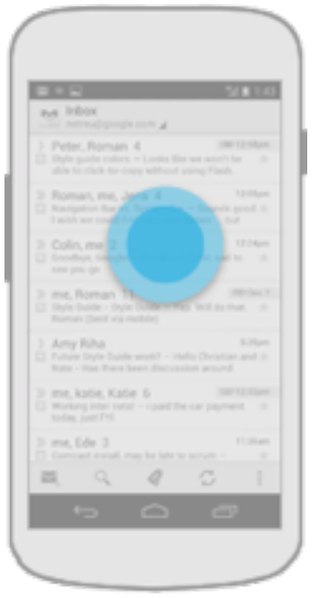
```
VelocityTracker velocity;

public boolean onTouch(View v, MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            velocity = VelocityTracker.obtain();
            velocity.addMovement(event);
            break;
        case MotionEvent.ACTION_MOVE:
            velocity.addMovement(event);
            lastEvent = event;
            break;
        case MotionEvent.ACTION_UP:
            velocity.computeCurrentVelocity(1000);
            Log.i("rew", "X vel " + velocity.getXVelocity() + " Y vel " + velocity.getYVelocity());
            velocity.recycle();
            velocity = null;
            break;
       default:
            Log.i("rew","other action " + event.getAction());
    }
    return true;
}
```

Gestures

Standard Android Gestures



Touch/
Tap



Long
Touch

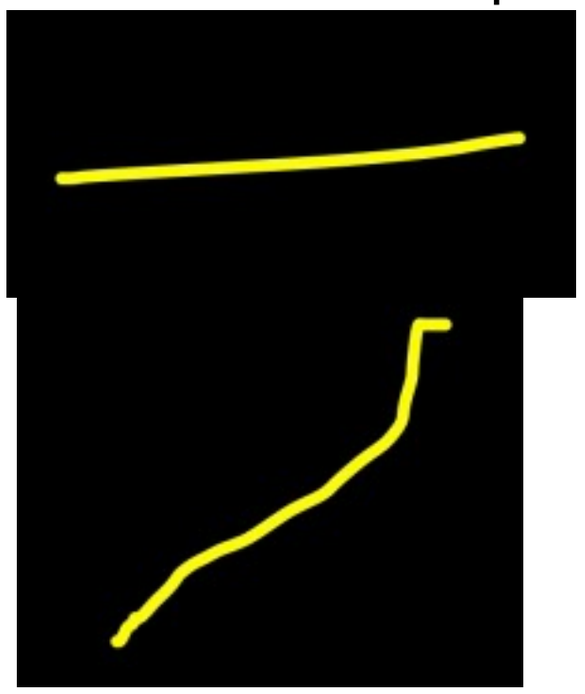
Zoom/
Pinch Open

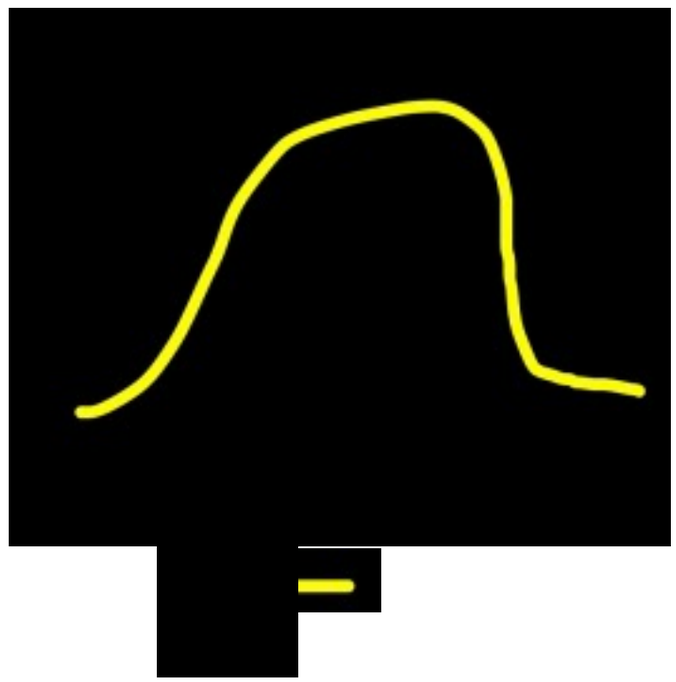
Pinch
Close

Double
Touch/tap

Swipe

Which is a swipe left(right)





Android Gesture Systems

Recognize standard gestures

Programmer & user can create new gestures

Recognize Standard Gestures

Get touch events

Pass touch events to Gesture detector

Gesture detector when detects gesture
    calls OnGestureListener methods on gesture listener

Basic Classes and Interfaces

GestureDetector

- Detects standard one finger gestures
- Does not detect pinch

OnGestureListener (Interface)

- onDown
- onFling
- onLongPress
- onScroll
- onShowPress
- onSingleTap

OnDoubleTapListener (Interface)

- double tap method

SimpleOnGestureListener

- Implement OnDoubleTapListener & OnGestureListener
- Subclass this class
- Implement just the methods you are interested in

ScaleGestureDetector

- Detects pinch gestures

OnScaleGestureListener (Interface)

SimpleOnScaleGestureListener

Basic Classes and Interfaces

ScaleGestureDetector

- Detects pinch gestures

OnScaleGestureListener (Interface)

- onScale
- onScaleBegin
- onScaleEnd

SimpleOnScaleGestureListener

- Subclass this class
- Implement just the methods you are interested in

Example

```
public class MainActivity extends Activity implements OnTouchListener {

    private GestureDetector mGestureDetector;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        View entireScreen = findViewById(R.id.sampleview);
        entireScreen.setOnTouchListener(this);
        mGestureDetector = new GestureDetector(this, new GestureListener());
    }
```

Example - OnTouch

```
@Override
public boolean onTouch(View v, MotionEvent event) {
        boolean didUseEvent =  mGestureDetector.onTouchEvent(event);
        Log.i("rew", "gesture did consume " + didUseEvent);
        return true;
}
```

Need to return turn on events that are gestures

GestureListener

```
private class GestureListener extends GestureDetector.SimpleOnGestureListener
{

        public boolean onDoubleTap(MotionEvent e) {
                Log.i("rew", "double tab");
                return true;
        }

        public boolean onSingleTapConfirmed(MotionEvent e) {
                Log.i("rew", "single tab");
                return true;
        }

        public void onLongPress (MotionEvent e) {
                Log.i("rew", "long press");
        }
```

These methods are called once per gesture

GestureListener - Fling

```
public boolean onFling(MotionEvent e1, MotionEvent e2,
    float velocityX, float velocityY) {
            Log.i("rew", "fling");
            return true;
        }
```

Called once per swipe/scroll

onScroll will also be called

GestureListener - onScroll

```
    public boolean onScroll(MotionEvent startEvent, MotionEvent endEvent,
      float distanceX, float distanceY) {
    float deltaX = Math.abs(startEvent.getX() - endEvent.getX());
          float deltaY = startEvent.getY() - endEvent.getY();
          if ((deltaX < 25) && (deltaY > 100)) {
                    Log.i("rew", "swipe up");
return true;
   }
```

This method is called multiple times per swipe/scroll

Called on all swipes/scrolls

You have to decide when it left/right/up/down

distanceX(Y) is distance traveled since last method call

startEvent is the first event in the swipe

25 & 100

```
if ((deltaX < 25) && (deltaY > 100)) {
        Log.i("rew", "swipe up");
```

Determined values experimentally

May not be the best

Common Case

Interested in one or two gestures

Just implement the methods you are interested in

SimpleOnGestureListener

- Implements all methods
- But they do nothing
- Subclass this class
- Override the methods you are interested in

The Return value

```
public boolean onSingleTapConfirmed(MotionEvent e) {
        return true;
}
```

Return true if you "consume the event"

2D Drawing

Basic Parts

Bitmap

Rectangular grid of pixels of an image

What is displayed on screen

PNG, JPG are bitmap formats

Canvas

Knows how to draw on bitmaps

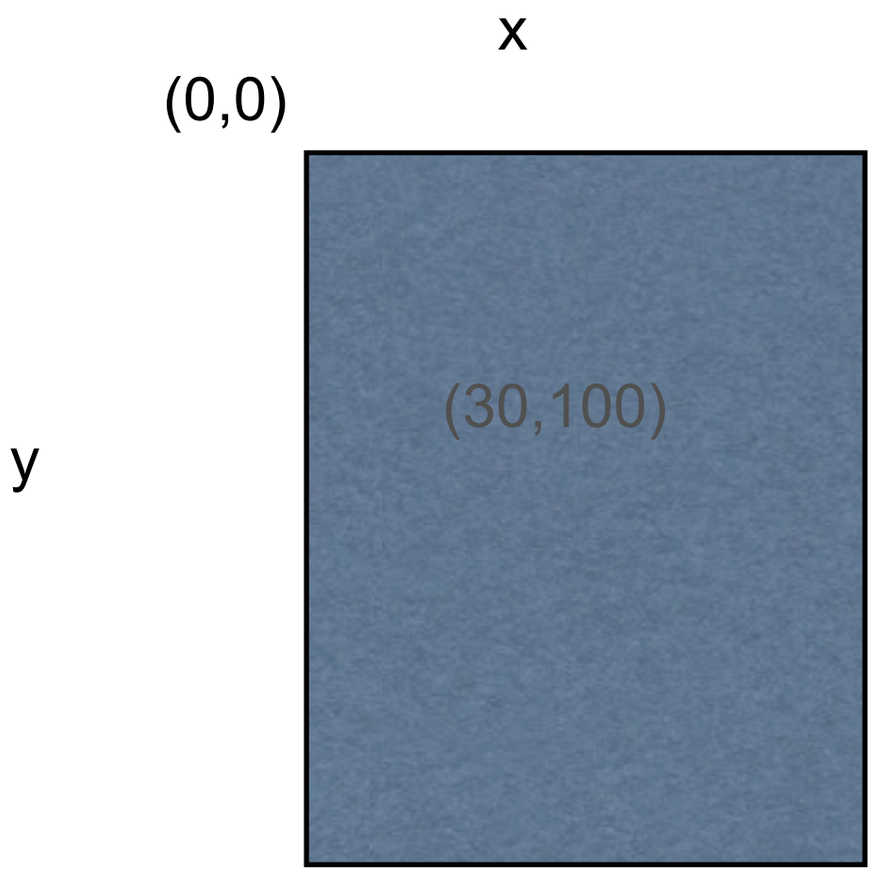
Paint

Style and color information about how to draw things

Drawing primitive

Rect, text , Path, Bitmap

Coordinates



Colors

(alpha << 24) | (red << 16) | (green << 8) | blue

Color value range 0..255

alpha red green blue

0xff74AC23

alpha = 0xff (255) red = 0x74 (116) green = 0xAC (172) blue = 0x23 (35)

2D Drawing

Create subclass of View

Implement onDraw(Canvas canvas)

Draw on the provided canvas

drawLine

drawRect

drawCircle

drawText

drawBitmap

etc

Paint

Style and color information about how to draw things

How thick lines should be

Are corners rounded or not

Font information

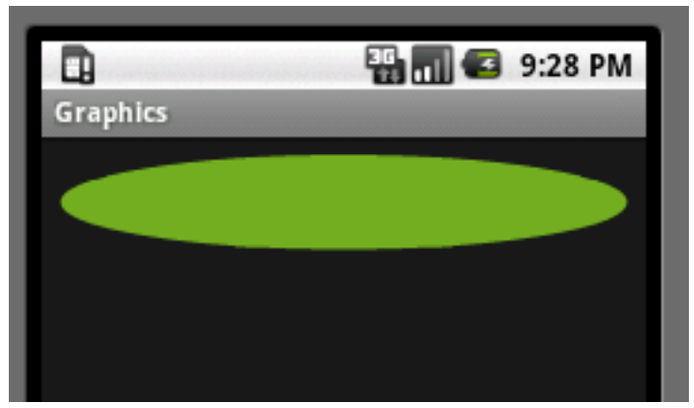
Shapes

Android has few predefined shapes

- PathShape
- RectShape
- ArcShape
- OvalShape
- RoundRectShape

Drawing an Oval with Shapes

```
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);
    }
```

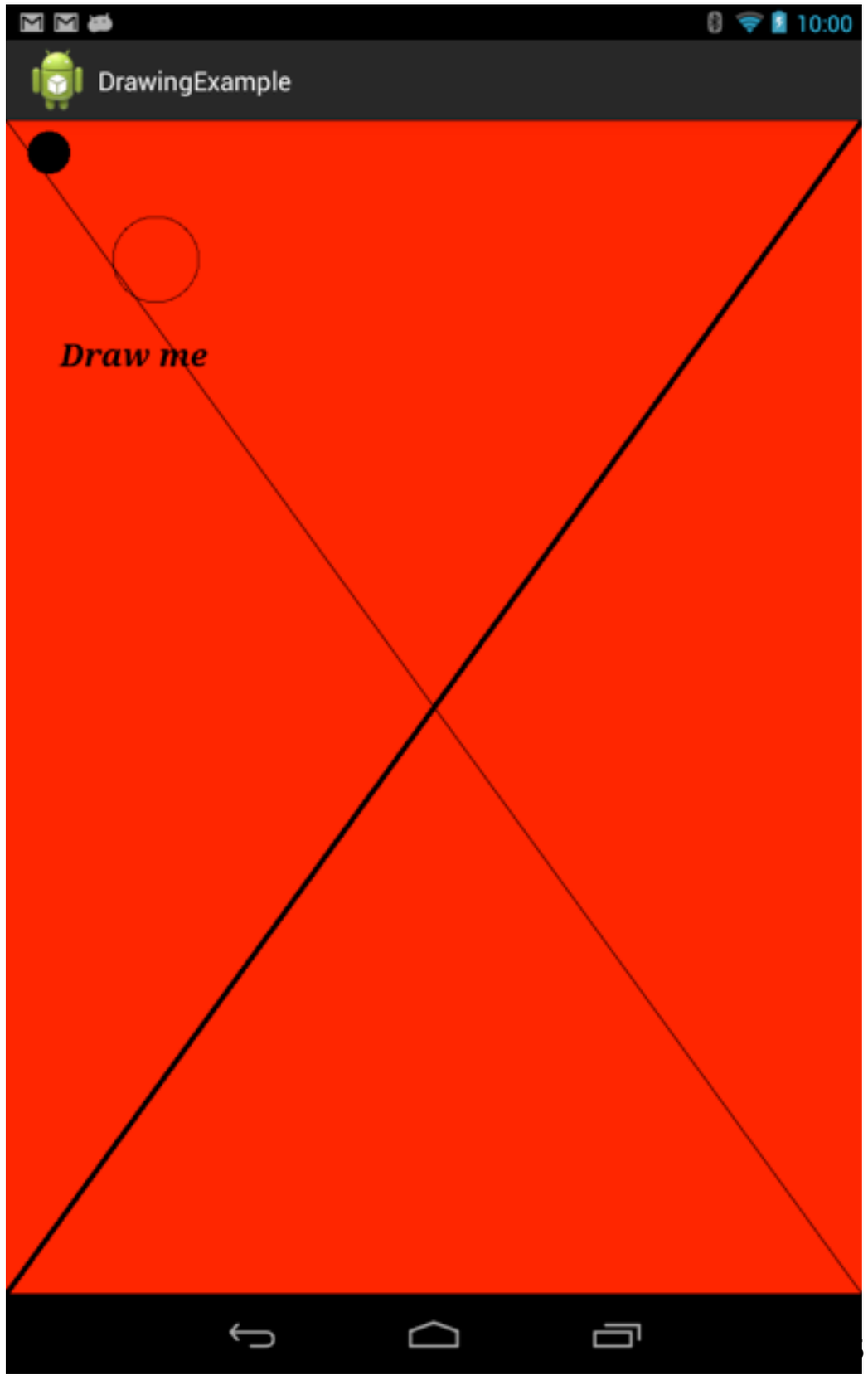
SimpleDrawing

```
public SimpleDrawing(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        this.oval.draw(canvas);
    }
}
```

onDraw

Each time the view needs to be drawn on the screen its onDraw method is called

Drawing on the Canvas



Activity

```
public class GraphicsExamples extends Activity {

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

SimpleDrawing

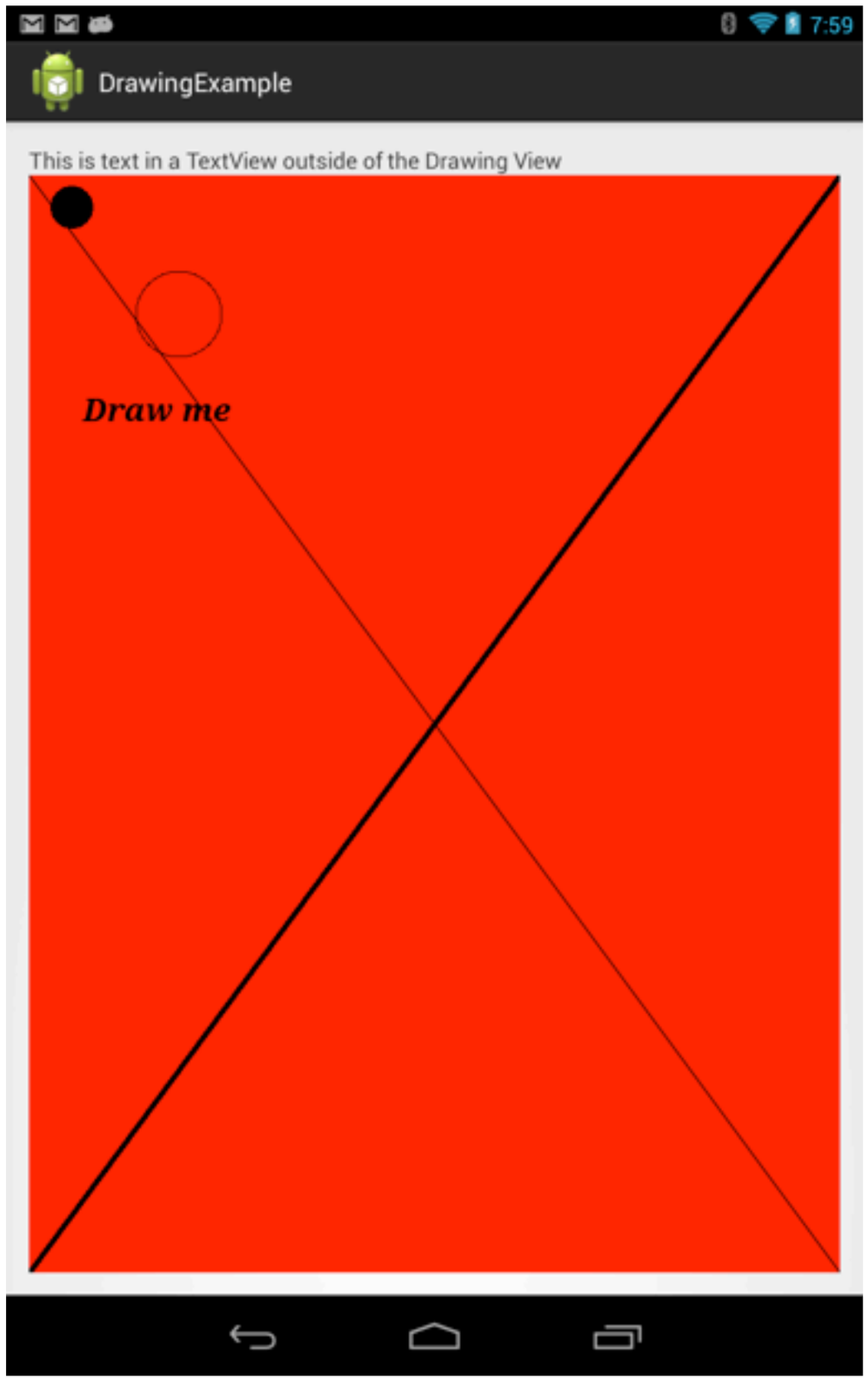
```
public class SimpleDrawing extends View {
    static Paint blackFill;
    static Paint blackFramed;
    static Paint blackThick;

    static {
        blackFill = new Paint();
        blackFill.setColor(Color.BLACK);

        blackThick = new Paint();
        blackThick.setColor(Color.BLACK);
        blackThick.setStrokeWidth(5.0f);

        blackFramed = new Paint();
        blackFramed.setColor(Color.BLACK);
        blackFramed.setStyle(Paint.Style.STROKE);
        blackFramed.setTextSize(30);
        blackFramed.setTypeface(Typeface.create("serif", Typeface.BOLD_ITALIC));
    }
```

The Drawing

```
public SimpleDrawing(Context context) {
        super(context);
}

protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.RED);
        canvas.drawCircle(40, 30, 20, blackFill);
        canvas.drawCircle(140, 130, 40, blackFramed);
        canvas.drawText("Draw me", 50, 230, blackFramed);
        int height = canvas.getHeight();
        int width = canvas.getWidth();
        canvas.drawLine(0, height, width, 0, blackThick);
        canvas.drawLine(0, 0, width, height, blackFramed);
    }
}
```


The constructor is required as parent class does not have default constructor

Drawing using Layout



SimpleDrawing

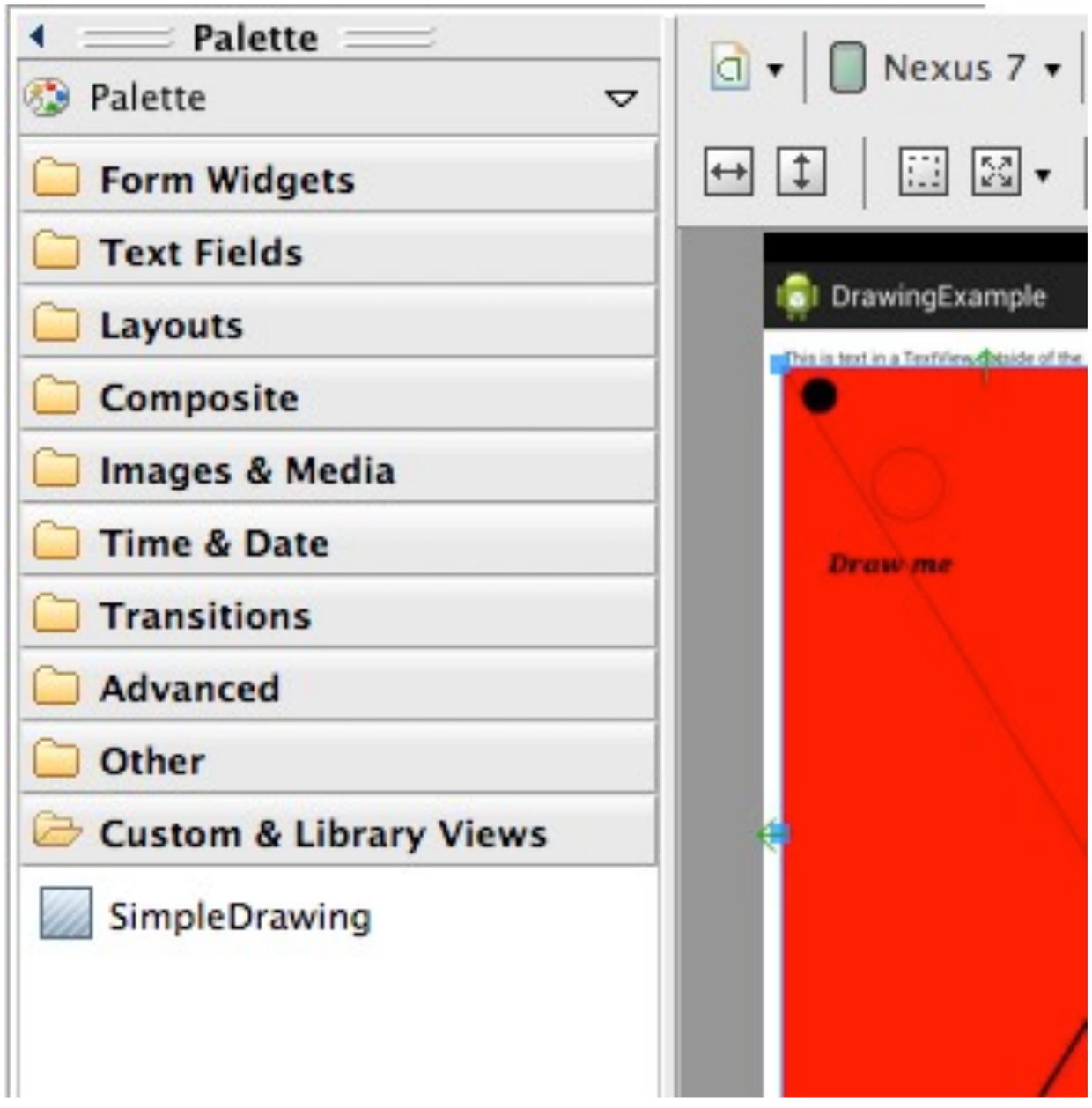
```
public class SimpleDrawing extends View {
        static Paint blackFill;
        static Paint blackFramed;
        static Paint blackThick;

        static {
                blackFill = new Paint();
                blackFill.setColor(Color.BLACK);
                blackThick = new Paint();
                blackThick.setColor(Color.BLACK);
                blackThick.setStrokeWidth(5.0f);
                blackFramed = new Paint();
                blackFramed.setColor(Color.BLACK);
                blackFramed.setStyle(Paint.Style.STROKE);
                blackFramed.setTextSize(30);
                blackFramed.setTypeface(Typeface.create("serif", Typeface.BOLD_ITALIC));
        }
```

SimpleDrawing

```
public SimpleDrawing(Context context, AttributeSet xmlAttributes) {
    super(context, xmlAttributes);
}

protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.RED);
        canvas.drawCircle(40, 30, 20, blackFill);
        canvas.drawCircle(140, 130, 40, blackFramed);
        canvas.drawText("Draw me", 50, 230, blackFramed);
        int height = canvas.getHeight();
        int width = canvas.getWidth();
        canvas.drawLine(0, height, width, 0, blackThick);
        canvas.drawLine(0, 0, width, height, blackFramed);
    }
}
```


The constructor is called when inflating the xml from the layout

SimpleDrawing View in GUI builder



Part of the Layout

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world" />

<edu.sdsu.cs.whitney.drawingexample.SimpleDrawing
    android:id="@+id/simpleDrawing1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/textView1" />
```

Drawing Multiple Lines

public void drawLines (float[] pts, Paint paint)

pts

Array of points to draw [x0 y0 x1 y1 x2 y2 ...]

Each line requires 4 points

x & y of start point

X & y of end point

How to handle multiple lines in Program

Idea 1. Keep an array of points

```
ArrayList<Float> lines = new ArrayList<Float>();
```

Adding new point

```
lines.add(startx);
lines.add(starty);
lines.add(endx);
lines.add(endx);
```

Drawing

```
protected void onDraw(Canvas canvas) {
    float[] linesToDraw =  new float[lines.size()];
    int index = 0;
    for (Float each : lines)
        linesToDraw[index++] = each;
    canvas.drawLines(linesToDraw, somePaint);
```

How to handle multiple lines in Program

Idea 2. Create Line Class

```
class Line {
    float startX;
    float startY;
    float endX;
    float endY;
    Paint lineColor;

    public Line(float x0, float y0, float x1, float y2) {
        startX = x0;
        etc.
    }

    public drawOn(Canvas canvas) {
        canvas.drawLine(startX, startY, endX, endY, lineColor);
```


Yes using points would simplify this

How to handle multiple lines in Program

Using Line Class in View class

```
ArrayList<Line> lines = new ArrayList<Line>();
```

Drawing

```
protected void onDraw(Canvas canvas) {
    for (Line each : lines)
        each.drawOn(canvas);
```

Rotation

Text and rectangles are always drawn perpendicular to the axis

Can not draw text at an angle

But there is a trick - Rotate the canvas then draw

Canvas Matrix

Matrix transforms the coordinate system used by canvas

Can translate, scale, rotate, skew and clip

canvas.rotate(DegreesToRotate,XPivot,YPivot)

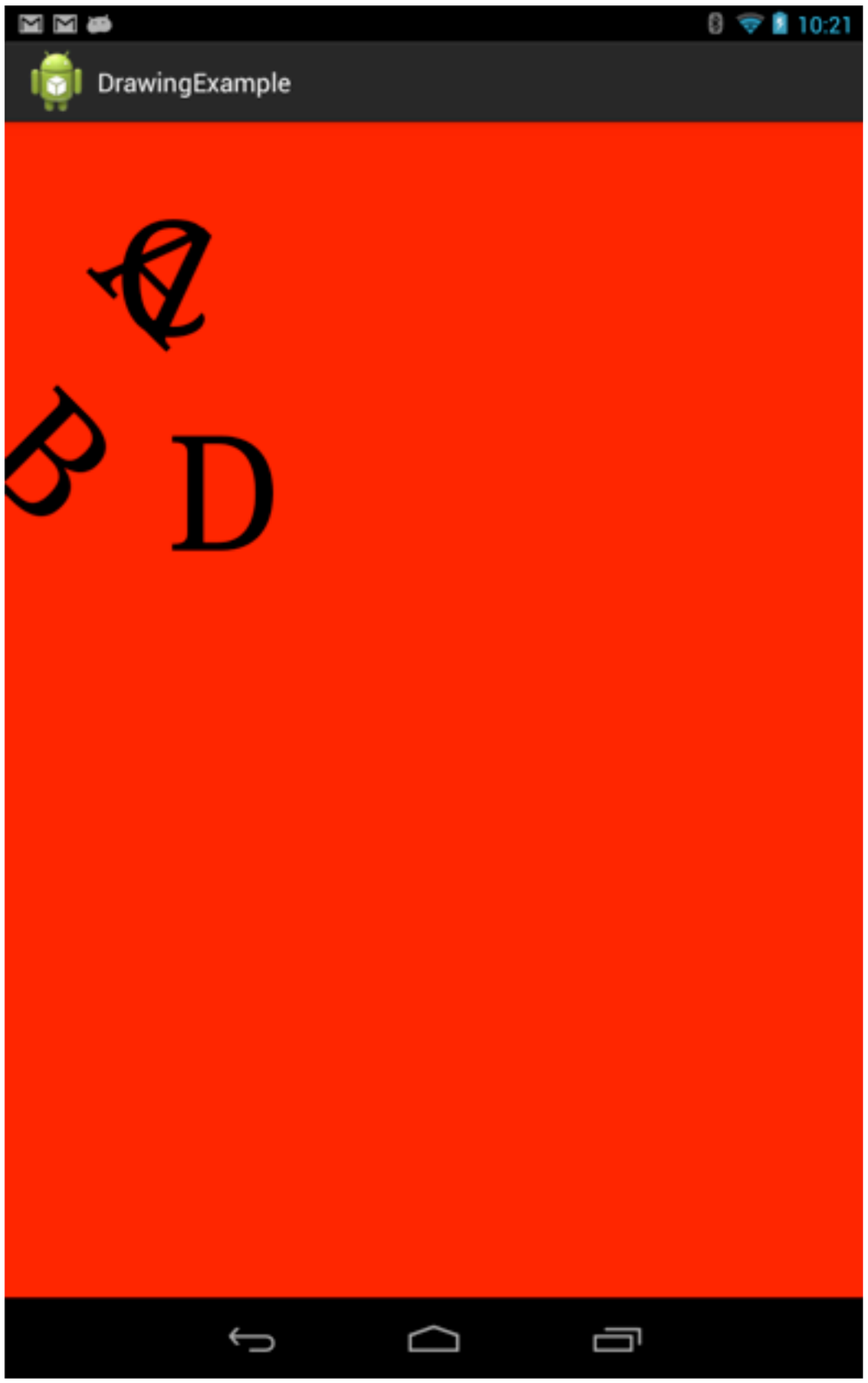
canvas.save()

Saves the current matrix

canvas.restore()

Replaces the current matrix with the last saved matrix

Rotation Example




```
protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.RED);
        int x = 100;
        int y = 200;
        float rotation = 45;
        canvas.save();
        canvas.rotate(rotation, x + 60, y - 60);
        canvas.drawText("A", x, y, black);
        canvas.drawText("B", x + 50, y + 200, black
        canvas.restore();
        canvas.drawText("C", x, y, black);
        canvas.drawText("D", x + 50, y + 200, black
    }
```

Some Performance Tricks

If drawing is complex and static

Draw on a bitmap object and just display bitmap

Draw on Picture object and display picture

Use thread to draw on a SurfaceView

Some Animation

Basic Animation

Draw on view

Change some aspect of drawing

Draw on view again

How to Trigger View to be Redrawn

Call invalidate() on the view

Have separate thread call invalidate()

Have some event trigger the call

Call invalidate at end of onDraw()

Rotating Graphics

Canvas rotate method

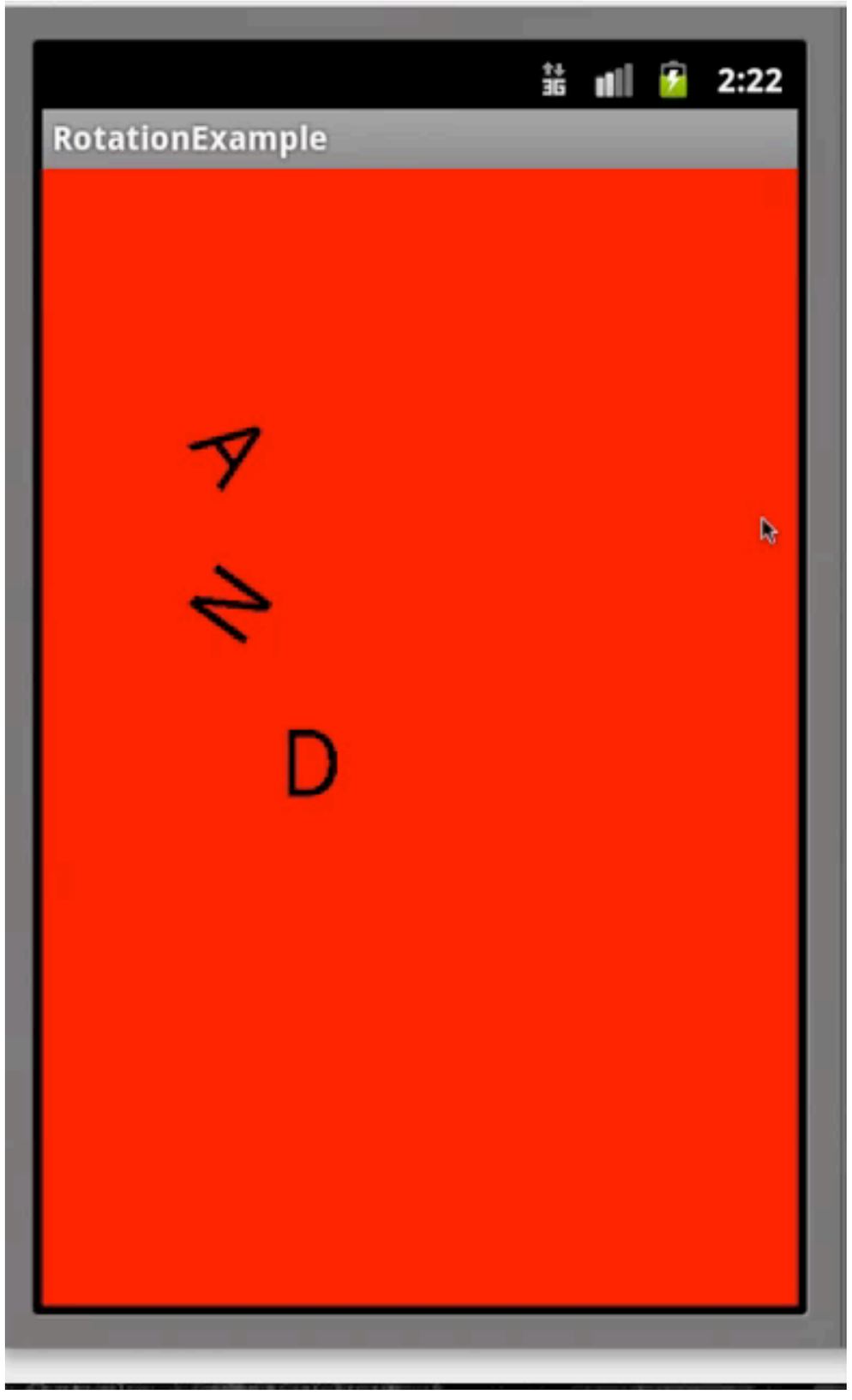
Applies rotation to current transformation matrix

save

Saves current matrix on stack

restore

Replaces current matrix with top of stack

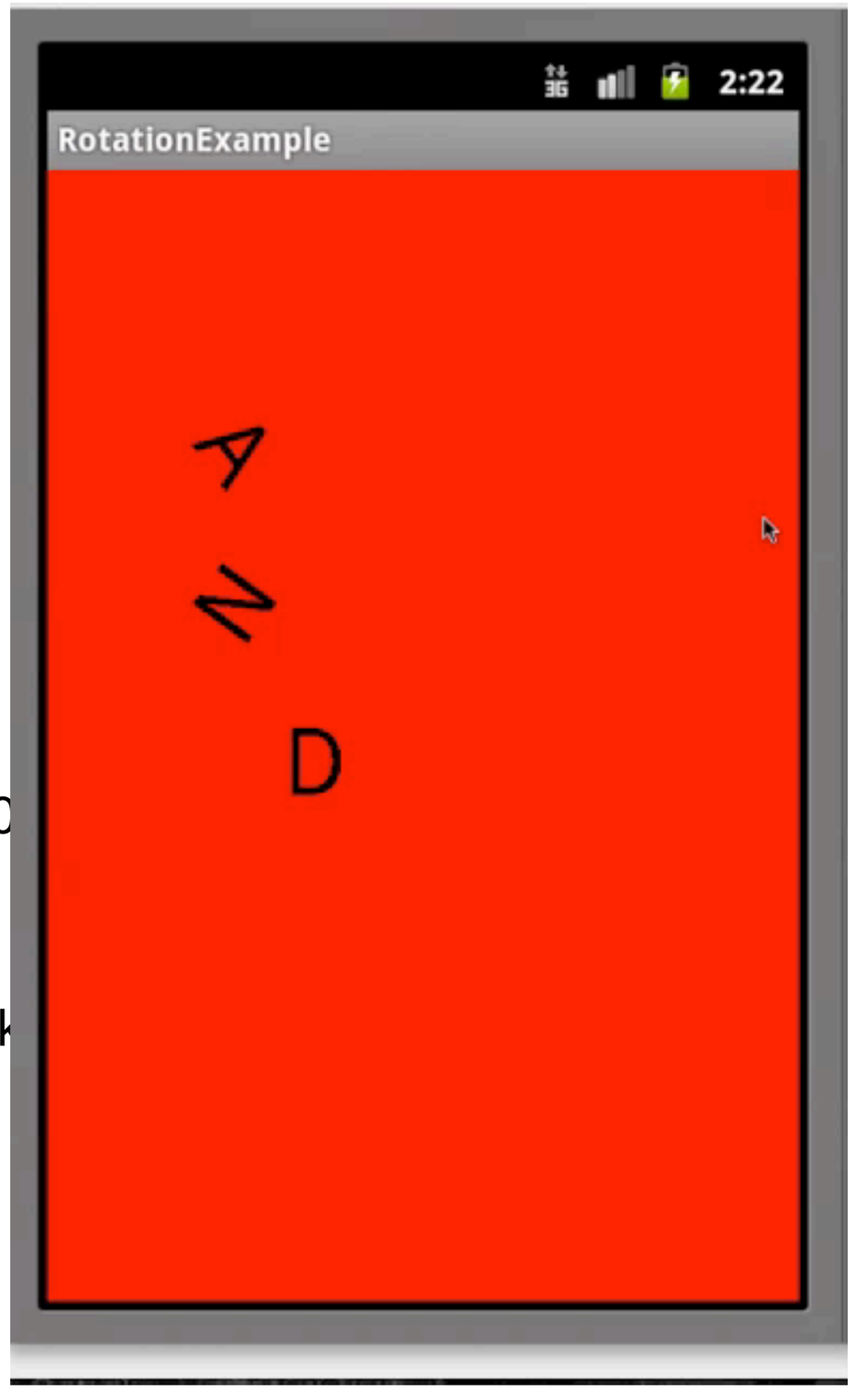


RotationView

```
public class RotationView extends View {
    private static Paint black;

    static {
        black = new Paint();
        black.setColor(Color.BLACK);
        black.setTextSize(60);

    }
    private float rotation = 0;

    public RotationView(Context context) {
        super(context);

    }
```

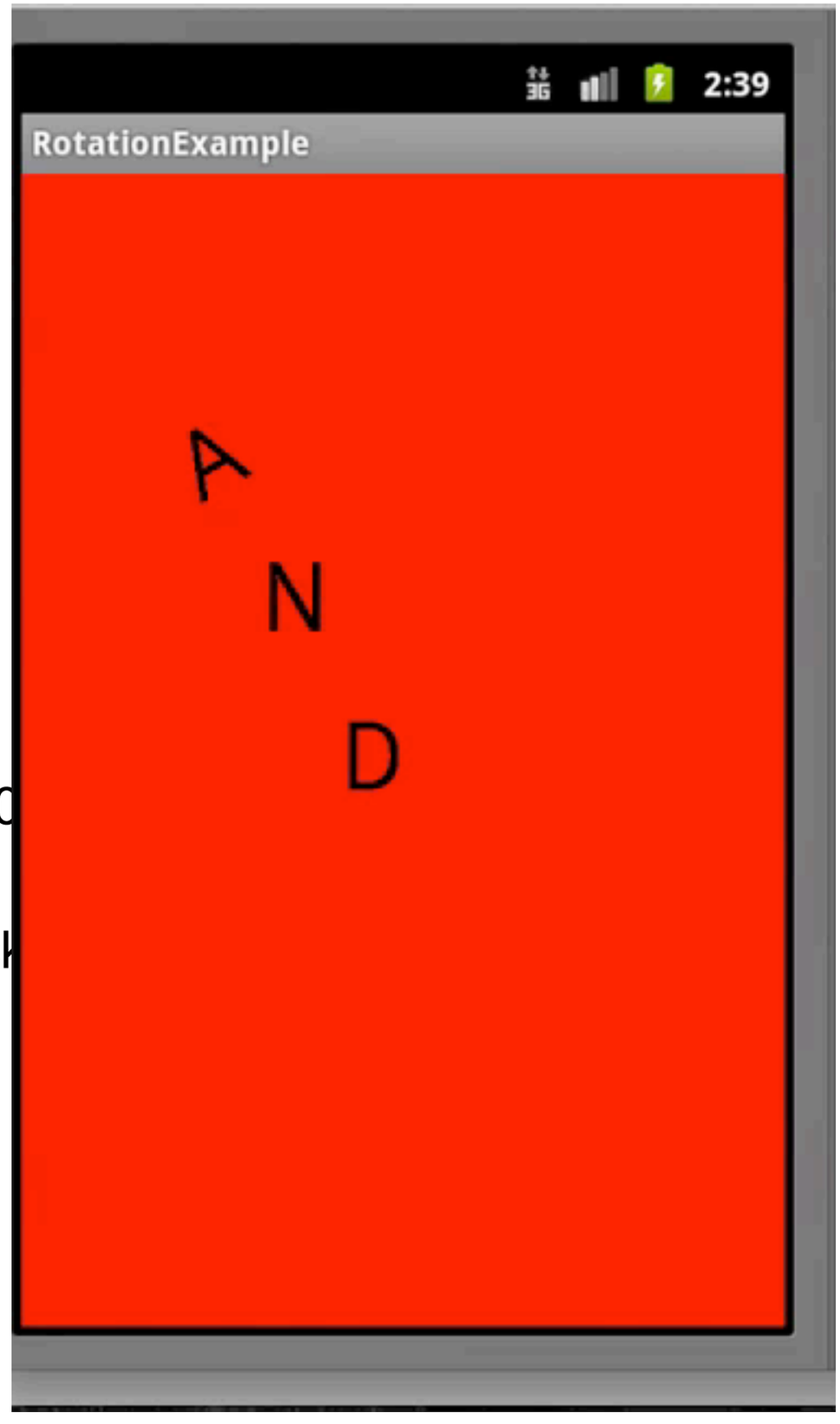
RotationView - onDraw

```
@Override
protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.RED);
        int x = 100;
        int y = 200;
        canvas.save();
        canvas.rotate(rotation, x + 20, y - 20);
        canvas.drawText("A", x, y, black);
        canvas.restore();
        canvas.save();
        canvas.rotate(-rotation, x + 20, y + 100 - 20
        canvas.drawText("N", x, y + 100, black);
        canvas.restore();
        canvas.drawText("D", x + 50, y + 200, black
        rotation += 2;
        invalidate();
}
```



Without Save

```
protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.RED);
        int x = 100;
        int y = 200;
        canvas.rotate(rotation, x + 20, y - 20);
        canvas.drawText("A", x, y, black);
        canvas.rotate(-rotation, x + 20, y + 100 - 20
        canvas.drawText("N", x, y + 100, black);
        canvas.drawText("D", x + 50, y + 200, black
        rotation += 2;
        invalidate();
}
```



Scaling

```
private float scale = 1;
private boolean isGrowing = true;

protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.RED);
        int x = 100;
        int y = 200;
        canvas.save();
        canvas.scale(scale, scale, x + 20, y - 20);
        canvas.drawText("A", x, y, black);
        canvas.restore();
        canvas.drawText("N", x, y + 100, black);
        canvas.drawText("D", x + 50, y + 200, black);
        if (scale > 10) isGrowing = false;
        if (scale < 0.5) isGrowing = true;
        if (isGrowing) scale += 0.1;
        else scale -= 0.1;
        invalidate();
}
```