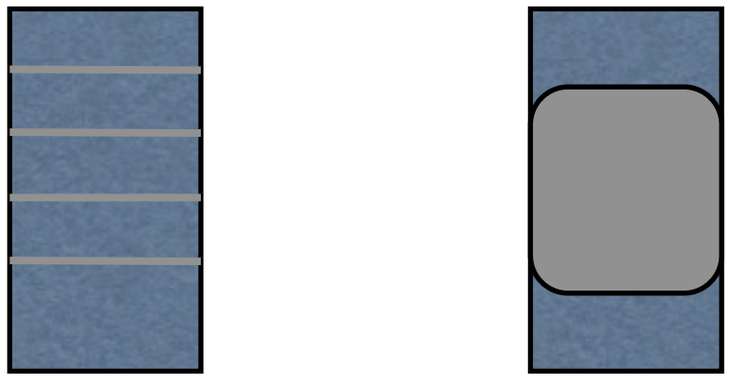
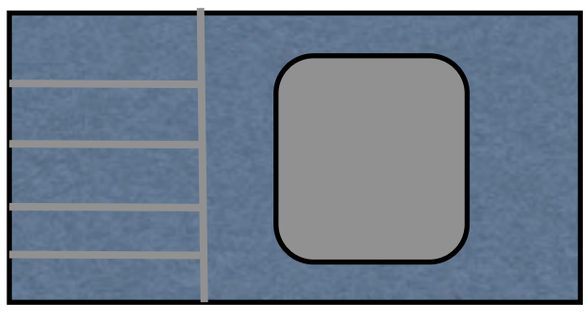
CS 646 Android Mobile Application Development
Spring Semester, 2015
Doc 13 Screen Sizes, Touch, Gestures
March 19, 2015

How does one Activity Support Different Views

Phone



Tablet



Issues

Detecting screen size - displaying correct layout

Different logic based on screen size

Screen Sizes
&
Resources

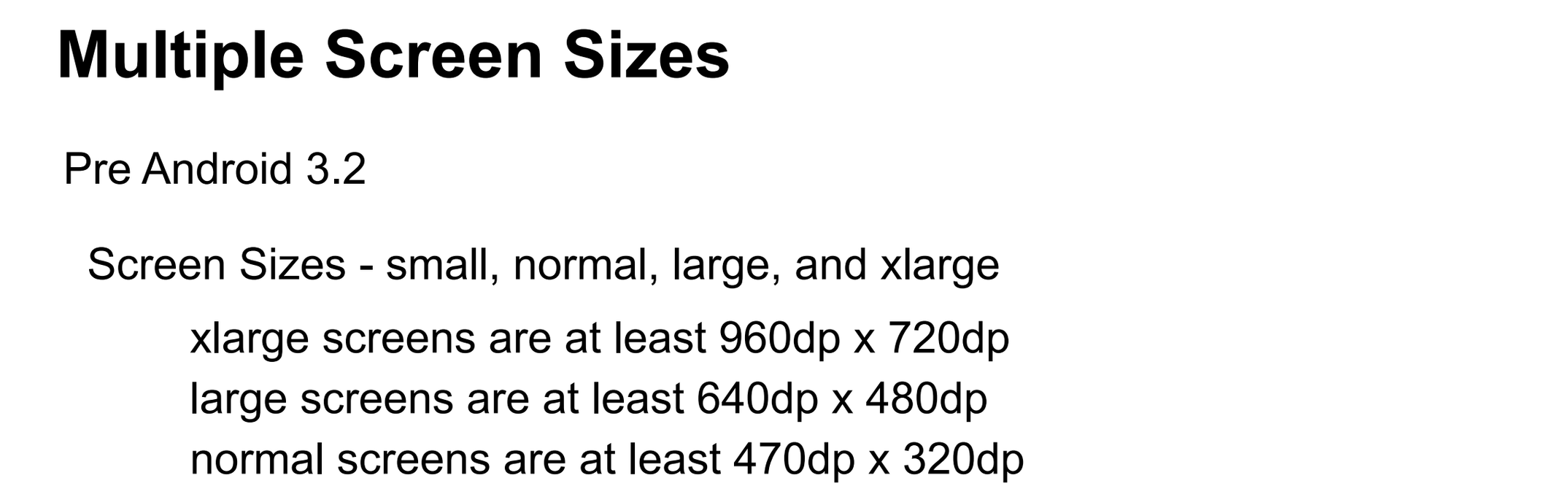
Multiple Screen Sizes

Pre Android 3.2

Screen Sizes - small, normal, large, and xlarge

xlarge screens are at least 960dp x 720dp
large screens are at least 640dp x 480dp
normal screens are at least 470dp x 320dp
small screens are at least 426dp x 320dp

Pixel Density - ldpi (low), mdpi, hdpi (high), xhdpi, tvdpi



Source:http://developer.android.com/guide/practices/screens_support.html

Supporting Multiple Screen Sizes

manifest file

Can declare which sizes/densities the app supports

layouts & resources

Different layout or resource files for different sizes/density

Layouts & Resources

res/layout-large-port-mdpi-qwerty/main.xml
res/layout-normal-land-mdpi-nokeys/main.xml
res/layout-small/main.xml
res/layout-land/main.xml

File with same name in each directory

Android will pick the one that matches current situation

Manifest File

```
<supports-screens
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:anyDensity="true"
    />

        Options

    android:resizeable=["true"| "false"]
    android:smallScreens=["true" | "false"]
    android:normalScreens=["true" | "false"]
    android:largeScreens=["true" | "false"]
    android:xlargeScreens=["true" | "false"]
    android:anyDensity=["true" | "false"]
    android:requiresSmallestWidthDp="integer"
    android:compatibleWidthLimitDp="integer"
    android:largestWidthLimitDp="integer"/>
```

Screen Sizes - Android 3.2+

Smallest Width (sw600dp) — res/layout-sw800dp-port

- Smallest Width — res/layout-sw800dp-land
- Does not change with device rotation

Available screen width (w720dp)

- Does change with device

Available screen height (h780dp)

- Does change with device

Directories allowed in res

animator/	XML files that define property animations.
anim/	XML files that define tween animations
color/	XML files that define a state list of colors
drawable/	Bitmap files
layout/	
menu/	XML files that define application menus
raw/	Arbitrary files to save in their raw form
values/	XML files that contain simple values
xml/	Arbitrary XML files

Qualifiers

MCC and MNC	
Language and region	en, fr, en-rUS
smallestWidth	sw<N>dp
Available width	w<N>dp
Available height	h<N>dp
Screen size	small, normal, large, xlarge
Screen aspect	long, notlong
Screen orientation	port, land
Dock mode	car, desk
Night mode	night, notnight
Screen pixel density (dpi)	ldpi, mdpi, hdpi, xhdpi, nodpi, tvdpi
Touchscreen type	notouch, stylus, finger
Keyboard availability	keysexposed, keyshidden, keyssoft
Primary text input method	nokeys, qwerty, 12key
Navigation key availability	navexposed, navhidden
Primary non-touch navigation method	nonav, dpad, trackball,wheel
Platform Version (API level)	V3, v4, etc

Source:http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources

Quantifier Order

Quantifiers must be used in order they are listed on previous slide

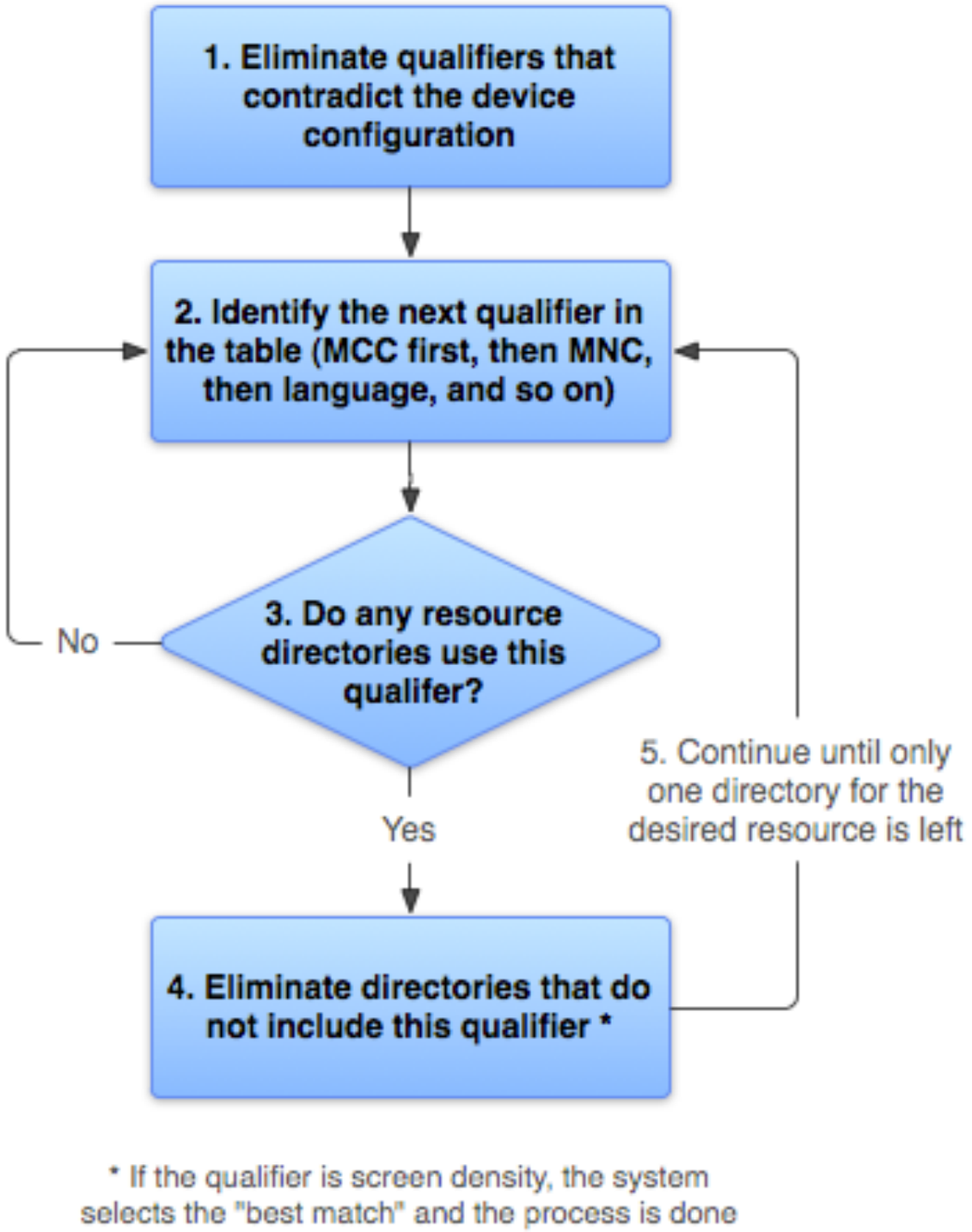
Legal

res/layout-large-port-mdpi-qwerty

Illegal

res/layout-large-mdpi-port-qwerty

Quantifier Match



Device

Locale = en-GB
Screen orientation = port
Screen pixel density = hdpi
Touchscreen type = notouch
Primary text input method = 12key

Resource Directories

drawable/
drawable-en/
drawable-fr-rCA/
drawable-en-port/
drawable-en-notouch-12key/
drawable-port-ldpi/
drawable-port-notouch-12key/

Thursday, March 19, 15

But this does not seem to work in all cases.

Issues

Detecting screen size - displaying correct layout

Different logic based on screen size

Use layout file in different res files

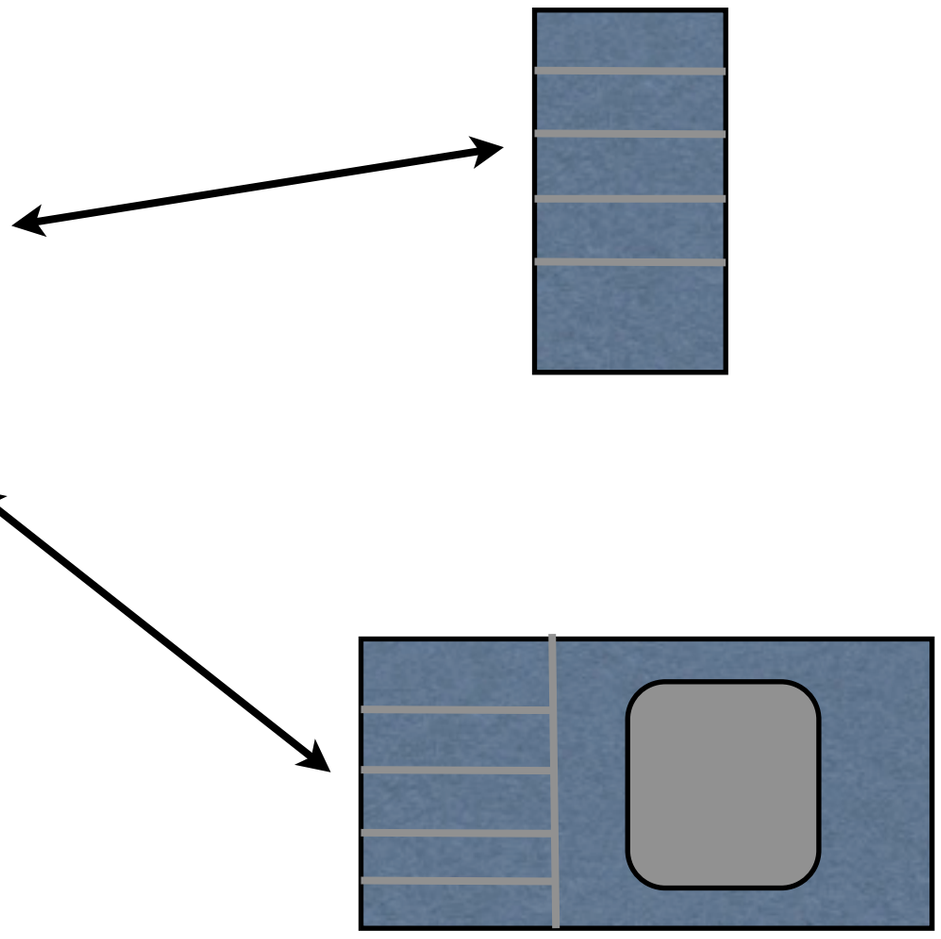
res

layout

main-layout.xml

layout-sw600dp

main-layout.xml



```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main-layout);
```

Beware of Images

Android 2.x does not know about sw600dp

res

layout

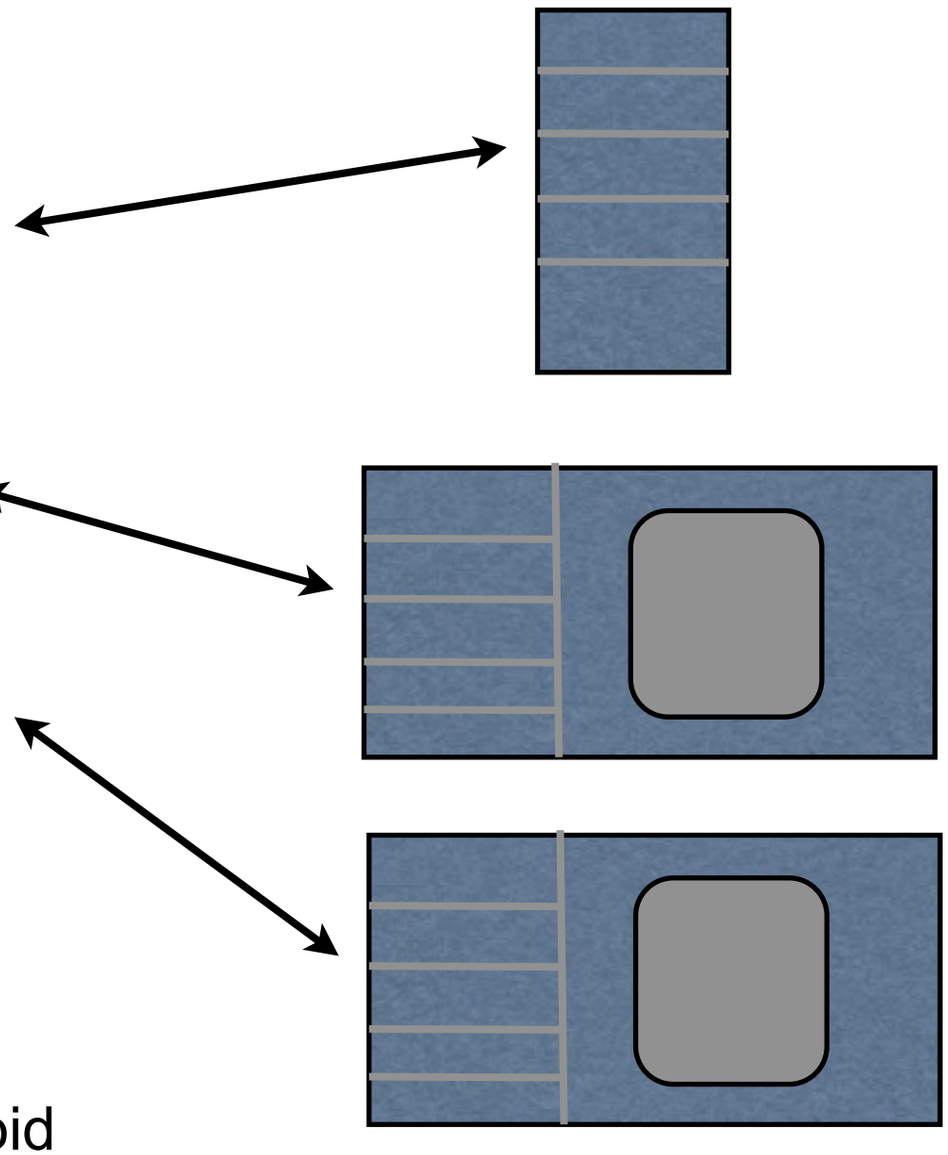
main-layout.xml

layout-sw600dp

main-layout.xml

layout-large

main-layout.xml



Add large for older Android

But have to duplicate same layout

Reference alias

A resource that points to actual resource

Place aliases in different folder

- Duplicates of aliases
- No duplicates of actual layouts

Using Alias

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);
```

res

layout

onepane.xml

twopane.xml

values

refs.xml

```
<resources>
    <item name="activity_list" type="layout">@layout/onepane</item>
</resources>
```

values-sw600dp

refs.xml

```
<resources>
    <item name="activity_list" type="layout">@layout/twopane</item>
</resources>
```

values-large

refs.xml

```
<resources>
    <item name="activity_list" type="layout">@layout/twopane</item>
</resources>
```

Issues

Detecting screen size - displaying correct layout

Different logic based on screen size

Activity Supporting Phone & Tablet

How to detect which layout using

How to handle different logic

How to detect which layout using

Two pane layout will have widget not in one pane layout
Search for that widget

```
private boolean mTwoPane;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ilist);

    if (findViewById(R.id.item_detail_container) != null) {
        mTwoPane = true;
    }
}
```

How to Handle Different Logic

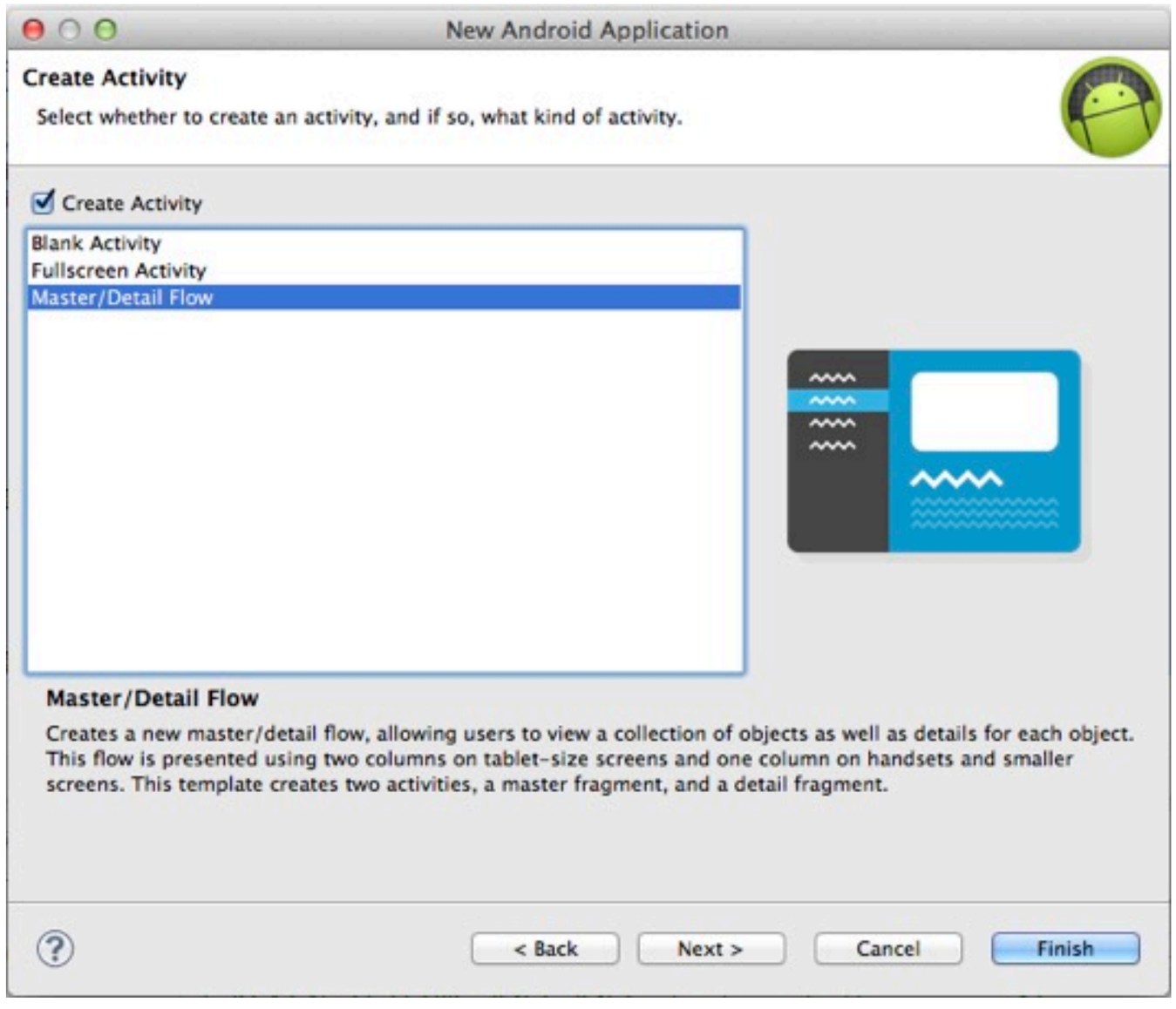
Each method becomes two methods

```
public void onItemSelected(String id) {
        if (mTwoPane) {
    //Handle the two pane case
        } else {
    //Handle the one pane case
        }
}
```

Master/Detail Flow Template

One pane on phone

Two pane on tablet



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>
```


Thursday, March 19, 15

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

Sony Ericsson

MotionEvent{2afd23a0 action=0 x=321.0 y=287.0 pressure=1.0 size=0.0}
down
number of touches; 1
x; 321.0 y: 287.0
MotionEvent{2afd23a0 action=261 x=321.0 y=287.0 pressure=1.0 size=0.0}
other action 261
number of touches; 2
x; 321.0 y: 287.0
x; 167.0 y: 308.0
MotionEvent{2afd23a0 action=2 x=320.0 y=287.0 pressure=1.0 size=0.0}
move 0
number of touches; 2
x; 320.0 y: 287.0
x; 167.0 y: 308.0
MotionEvent{2afd23a0 action=2 x=320.0 y=287.0 pressure=1.0 size=0.0}
move 0
number of touches; 2
x; 320.0 y: 287.0
x; 167.0 y: 308.0
MotionEvent{2afd23a0 action=2 x=249.0 y=294.0 pressure=1.0 size=0.0}
move 0
number of touches; 2
x; 249.0 y: 294.0
x; 150.0 y: 541.0
MotionEvent{2afd23a0 action=6 x=249.0 y=294.0 pressure=1.0 size=0.0}
other action 6
number of touches; 2
x; 249.0 y: 294.0
x; 150.0 y: 541.0
MotionEvent{2afd23a0 action=1 x=150.0 y=541.0 pressure=1.0 size=0.0}
UP
number of touches; 1
x; 150.0 y: 541.0

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

Thursday, March 19, 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

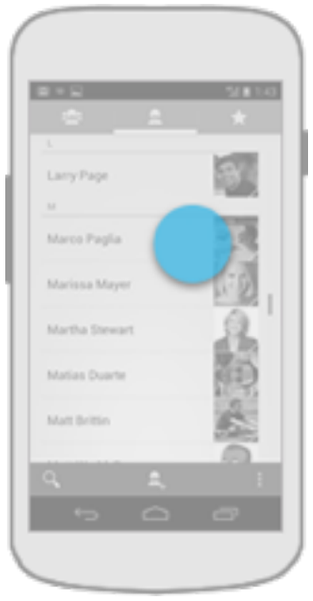
Finger A touches screen
Then Finger B touches screen
Finger A - index = 0
Finger B - index = 1
Finger A lifted off screen
Finger B - index = 0

But each finger is given an id which does not change

So use ids rather than location to track individual fingers

Gestures

Standard Android Gestures



Touch/
Tap

Long
Touch

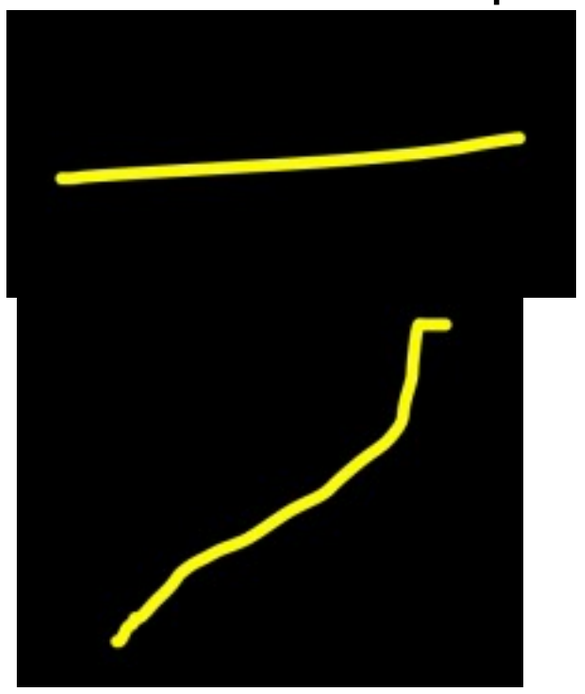
Zoom/
Pinch Open

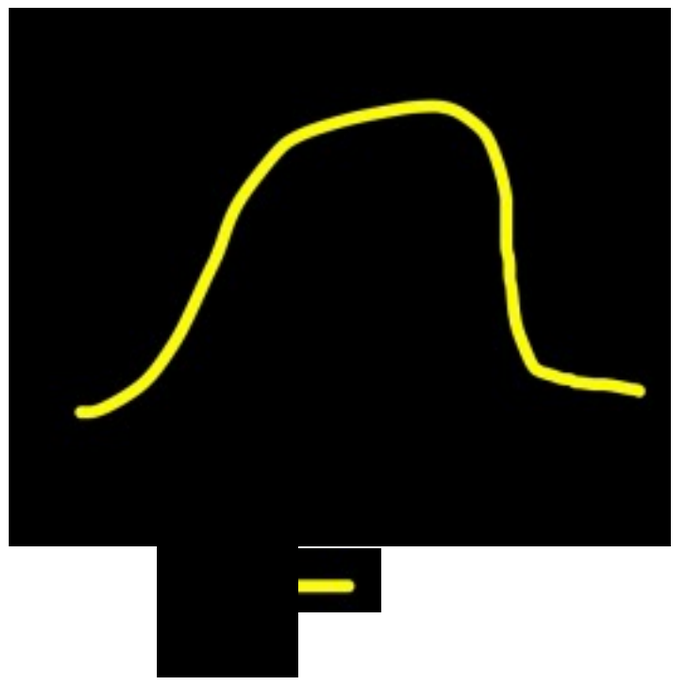
Pinch
Close

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"