UNIVERSITY of WASHINGTON

Android 250 - Lecture 6
Touch and Gestures

Margaret Maynard-Reid
May 4, 2015

Agenda

- Custom View, Canvas
- Touch & Multi-touch
- Gesture
- Drag & Drop

Sample Code

- SampleCustomView
- SampleTouch
- SampleGesture
- SampleDragDrop

Home 1 Graded

Overall everyone did well

Some common issues:

- Wrap Views in a container unnessarily
- Putting RelativeLayout in a LinearLayout
- Didn’t apply a style to EditText fields
- Created styles in a separate file
- 9-Patch should stretch single color area

Android Stories

- Build 2015 : Microsoft Announces Support For Android, iOS Apps On Windows Devices

Review from Last Week

- What is a Content Authority?
- What is a Content URI?
- What is a CursorLoader?
- What is a SimpleCursorAdapter?

Canvas

Canvas

A surface on which to do drawing

- Every View has a Canvas
- View.onDraw() passes in a Canvas
- Use it to create custom drawing or tweak an existing drawing
  - super is drawn depending on where you call it (before or after)
- Consists of:
  - A Bitmap to draw to
  - A drawing primitive to draw
  - A Paint on how to draw

Canvas Methods

From the names of the draw* methods, you should get a feel for what Canvas is useful for:

- drawArc()
- drawBitmap()
- drawCircle()
- drawColor()
- drawLine()
- drawLines()
- drawOval()
- drawPaint()
- drawPath()
- drawPoint()
- drawPoints()
- drawPosText()
- drawRect()
- drawRoundRect()
- drawText()
- drawTextOnPath()
- drawTextOnPath()

Paint

- Color and Style information used in drawing
  - Color
  - Stroke
  - Text Style
  - Shadows
- In other drawing models, this is sometimes referred to as a Pen

How to Draw?

- Obtain a **Canvas** class ← where and what to draw: a line, circle, word or shape
- Obtain a **Paint** class ← how to draw: fill the shape, text font, color etc.
- Use draw* methods to create your masterpiece

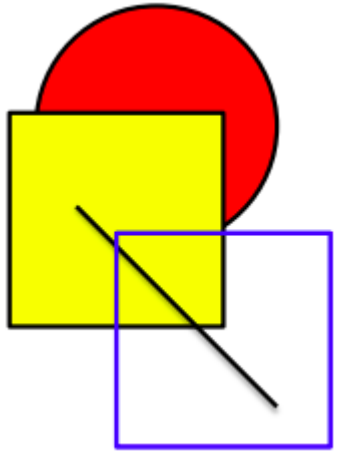
What model does this approach emulate?

How to Draw?

Drawing is layered on the canvas in the order methods are called

For example:

- drawCircle – Red Fill
- drawCircle – Black Stroke
- drawRect – Yellow Fill
- drawRect – Black Stroke
- drawLine – Black Stroke
- drawRect – Blue Stroke



Create a Custom View

- Extend from View
- Add CustomView to layout
  - via code
  - via XML ← make sure to include the full path
- Use Paint to set the paint
- Override onDraw()

Sample Code

- Walk through SampleCustomView

Break

Touch

Click vs. Touch

- You normally don't combine the two
- The distinction is really just a matter of handling
  - onClick()
  - onLongClick()
  - onKey()
  - onTouch()
- All just have separate event handlers
  - Need to capture a click? Register onClick()
  - Need to capture more touch events? Register onTouch()

onTouch()

- onTouch() is found within View.OnTouchListener
- Callback events of User press/move/release across the display area of the listening item
- Uses a Boolean to indicate whether you care or have consumed the event
  - If you return true, you have handled the event so it doesn't propagate
  - If you return false, you have not handled the event and you won't get notified anymore

Using Touch

To capture and process touch events

- Define an OnTouchListener
- Set your View.setOnTouchListener()
  - Optionally at ViewGroup.onInterceptTouchEvent (MotionEvent)
- Respond to events via onTouchEvent(MotionEvent)
  - Use getAction, getActionMasked(), getActionIndex(), getPointerId()

Pointer

The trace of an individual fingers or stylus.

- All active pointer information is stored in each MotionEvent
- This is what we use to track multi-touch events...

MotionEvents

The Touch callbacks return MotionEvents

- An Action Code
  - ACTION_DOWN
  - ACTION_MOVE
  - ACTION_UP
  - …
- Axis Values
- Pointer Information
- Potentially other stuff
- Pressure, Size, Orientation...

Single Touch MotionEvents

ACTION_DOWN	A pressed gesture has started, the motion contains the initial starting location
ACTION_MOVE	• A change has happened during a press gesture • Happens between ACTION_DOWN and ACTION_UP
ACTION_UP	A pressed gesture has finished, the motion contains the final release location and any intermediate points since the last down or move event
ACTION_CANCEL	The current gesture has been aborted or taken by a parent View

Multi-Touch MotionEvents

ACTION_POINTER_DOWN	● A non-primary pointer has gone down. ○ Extra finger touch is down on the screen ● Use getActionIndex() to get the pointer data
ACTION_POINTER_UP	● A non-primary pointer has gone up. ○ Extra touch is up on the screen ● Use getActionIndex() to get the pointer data

API 8

getAction() vs getActionIndex()

- getAction()
  - A mask of the current pointers
  - Returns the pointer index for ACTION_POINTER_DOWN or ACTION_POINTER_UP
- getActionMasked ()
  - Action without the pointer index information
  - Use getActionIndex() to return the index associated with pointer actions
- Once you have the index you can get extra information
  - getPointerId(int)
  - getX(int), getY(int)
  - getPressure(int)
  - getSize(int)
  - ...

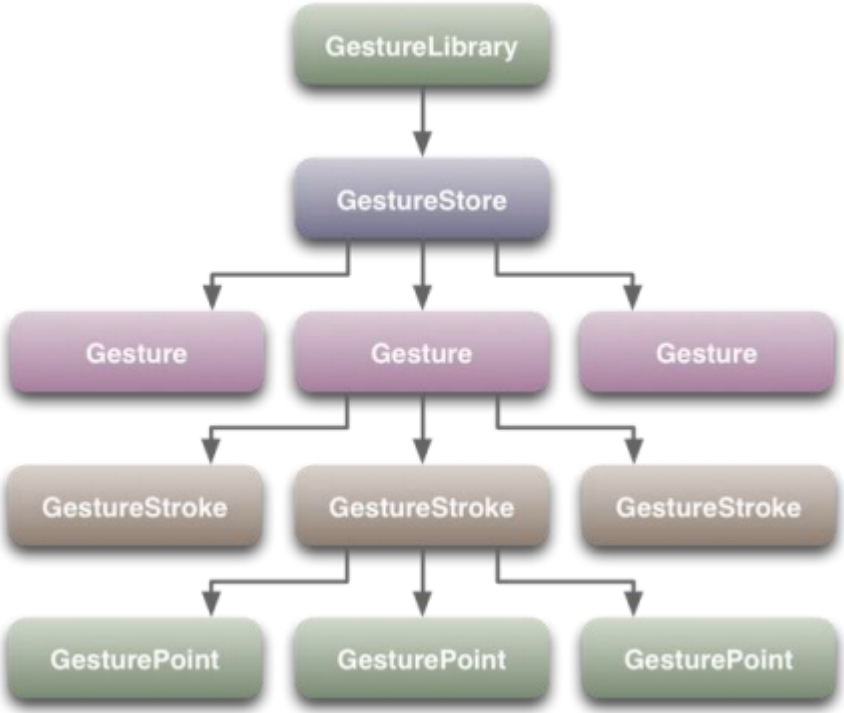
Sample Code

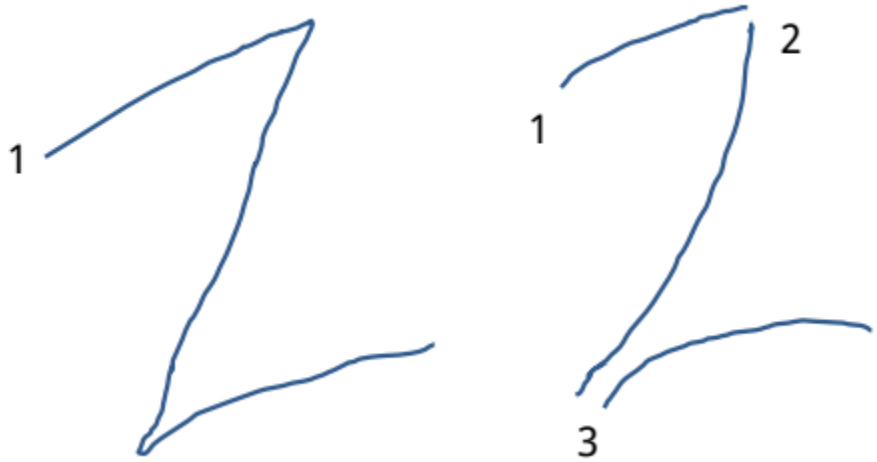
- Walk through SampleTouch
  - Paint with your figure
  - Single touch
  - Multi-touch (maybe?)

Gesture

Gesture

- Gesture - "A gesture is a hand-drawn shape on a touch screen. It can have one or multiple strokes. Each stroke is a sequence of timed points.”
  - Made up of one or multiple GestureStrokes
  - Each GestureStroke is made up of GesturePoints
- Gesture Store / Gesture Library – A collection of gestures
- Prediction - A possible match to one of the defined Gestures represented as a score, use the highest score


GestureLibrary
GestureStore
Gesture
Gesture
Gesture
GestureStroke
GestureStroke
GestureStroke
GesturePoint
GesturePoint
GesturePoint



Both of these are GestureStrokes comprising a Gesture

Gestures

- Supported from android.gesture package
- Somewhat interchangeable with MultiTouch
  - Pinch / Zoom are really handled as MultiTouch
- Simple vs. Custom Gestures

Simple Gestures

- Use a GestureDetector
  - Fling, DoubleTap, Scroll, etc.
- If your aren't doing anything custom, this is the way

Gestures vs. Swipes

Swipes, scrolls, etc. are just common Gestures detected and handled by a control

- ViewPagers
- ScrollViews
- Etc.

Custom Gesture

- Create Gestures
  - Manually via addGesture()
  - GestureBuilder
- Load the GestureLibrary
- Place a GestureOverlayView
  - Implement addOnGesturePerformedListener()

Sample Code

- Walk through SampleGesture

Break

Drag & Drop

Why Use Drag and Drop?

Intuitive for some things

- Reordering lists or collections
- Shopping carts
- Simple games
- ...

Drag & Drop

- Introduced in API 11
- Moving an "Object" from View to View
  - Really this is just ClipData

Drag and Drop States

- Started
  - The user has begun to drag.
  - An ACTION_DRAG_STARTED event is generated
- Continuing
  - The user continues the drag
  - Maybe an ACTION_DRAG_ENTERED if a listening View is entered
- Dropped
  - The user has dropped the drag shadow within the bounding box of a View that can accept the data.
  - And ACTION_DROP event is sent
- Ended
  - The drag has ended.
  - An ACTION_DRAG_ENDED event is sent

http://developer.android.com/guide/topics/ui/drag-drop.html

Implementing Drag and Drop

- Use DragShadowBuilder to show the View that is being moved
- Implement an View.OnDragListener
- Register each of the Views you want to drag and receive drops to the OnDragListener
- Use onTouch() or onLongClick() to startDrag()
- Handle DragEvents in onDrag()

Sample Code

- Walk through SampleDragDrop

Homework

- Walk through homework 3 requirements
- How is everyone doing on homework 2?
- Due dates:
  - Homework 2 due 5/18/2015
  - Homework 3 due 6/1/2015