Programming Handheld Systems
Multi-Touch & Gestures
Today's Topics

Motion Events
Touch Handling
Gestures
MotionEvent

Represents a movement in an input device reading pen, trackball, mouse, finger
MotionEvent

Action Code

State change that occurred

Action Values

Position and movement properties, such as time, source, location, pressure, and more

This lesson focuses on touch events read from a touch screen
MultiTouch

MultiTouch screens emit one movement trace per touch source

Individual touch sources are called pointers
MultiTouch

Each pointer has a unique ID for as long as it is active

MotionEvents can refer to multiple pointers

Each pointer has an index within the event, but that index may not be stable over time
Some MotionEvent actions

ACTION_DOWN
ACTION_POINTER_DOWN
ACTION_POINTER_UP
ACTION_MOVE
ACTION_UP
ACTION_CANCEL
Consistency Guarantees

For touch events, Android tries to guarantee that touches

Go down one at a time
Move as a group
Come up one at a time or are cancelled

Applications should be tolerant to inconsistency
**MotionEvent methods**

`getActionMasked()`

`getActionIndex()`

`getPointerId(int pointerIndex)`

`getPointerCount()`

`getX(int pointerIndex)`

`getY(int pointerIndex)`

`findPointerIndex(int pointerId)`
Handling Touch Events on a View

The View being touched receives `View.onTouchEvent(MotionEvent event)`
`onTouchEvent()` should return true if the `MotionEvent` has been consumed; false otherwise.
Handling Touch Events with a Listener

View.OnTouchListener defines touch event callback methods

View.setOnTouchListener() registers listener for Touch callbacks
Handling Touch Events with a Listener

onTouch() called when a touch event, such as pressing, releasing or dragging, occurs

onTouch() called before the event is delivered to the touched View

Should return true if it has consumed the event; false otherwise
Handling Multiple Touch Events

Multiple touches combined to form a more complex gesture.

Identify & process combinations of touches,

For example, a double tap:

```
ACTION_DOWN, ACTION_UP, ACTION_DOWN, ACTION_UP in quick succession
```
# Multi-touch Handling

<table>
<thead>
<tr>
<th>Action</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION_DOWN</td>
<td>0</td>
</tr>
<tr>
<td>ACTION_MOVE ...</td>
<td>0</td>
</tr>
<tr>
<td>ACTION_POINTER_DOWN</td>
<td>1</td>
</tr>
<tr>
<td>ACTION_MOVE ...</td>
<td>0,1</td>
</tr>
<tr>
<td>ACTION_POINTER_UP</td>
<td>0</td>
</tr>
<tr>
<td>ACTION_UP</td>
<td>1</td>
</tr>
</tbody>
</table>
Multi-touch Handling

<table>
<thead>
<tr>
<th>#1 touch</th>
<th>Action</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACTION_DOWN</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>ACTION_MOVE ...</td>
<td>0</td>
</tr>
<tr>
<td>#2 touch</td>
<td>ACTION_POINTER_DOWN</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ACTION_MOVE ...</td>
<td>0,1</td>
</tr>
<tr>
<td>#2 lift</td>
<td>ACTION_POINTER_UP</td>
<td>1</td>
</tr>
<tr>
<td>#1 lift</td>
<td>ACTION_UP</td>
<td>0</td>
</tr>
</tbody>
</table>
# Multi-touch Handling

<table>
<thead>
<tr>
<th>#1 touch</th>
<th>#2 touch</th>
<th>#3 touch</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION_DOWN</td>
<td>ACTION_POINTER_DOWN</td>
<td>ACTION_POINTER_DOWN</td>
</tr>
<tr>
<td>ACTION_DOWN</td>
<td>ACTION_POINTER_DOWN</td>
<td>ACTION_UP</td>
</tr>
<tr>
<td>ACTION_MOVE</td>
<td>ACTION_POINTER_UP</td>
<td>ACTION_POINTER_UP</td>
</tr>
<tr>
<td>ACTION_UP</td>
<td>ACTION_UP</td>
<td>ACTION_UP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION_DOWN</td>
<td>0</td>
</tr>
<tr>
<td>ACTION_POINTER_DOWN</td>
<td>1</td>
</tr>
<tr>
<td>ACTION_POINTER_DOWN</td>
<td>2</td>
</tr>
<tr>
<td>ACTION_MOVE</td>
<td>0,1,2</td>
</tr>
<tr>
<td>ACTION_POINTER_UP</td>
<td>1</td>
</tr>
<tr>
<td>ACTION_POINTER_UP</td>
<td>0</td>
</tr>
<tr>
<td>ACTION_UP</td>
<td>2</td>
</tr>
</tbody>
</table>
Touch Indicate Touch Location

Application draws a circle wherever the users touches the screen

Circle’s color is randomly selected

Redraws circles when user drags finger across the screen
The size of the circles are proportional to the number of currently active touches.
**GestureDetector**

**A class that recognizes common touch gestures**

**Some built-in gestures include confirmed single tap, double tap, fling**
GestureDetector

Activity creates a GestureDetector that implements the GestureDetector. OnGestureListener interface

Activity will receive calls to onTouchEvent() when activity is touched
onTouchEvent delegates call to GestureDetector.OnGestureListener
TouchGestureViewFlipper

Shows a TextView displaying a number
If the user performs a right to left "fling" gesture,
The TextView will scroll off the screen
A new TextView will scroll in behind it
Creating Custom Gestures

The GestureBuilder application lets you create & save custom gestures.
Comes bundled with SDK.
Creating Custom Gestures

GestureLibraries supports loading custom gestures & then recognizing them at runtime.
Creating Custom Gestures

Include a GestureOverlayView in your layout

The Overlay intercepts user gestures and invokes your application code to handle them
**GestureBuilder**

Stores gestures to `/mnt/sdcard/gestures`

Copy this file to `/res/raw` directory
Touch Gestures

Application displays a small View with a colored background.

User can swipe left and right to cycle between different candidate background colors.

Can make an check or X-like gesture to set or cancel the application’s current background color.
Next Time

MultiMedia