



Touch & Gestures

Programming the Android Platform

Topics

- MotionEvent
- Touch Handling
- Gestures

MotionEvent

- Represents a movement in an input device reading
 - e.g., pen, trackball, mouse, finger
- Records event's time, action, pressure, location, source, and more.

MotionEvent (cont.)

- Some common MotionEvent actions
 - ACTION_DOWN
 - ACTION_UP
 - ACTION_POINTER_DOWN
 - ACTION_POINTER_UP
 - ACTION_MOVE

MotionEvent (cont.)

- Some useful MotionEvent methods
 - `getPointerCount()`
 - `getActionIndex()`
 - `getActionMasked()`
 - `getX()`
 - `getY()`

Handling Touch Events

- `View.onTouchEvent()` delivers `MotionEvent` to the View being touched
- If subclassing, `onTouchEvent()`, return `true` if the `MotionEvent` has been consumed; `false` otherwise

Handling Touch Events

- `View.OnTouchListener` defines touch event callback methods
 - `onTouch()` called when user performs a touch event, such as pressing, releasing or dragging an item on the screen
 - Should return `true` if it has consumed the event
- `View.setOnTouchListener()` registers listener for Touch callbacks

Handling Touch Events (cont.)

- 1-touch – each touch is a single event
 - Process ACTION_DOWN, ACTION_MOVE & ACTION_UP independently
- 2+-touch – multiple touches combined to form a gesture
 - Identify & process combinations of touches such as ACTION_DOWN, followed by ACTION_POINTER_DOWN
 - Some common Gestures can be recognized by GestureDetector
 - E.g., double tap, fling, confirmed single tap, etc.

Multi-touch Handling

	Action	ID
#1 touch →	ACTION_DOWN	0
	ACTION_MOVE	0
#2 touch →	ACTION_POINTER_DOWN	1
	ACTION_MOVE	0
	ACTION_MOVE	0
#1 lift →	ACTION_POINTER_UP	0
#2 lift →	ACTION_UP	1

Multi-touch Handling (cont.)

	Action	ID
#1 touch →	ACTION_DOWN	0
	ACTION_MOVE	0
#2 touch →	ACTION_POINTER_DOWN	1
	ACTION_MOVE	0
	ACTION_MOVE	0
#2 lift →	ACTION_POINTER_UP	1
#1 lift →	ACTION_UP	0

Multi-touch Handling (cont.)

#1 touch →

#2 touch →

#3 touch →

#2 lift →

#1 lift →

#3 lift →

Action	ID
ACTION_DOWN	0
ACTION_POINTER_DOWN	1
ACTION_POINTER_DOWN	2
ACTION_MOVE	0
ACTION_POINTER_UP	1
ACTION_POINTER_UP	0
ACTION_UP	2

Touch Handling Example

```
public class IndicateTouchLocationActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        ...
        final FrameLayout frame = ...
        frame.setOnTouchListener(new OnTouchListener() {
            public boolean onTouch(View v, MotionEvent event) {
                int actionCode = event.getActionMasked();
                for (int i = 0; i < event.getPointerCount(); i++) {
                    switch (actionCode) {
                        case /* interesting motion event */ : {
                            // make a view showing touch location
                            frame.addView(/* new view */);
                        }
                    }
                }
            }
        });
        ...
    }
}
```

GestureDetector Example

```
public class ViewFlipperTestActivity extends Activity {  
    private ViewFlipper mFlipper;  
    private int mCurrentLayoutState;  
    private GestureDetector gd;  
    public void onCreate(Bundle savedInstanceState) {  
        ...  
        mFlipper = (ViewFlipper) findViewById(R.id.view_flipper);  
        mCurrentLayoutState = 0;  
        ....  
    }  
}
```

GestureDetector Example

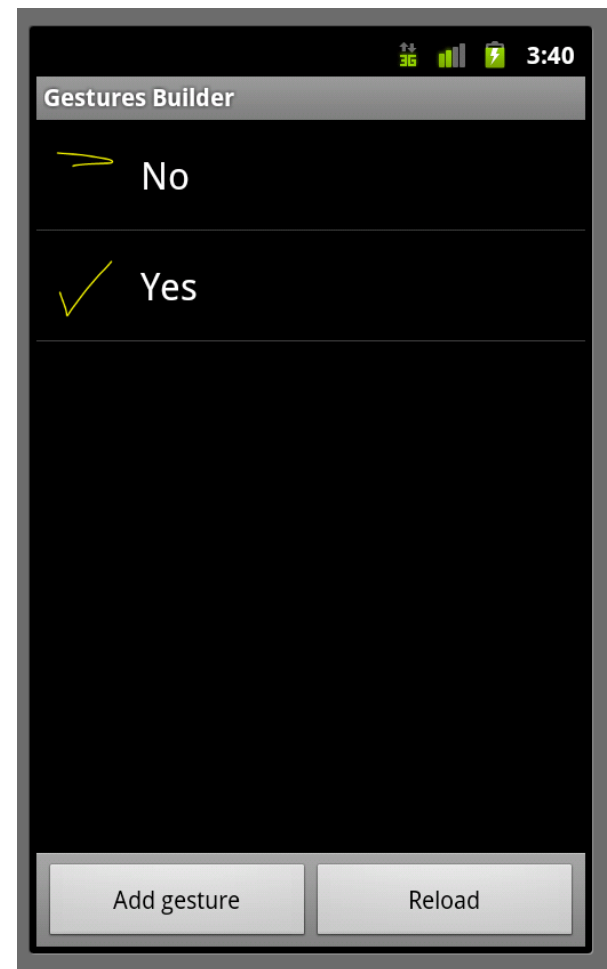
```
...
gd = new GestureDetector(this,
    new GestureDetector.SimpleOnGestureListener() {
    public boolean onFling( MotionEvent e1, MotionEvent e2,
        float velocityX, float velocityY) {
        if (velocityX < -10.of) { /* change view */ }
        return true;
    }
});
}
public boolean onTouchEvent(MotionEvent event) {
    return gd.onTouchEvent(event);
}
```

Custom Gestures

- GestureBuilder allows developers to create & save custom gestures
- GestureLibraries supports loading custom gestures & recognizing them at runtime
- GestureOverlayView intercepts user gestures and invokes application code to handle them

GestureBuilder

- Comes bundled with emulator
- Stores gestures to /mnt/sdcard/gestures
- Copy this file to /res/raw directory



GestureOverlayView

- View intercepts gestures
- Invokes OnGesturePerformedListener

GestureActivity Example

```
public class GesturesActivity extends Activity implements
    OnGesturePerformedListener {
    private GestureLibrary mLibrary;
    public void onCreate(Bundle savedInstanceState) {
        ...
        mLibrary = GestureLibraries.fromRawResource(this, R.raw.gestures);
        if (!mLibrary.load()) { finish(); }
        GestureOverlayView gestures =
            (GestureOverlayView) findViewById(R.id.gestures);
        gestures.addOnGesturePerformedListener(this);
        ...
    }
    ...
}
```

GestureActivity Example (cont.)

```
...
public void onGesturePerformed(
    GestureOverlayView overlay, Gesture gesture) {
    ArrayList<Prediction> predictions = mLibrary.recognize(gesture);
    if (predictions.size() > 0) {
        Prediction prediction = predictions.get(0);
        if (prediction.score > 1.0) {
            // Check prediction.name and take appropriate action
        }
    }
}
}
```

Source Code Examples

- TouchIndicateTouchLocation
- TouchGestureViewFlipper
- TouchGestures