SCIENCES SORBONNE UNIVERSITÉ

Gesture Detection

F. Kordon - Sorbonne Université - CC2018

Renault@lrde.epita.fr

Touch Detection

Implement onTouchEvent(MotionEvent)

- Return true if the event has been capture, false otherwise

Event	Description
MotionEvent.ACTION_DOWN	New touch
MotionEvent.ACTION_MOVE	The touch is moving
MotionEvent.ACTION_UP	The finger is up
MotionEvent.ACTION_CANCEL	Current touch is cancelled (an another touch as the focus)
MotionEvent.ACTION_POINTER_DOWN	New touch
MotionEvent.ACTION_POINTER_UP	End of a touch (multitouch)

How to build a Drawing Canvas? (1/3)

There is no drawing canvas

- But building a fresh one is easy!
  - **Detect touches down, up and move**
  - **updates the view according to these**
  - **View.onDraw() allows to draw on a View**

Extend the View Class

```
public class DrawCanvas  extends View {
    private Paint paint = null;
    private Path path = null;

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

How to build a Drawing Canvas? (2/3)


```
public void reset(){
    path = new Path();
    paint = new Paint();
    paint.setAntiAlias(true);
    paint.setStrokeWidth(6f);
    paint.setColor(Color.BLACK);
    paint.setStyle(Paint.Style.STROKE);
    paint.setStrokeJoin(Paint.Join.ROUND);
    invalidate();
}

@Override
protected void onDraw(Canvas canvas) {
    canvas.drawPath(path, paint);
}

@Override
protected void onDraw(Canvas canvas) {
    canvas.drawPath(path, paint);
}
```

How to build a Drawing Canvas? (3/3)


```
@Override
public boolean onTouchEvent(MotionEvent event) {
    float eventX = event.getX();
    float eventY = event.getY();
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            path.moveTo(eventX, eventY);
            return true;
        case MotionEvent.ACTION_MOVE:
            path.lineTo(eventX, eventY);
            break;
        case MotionEvent.ACTION_UP:
            // nothing to do
            break;
        default:
            return false;
    }
    // Schedules a repaint.
    invalidate();
    return true;
  }
}
```

Instanciate the Component

How to instantiate programmatically a *new* Component

```
setContentView(new DrawCanvas(this, null));
```


Though the graphical interface (XML)

- Declare the new component in res/values/attr.xml

```
<resources>
      <declare-styleable name="DrawCanvas">
      </declare-styleable>
</resources>
```

- The component is now available for drag-and-drop in the Visual Editor Custom > CustomView

Handle Multitouch

Implement onTouchEvent(MotionEvent event)

- Get the action

```
int maskedAction = event.getActionMasked();
```

- Get the number of actions in progress

```
event.getPointerCount()
```

- Get the index of a new touch (ACTION_DOWN, ACTION_POINTER_DOWN) or a touch released (ACTION_UP, ACTION_POINTER_UP, ACTION_CANCEL)

```
event.getActionIndex()
```

- Get the coordinates of an event

```
event.getX(i);
event.getY(i);
```

Handle Multitouch

Implement onTouchEvent(MotionEvent event)

- Get the action

```
int maskedAction = event.getActionMasked();
```

- Get the number of actions in progress

```
event.getPointerCount()
```

- Get the index of the pointer associated with the action. Typically, ACTION_POINTER_DOWN, ACTION_POINTER_UP (ACTION_UP, ACTION_POINTER_UP, ACTION_CANCEL)

```
event.getActionIndex()
```

- Get the coordinates of an event

```
event.getX(i);
event.getY(i);
```

Works as in single touch mode

You only have to keep touches in memory!

Predefined gestures

GestureDetector helps to catch predefined gestures

- All events detected by onTouchEvent must be forwarded to GestureDetector
- Warning: onTouchEvent must call the super constructor to trigger the callback

Implement GestureDetector.OnGestureListener

- onDown: detects a single touch
- onShowPress: detects a touch not yet up/cancelled
- onSingleTapUp: detects a single tap
- onScroll: detects a scrolling move
- onLongPress: detects a long press
- onFling: detects a dynamic move

Implement Swipe Up-Down-Left-Right


```
private static final int SWIPE_MIN_DISTANCE = 120;
private static final int SWIPE_THRESHOLD_VELOCITY = 200;

@Override
public boolean onFling(MotionEvent e1, MotionEvent e2,
                       float velocityX, float velocityY) {
    try {
     if(e1.getX() - e2.getX() > SWIPE_MIN_DISTANCE &&
        Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY)
         // Left Swipe ...
     else if (e2.getX() - e1.getX() > SWIPE_MIN_DISTANCE &&
              Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY)
       // Right Swipe ...
     if(e1.getY() - e2.getY() > SWIPE_MIN_DISTANCE &&
        Math.abs(velocityY) > SWIPE_THRESHOLD_VELOCITY)
       // Swipe up ...
     else if (e2.getY() - e1.getY() > SWIPE_MIN_DISTANCE &&
              Math.abs(velocityY) > SWIPE_THRESHOLD_VELOCITY) {
       // Swipe down ...
    } catch (Exception e) { }
    return false;
}
```

Other Gestures

Scaling effects

- ScaleGestureDetector: works similarly to GestureDetector and must analyse events received in onTouchEvent

- Implement ScaleGestureDetector.OnScaleGestureListener
  - **onScale, onScaleBegin, onScaleEnd**

- There is a default listener ScaleGestureDetector.OnScaleGestureListener if we only want to track special events

Double tap

- ScaleGestureDetector.OnDoubleTapGestureListener
  - **onDoubleTap, onDoubleTapEvent, onSingleTapConfirmed**

Summary

Detect single or multiple touches

- implement onTouchEvent
- For multi-touches the developper must keep track of previous events

Detecting touches helps to define a drawing canvas

- A new component can then be used directly in the GUI Editor

Predefined events can be easily detected

- onFling can capture almost all events
- Scaling, tap, doubletap, can easily be capture without reinventing the wheel