LE/EECS 4443: Mobile User Interfaces (LAB)

Week 5: Touch, Multi-Touch, & Gesture Detection

Shogo Toyonaga[1]

[1]Lassonde School of Engineering
York University

January 22, 2025

Table of Contents

Introduction

By the end of this tutorial, you will be able to...

1. Write applications that leverage touch gestures
2. Implement your own touch controller(s)
3. Write applications that support complex gestures
4. Load images into an ImageView with an Intent
5. Play a sound (tone)

Introduction

Remark

A touch gesture occurs when a user places one or more fingers on the touchscreen and your app interprets this pattern of touches as a gesture. There are two phases to gesture detection:

1. Gathering Touch Event Data
2. Interpreting the data to determine whether it meets the criteria for the gestures you app supports

https://developer.android.com/develop/ui/views/touch-and-input/gestures/detector

Supporting Touch Events

1. Implement View.OnTouchListener
2. Override:
   - public boolean onTouch(View view, MotionEvent motionEvent)
   - public void onPointerCaptureChanged(boolean hasCapture)
3. Initialize Views in the Controller
4. Set OnTouchListener to the View which supports touch events.

onTouch

- Returning true indicates that the event has been "consumed", which is to say, no further processing is needed.
- If false is returned, then further processing is needed, for example, in a parent view (if any) or in the Activity hosting the view. For further discussion, see the description of onTouchEvent in the Demo_Touch Activity API.

What is MotionEvent?

- An Object used to report movement events
- May hold either absolute or relative movements and other data, depending on the device:
    1. ACTION_DOWN
    2. ACTION_UP
    3. ACTION_MOVE

       ⋮

https://developer.android.com/reference/android/view/MotionEvent

Read the documentation to understand the differences between supporting touch versus multi-touch with MotionEvent.

Gestures

- Gestures represent a high-level sequence of touch events.
  1. Long Press
  2. Double Tap
  3. Drag
  4. Fling
- There are two approaches to supporting gestures in your application:
  1. Custom Support with onTouch and MotionEvent
  2. Use Androids Gesture Detector classes & interfaces.

Detecting Gestures

1. GestureDetector Class (Extends · · ·)
   - Touch: GestureDetector.SimpleOnGestureListener
   - Multi-Touch: ScaleGestureDetector.SimpleOnScaleGestureListener
2. Gesture Interface (Implements · · ·)
   - Touch: GestureDetector.OnGestureListener
   - Multi-Touch: ScaleGestureDetector.OnScaleGestureListener

Putting it all together...

- Download Demo_Gestures and test the following gestures on the ImageView:
  1. onDoubleTap()
  2. onFling()
  3. onLongPress()
- Look at the source code! Try to reverse-engineer it.

  Hint: Some of the code will be very helpful for Lab #3

Gestures: Zooming

- To support zooming through any of the simple or scale gesture detectors, it is important to consider the concept of a focus point.
- The focus point is where the user is looking. This means that once you scale the image, it must be shifted in a certain way to re-position it to the area that you initially tapped.
- Further Rationale: Touch events occur with respect to the View and not the image. This means that when we expand an image after tapping (or pinching), we have to shift its position to maintain a focus point.

Remark

Thank you for your attention!
Questions?