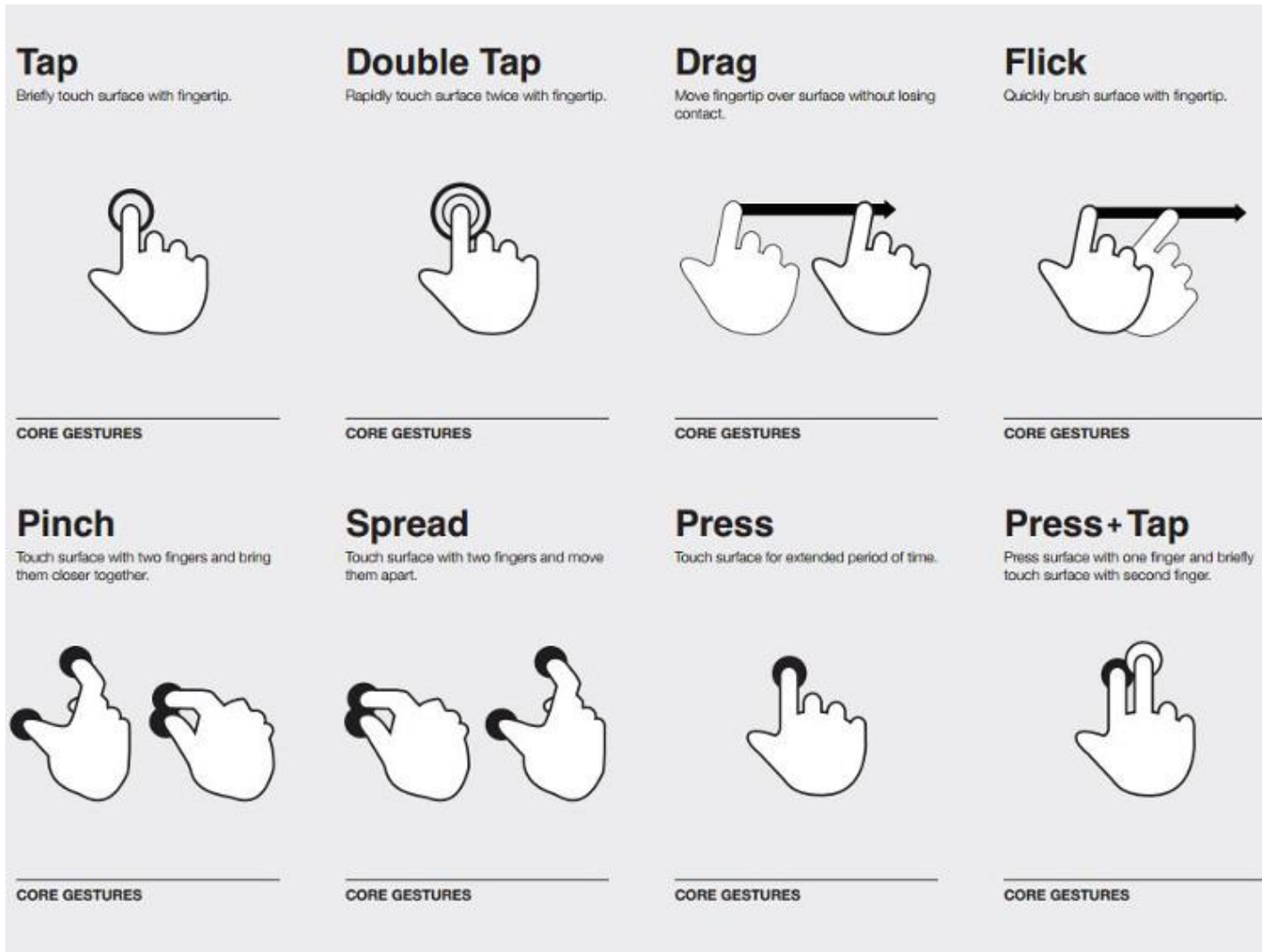


G53MDP

Mobile Device Programming

Lecture 16 – Gestures

Touch Gestures



What else? rotate, scroll, multiple finger interaction, etc.

Learning Outcomes

- Know how to use:
 - MotionEvent object
 - Built-in gesture detector
 - onTouchEvent
- Implementation of:
 - Single-touch gestures (e.g. scroll, fling)
 - Multi-touch gestures (e.g. pinch to zoom, rotate)

Programming for Touch

- Similar to programming for mouse
 - `onClick`, `onMouseDown`, `onMouseMove`
- Handle a sequence of events
 - `TouchBegin`
 - `TouchMoved`
 - `TouchEnded`
- A gesture is a series of touch events that occur over a period of time

Touch Events

- Two ways to get touch events:
 - Register a new `OnTouchListener` with a view, (with `setOnTouchListener(...)`)
 - Implement `onTouchEvent()` in a View
- Either way, we are delivered a series of `MotionEvent`s

MotionEvent

- This object encapsulates information of
 - Touch events
 - A touch begins (ACTION_DOWN)
 - The finger moves (ACTION_MOVE)
 - The touch ends (ACTION_UP)
 - X,Y coordinates of the touch, information about pressure, size and orientation, etc.
 - Additional information for multitouch (e.g. pointer id, action index, etc.)

Action Down

- A gesture starts when a finger is pressed
- A MotionEvent is generated for this ACTION_DOWN
- Can find the action by calling `getAction()`
- This can also have the identifier of the 'pointer' so use `getActionMasked()` instead (for multi-touch).

Action Move

- As the finger moves, a series of ACTION_MOVE events will be sent
- Can find the new position using getX() and getY() (return floats)
- Note that Android may bundle up a series of touch events (i.e. Ability to get 'historic' touches)

Action Up

- A gesture ends in three ways:
 - ACTION_UP event (i.e. last finger has been taken off the display)
 - ACTION_CANCEL event (e.g. phone rings)
 - ACTION_OUTSIDE event if the finger moves outside the relevant view

Single Touch

- A touch gesture is formed by
 - A single ACTION_DOWN
 - Zero or more ACTION_MOVE
 - An ACTION_UP to finish
- Single-touch interactions
 - Positions are tracked to move an object
 - Use movement velocity for a swipe / fling

Dragging / Scrolling

- Store the original x, y touch location from ACTION_DOWN
- Calculate changes from stored value and value returned from ACTION_MOVE or ACTION_UP
- Adding the coordinate changes to the original object location

Swipe / Fling

- Rather than moving the object, calculate the velocity and direction
- On ACTION_UP, continue to move the object with that velocity
- Gives the user an obvious visual feedback of “flinging” UI elements across the screen

Demo

Android provides built-in callbacks for common gestures via `GestureDetector.SimpleOnGestureListener` class

- `onSingleTap`
- `onDoubleTap`
- `onShowPress`
- `onLongPress`
- `onScroll`
- `onFling`



Multi-touch


- Very similar to single touch
- Same sequence of events as before with a few more events (e.g. ACTION_POINTER_DOWN and ACTION_POINTER_UP)
- Support for 256 touch, but some Android devices only support 2 touch

Which finger?

- **Index:** position within an array in a MotionEvent
- **ID:** Unique for each pointer to allow tracking individual pointer across the entire gesture
- Number of pointers can change as fingers are lifted or placed, so does the indices of the pointers.
- Each pointer is given an id that won't change
 - Need to track both the id and past locations of pointers to move things about

Which finger?

First finger down



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



Last finger up

Example

```
private int mActivePointerId;

public boolean onTouchEvent(MotionEvent event) {
    ....
    // Get the pointer ID
    mActivePointerId = event.getPointerId(0);

    // ... Many touch events later...

    // Use the pointer ID to find the index of the active pointer
    // and fetch its position
    int pointerIndex = event.findPointerIndex(mActivePointerId);
    // Get the pointer's current position
    float x = event.getX(pointerIndex);
    float y = event.getY(pointerIndex);
}
```

onTouchEvent

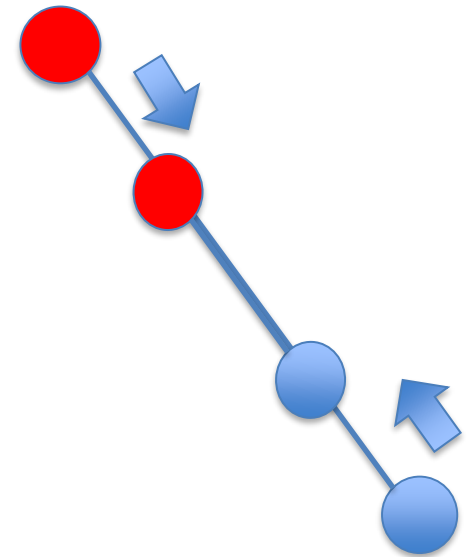
```
public boolean onTouchEvent(MotionEvent event) {  
  
    int pointerIndex = event.getActionIndex();  
    int pointerId = event.getPointerId(pointerIndex);  
    int maskedAction = event.getActionMasked();  
  
    switch (maskedAction) {  
  
        case MotionEvent.ACTION_DOWN:  
        case MotionEvent.ACTION_POINTER_DOWN:  
        case MotionEvent.ACTION_MOVE:  
        case MotionEvent.ACTION_UP:  
        case MotionEvent.ACTION_POINTER_UP:  
        case MotionEvent.ACTION_CANCEL:  
  
    }
```

Pointers to Gestures

- Maintain state of pointer IDs
 - Track movement of multiple fingers
- How do we convert these into gestures?
 - Pinch to zoom
 - Two-finger rotation
- Little SDK support for specific gestures
 - Implement ourselves with some simple trigonometry

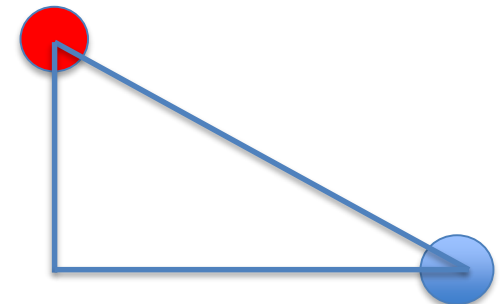
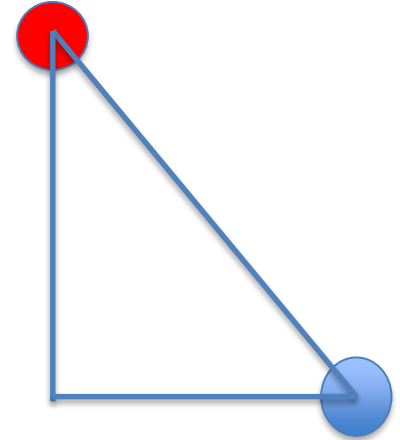
Pinch to Zoom

- Obtain the IDs of the two pointers
- Obtain the location of each pointer
- Calculate and store the distance between the two pointers
- The ratio of new distance to the old one gives the zoom ratio
- Rescale the object using the ratio



Two-finger rotation

- Obtain the IDs of the two pointers
- Use the vertical and horizontal location difference to calculate the initial angle
- Obtain the new locations of the two pointers to derive the new angle
- Object (e.g. photos) can be rotated using the angle difference



Multi Touch Demo



Summary

- `OnTouchEvent`, `OnTouchListener` and `GestureDetector` can be used for gesture detection & implementation.
- `MotionEvent` object include coordinates of touch, index & id of pointers, which is used to implement customised gestures.
- Implementation of pinch to zoom, rotation, fling, scroll.
- Additional to this lecture: custom gestures can be created and saved as binary resources into `res/raw` (i.e. A pattern of movements)

References

- <https://www.ifixit.com/Teardown/iPhone+6s+Display+Teardown/49951>
- <http://developer.android.com/training/gestures/index.html>
- <http://developer.android.com/training/gestures/multi.html>