Lec10: Touch

Ekarat Rattagan

Some slides obtained from http://www.cs.cornell.edu/courses/CS4152

The cheap way out

The cheap way out

- (Multi) Touch Controls
  - Pointing, dragging
  - Clicking, selecting
  - More advanced gestures
- Accelerometer Support
  - Tilting
  - Rotating

Balancing Multi-touch

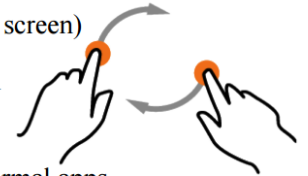
- Mouse games are "balanced" for a single pointer
  - Multitasking requires a lot of back and forth
  - Challenge is to do actions in an efficient order
- Multitouch eliminates this challenge
  - Can quickly move fingers anywhere
  - Can use multiple fingers at once
  - **Example**: Whack-a-Zombie
- Need to rethink gameplay

Size Matters

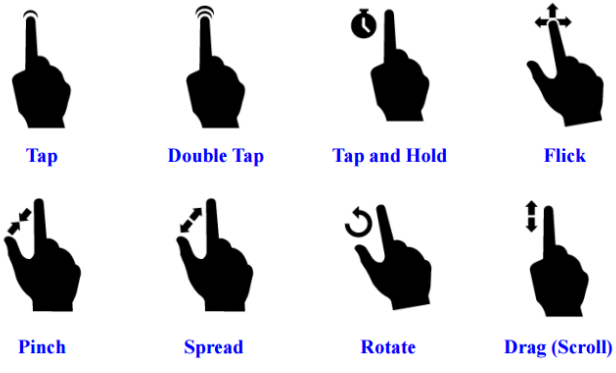
- Small screen makes multitouch *hard*
  - True multitouch only on a tablet
  - Phones are largely limited to gestures
- Fingers are **fatter** than mouse pointers
  - I did not mean to click that!
  - Also, fingers cover up the screen
  - Touch needs to be very forgiving

Touch: Gestures

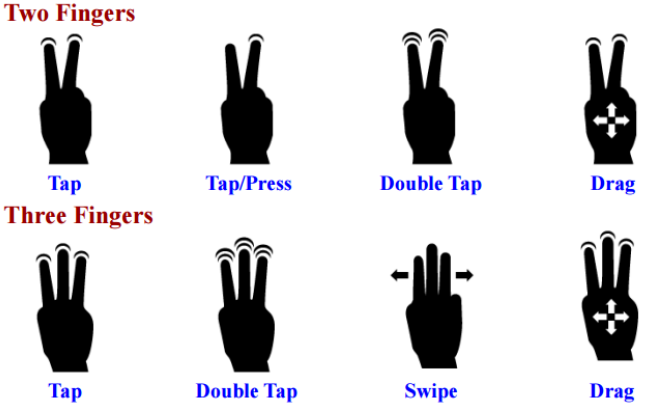
- Can also leverage device **gestures**
  - Manipulation strokes common to device
  - **Example**: Pinching for zoom
  - **Example**: Rotating (object, screen)
- Natural for camera control
- **Design Approach**:
  - Think about how used in normal apps
  - How do you leverage this in a game?



Basic touch mechanics



Multi-touch mechanics



Touch programming

GestureDetector

- Android class receives motion events
- Implement GestureListener interface, etc.
- E.g.,

  Class

  GestureDetector

  implements

  GestureDetector.OnGestureListener,
  GestureDetector.OnDoubleTapListener

Touch programming

Methods in class

- onDown
- onSingleTapConfirmed
- onSingleTapUp
- onShowPress
- onDoubleTap
- onDoubleTapEvent
- onLongPress
- onScroll
- onFling

Touch programming

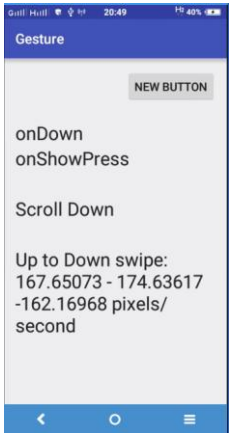
```
public boolean onScroll
(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)
{
   Log.d("Gesture ", " onScroll");

     if (e1.getY() < e2.getY())
     {
          Log.d("Gesture ", " Scroll Down");
     }

     if(e1.getY() > e2.getY())
     {
          Log.d("Gesture ", " Scroll Up");
     }
      return true;
}
```

Touch programming

```
public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)
{
    if (e1.getX() < e2.getX())
    {
         Log.d("Gesture ", "Left to Right swipe: "+ e1.getX() + " - " + e2.getX());
         Log.d("Speed ", String.valueOf(velocityX) + " pixels/second");
    }

    if (e1.getX() > e2.getX())
    {
         Log.d("Gesture ", "Right to Left swipe: "+ e1.getX() + " - " + e2.getX());
         Log.d("Speed ", String.valueOf(velocityX) + " pixels/second");
    }

    if (e1.getY() < e2.getY())
    {
         Log.d("Gesture ", "Up to Down swipe: " + e1.getX() + " - " + e2.getX());
         Log.d("Speed ", String.valueOf(velocityY) + " pixels/second");
    }

    if (e1.getY() > e2.getY())
    {
         Log.d("Gesture ", "Down to Up swipe: " + e1.getX() + " - " + e2.getX());
         Log.d("Speed ", String.valueOf(velocityY) + " pixels/second"); } return true;
    }
    return true;
}
```

Touch programming

Example1



Touch programming

Pinch (คีบ) gesture

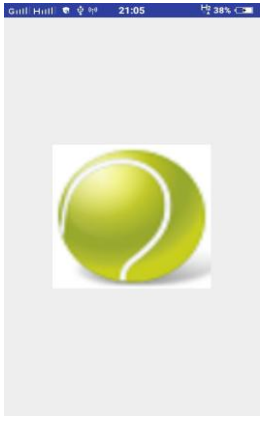
- To scale UI element
- Use ScaleGestureDetector class

Touch programming

```
public boolean onScale(ScaleGestureDetector detector)
{
    return true;
}

public boolean onScaleBegin(ScaleGestureDetector detector)
{
    return true;
}

public void onScaleEnd(ScaleGestureDetector detector)
{
    super.onScaleEnd(detector);
}
```

Touch programming

Example2



Motion Event

To detect the touch of three fingers
```
getPointerCount();
```

ACTION_DOWN
For the first pointer that touches the screen. New touch started.

ACTION_MOVE
A change has happened in the touch gesture. Finger is moving.

ACTION_UP
The last pointer leaves the screen.

ACTION_POINTER_DOWN
For extra pointers that enter the screen beyond the first. (multi-touch)

ACTION_POINTER_UP
Sent when a non-primary pointer goes up. Pointer up (multi-touch)

ACTION_CANCEL
The touch event has been canceled, something else took control of the event.

Motion Event

Example3