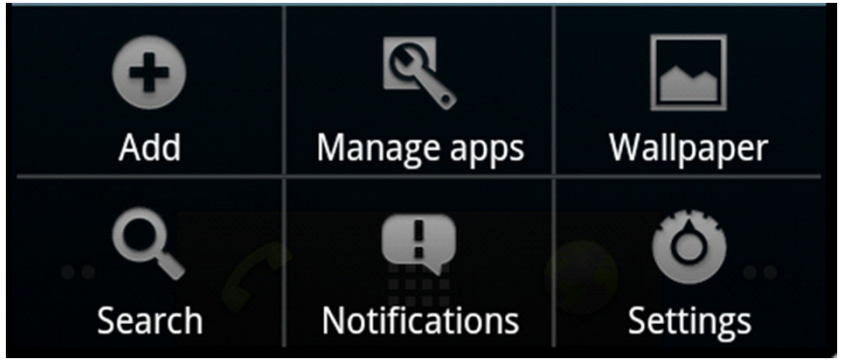
CS378 - Mobile Computing

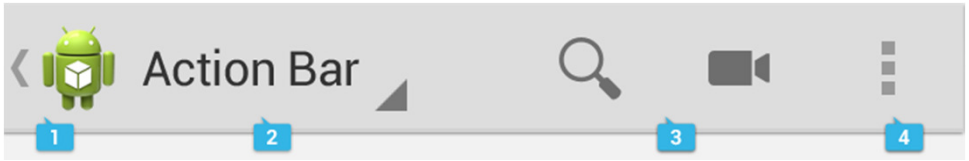
More UI - Part 2

Special Menus

- Two special application menus
  - options menu
  - context menu
- Options menu replaced by action bar (API 11)



menu



action bar

OptionsMenu

- User presses Menu Button
- Activities onCreateOptionsMenu method is called

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);

    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.options_menu, menu);
    return true;
}
```

- In example options_menu.xml in res/menu folder

OptionsMenu

- Alternate creation of OptionsMenu
- add item to menu programmatically

```
menu.add("Big About")
        .setIcon(R.drawable.about)
        .setIntent(new Intent(this, AboutActivity.class));
```

- chained method calls

SubMenus

- Option on Menu may be creation of a SubMenu
- In XML nest menu inside menu or programmatically by adding SubMenus to Menu in onCreateOptionsMenu method

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/file"
          android:title="@string/file" >
        <!-- "file" submenu -->
        <menu>
            <item android:id="@+id/create_new"
                  android:title="@string/create_new" />
            <item android:id="@+id/open"
                  android:title="@string/open" />
        </menu>
    </item>
</menu>
```

Menu Options Selected

- if Menu Option is another Activity it is launched when Menu button pressed
  - The Big About in previous example
- For other items
  - onOptionsItemSelected(MenuItem item)

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    super.onOptionsItemSelected(item);
    switch (item.getItemId()) {
    case R.id.new_game:
        startNewGame();
        return true;
    case R.id.ai_difficulty:
        showDialog(DIALOG_DIFFICULTY_ID);
        return true;
```

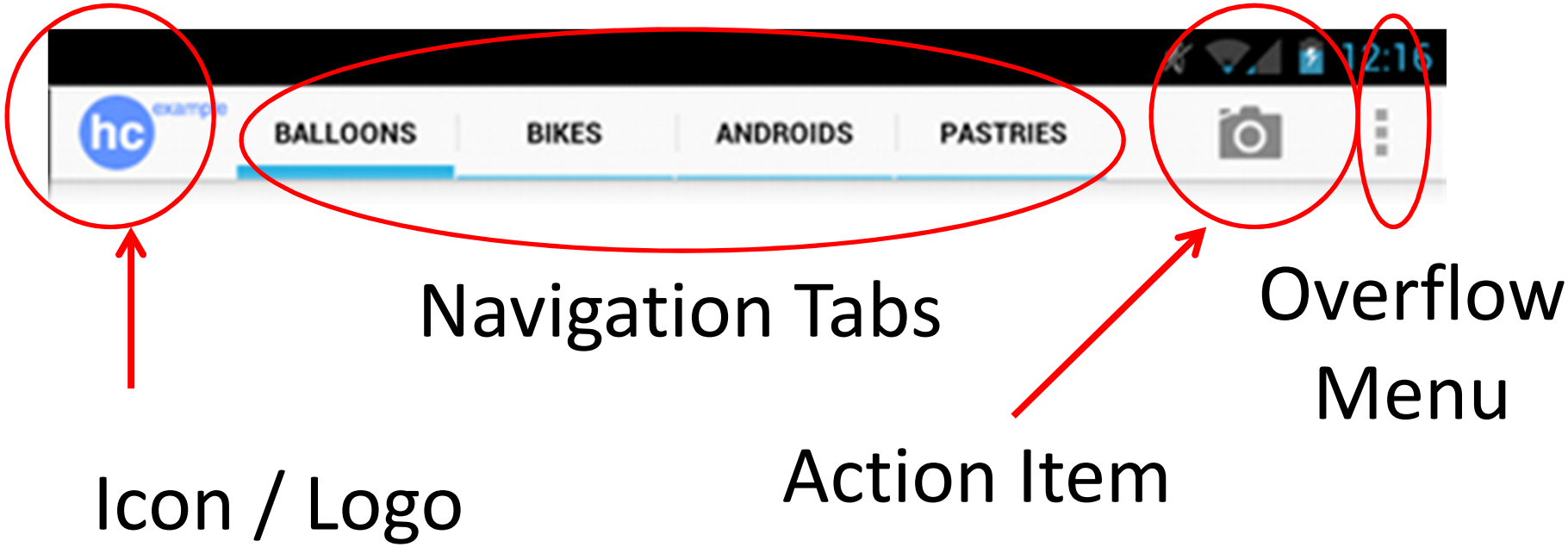
ACTION BAR

ActionBar

- Introduced in Android 3.0
  - Honeycomb, tablet only
- 4.0, Ice Cream Sandwich, tablet and phones
- "The action bar is a window feature that identifies the application and user location, and provides user actions and navigation modes"
- http://developer.android.com/guide/topics/ui/actionbar.html

Purpose of ActionBar

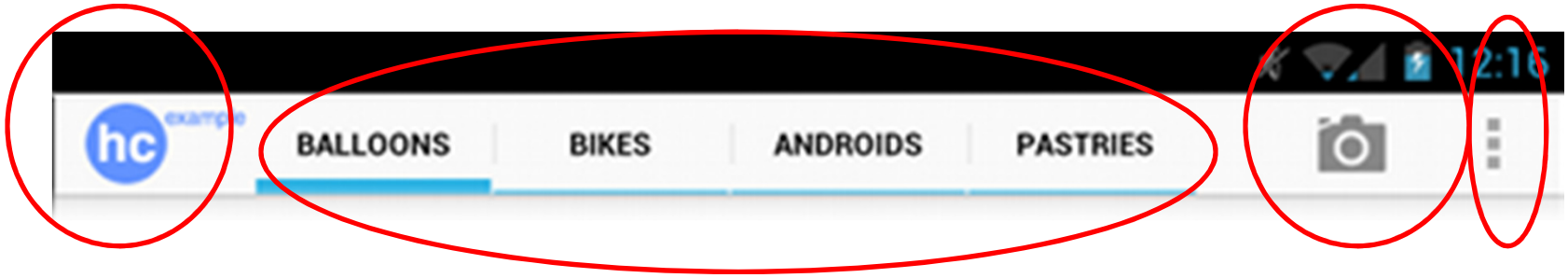
- identification
- navigation
- actions



ActionBar

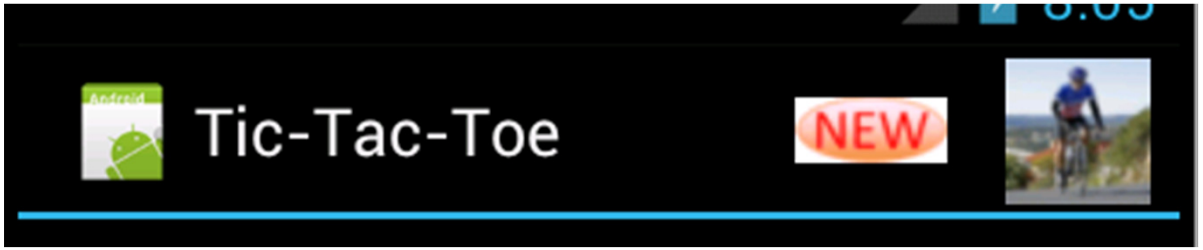
- ActionBar items declared in menu.xml

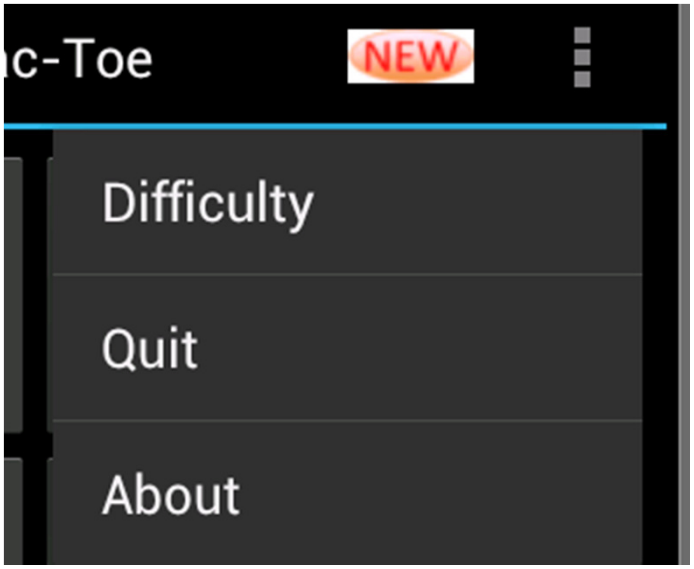
```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/new_game"
        android:icon="@drawable/new_game"
        android:title="New Game"
        android:showAsAction="ifRoom|withText" />
```



ActionBar

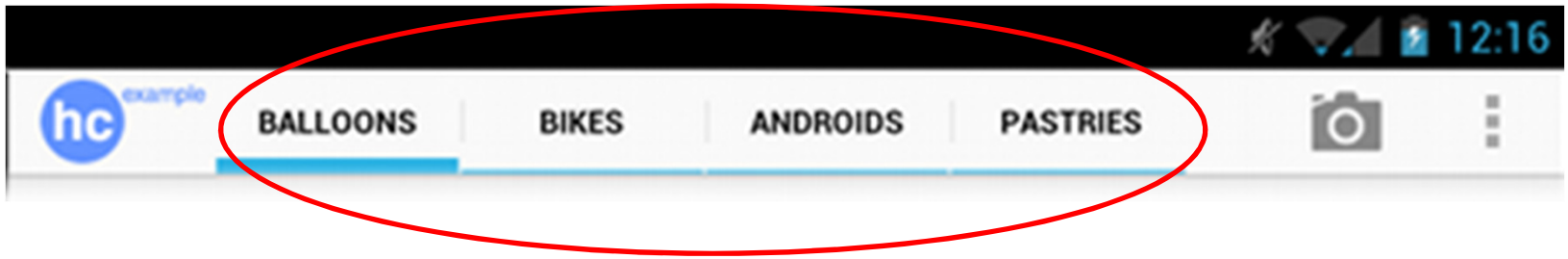
- If menu items declared in xml, added to menu in order they appear
- Extra items brought up with menu button





Navigation Tabs

- Used to switch between fragments



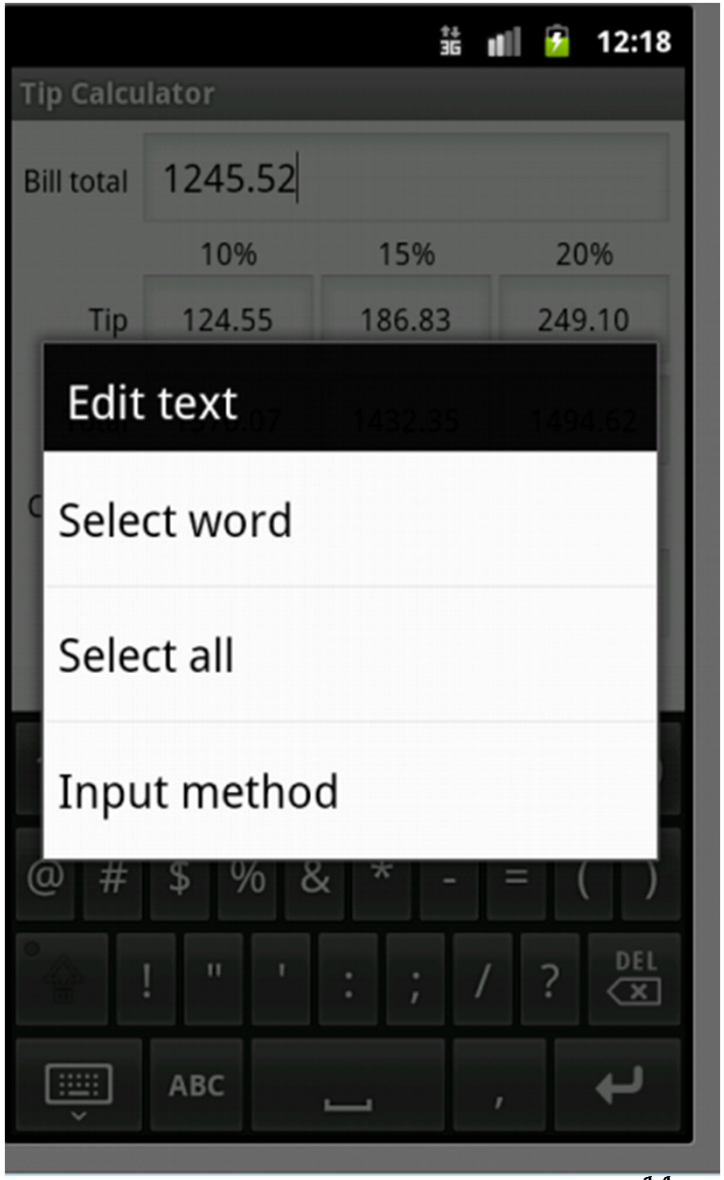
- http://developer.android.com/guide/topics/fundamentals/fragments.html

ContextMenu

- pre 3.0, aka Floating Menus
- subtype of Menu
- display when a long press is performed on a View
  - Activity is a descendant of View
  - Activity may be broken up into multiple views
- implement `onCreateContextMenu` method
- must call `registerForContextMenu` method and pass View

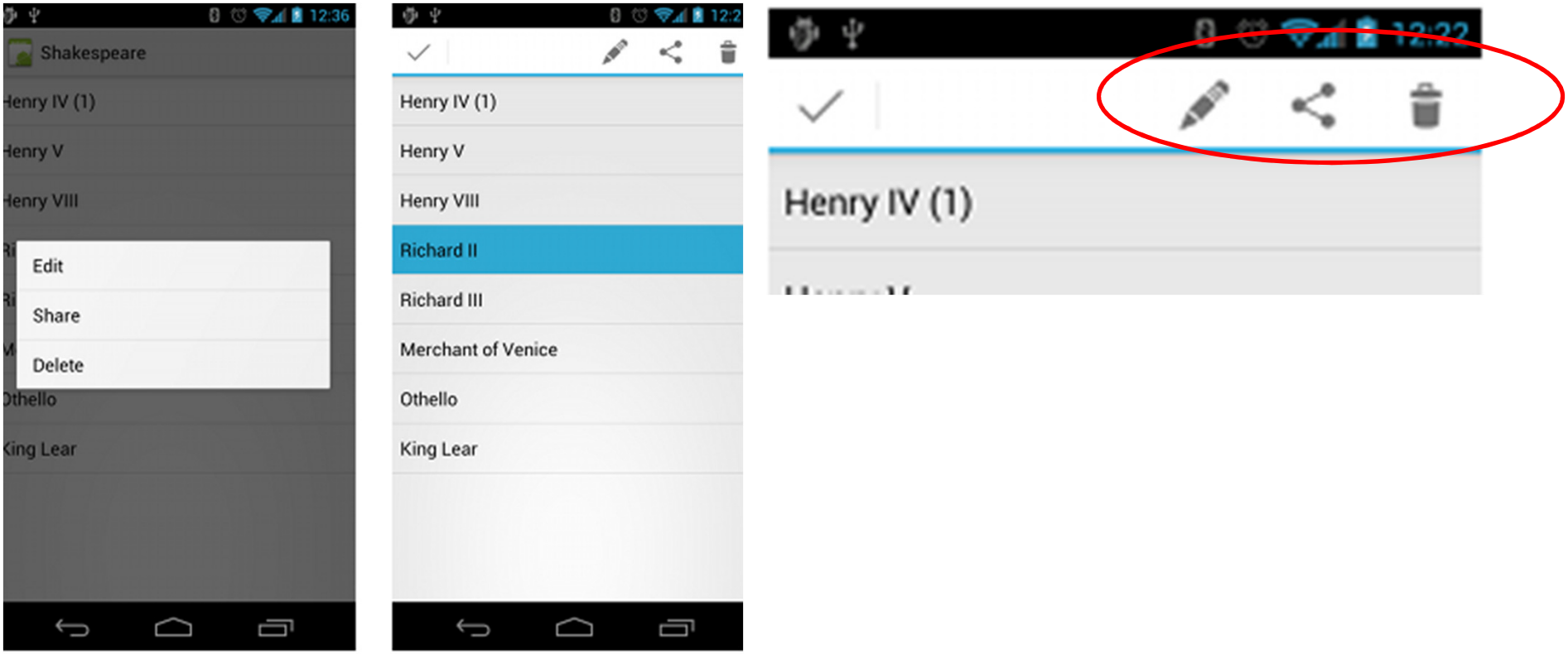
ContextMenu

- From Tip Calculator
- Long press on total amount EditText
- Default behavior for EditText
- Nothing added in TipCalculator to create this



Contextual Action Mode

- 3.0 and later



http://developer.android.com/guide/topics/ui/menus.html#CAB

STYLES

Styles

- Defined in XML file
- res/values/style
- similar to a cascading style sheet as used in html
- group layout attributes in a style and apply to various View objects (TextView, EditText, Button)

Sample Styles, in styles.xml

```
<style name="sample1">
    <item name="android:textSize">20pt</item>
    <item name="android:textColor">@color/Orange</item>
    <item name="android:textStyle">bold</item>
    <item name="android:gravity">center</item>
    <item name="android:padding">10dp</item>
</style>

<style name="sample2">
    <item name="android:textSize">8pt</item>
    <item name="android:textColor">@color/AliceBlue</item>
    <item name="android:textStyle">italic</item>
    <item name="android:gravity">right</item>
    <item name="android:padding">2dp</item>
</style>
```

Apply Style - in main xml

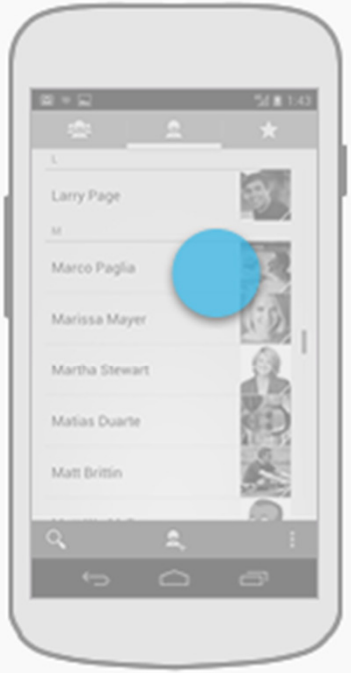
```
<TextView
    android:id="@+id/textView1"
    style="@style/sample2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="field number 1" />

<EditText
    android:id="@+id/editText1"
    style="@style/sample1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textCapWords"
    android:text="First Edit Text" />

<TextView
    android:id="@+id/textView2"
    style="@style/sample2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="field number 2" />
```

Result of Styles

- can override elements of style
  - bottom edit text overrides color
- one style can inherit from another
- use UI editor to create view and then extract to style

GESTURES

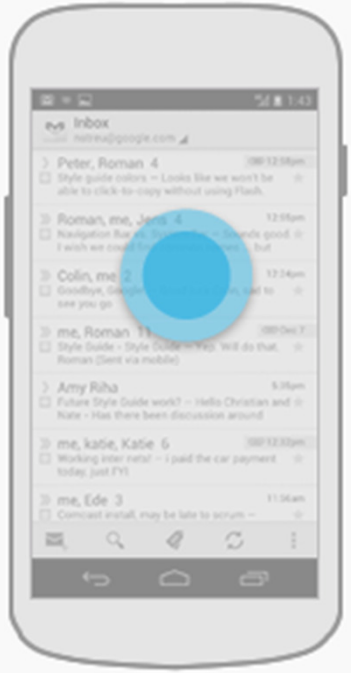
Common Gestures



Touch

Triggers the default functionality for a given item.

Action
Press, lift



Long press

Enters data selection mode. Allows you to select one or more items in a view and act upon the data using a contextual action bar. Avoid using long press for showing contextual menus.

Action
Press, wait, lift

Swipe

Scrolls overflowing content, or navigates between views in the same hierarchy.

Action
Press, move, lift

Common Gestures

Drag

Rearranges data within a view, or moves data into a container (e.g. folders on Home Screen).

Action
Long press, move, lift

Double touch

Zooms into content. Also used as a secondary gesture for text selection.

Action
Two touches in quick succession

Pinch open

Zooms into content.

Action
2-finger press, move outwards, lift

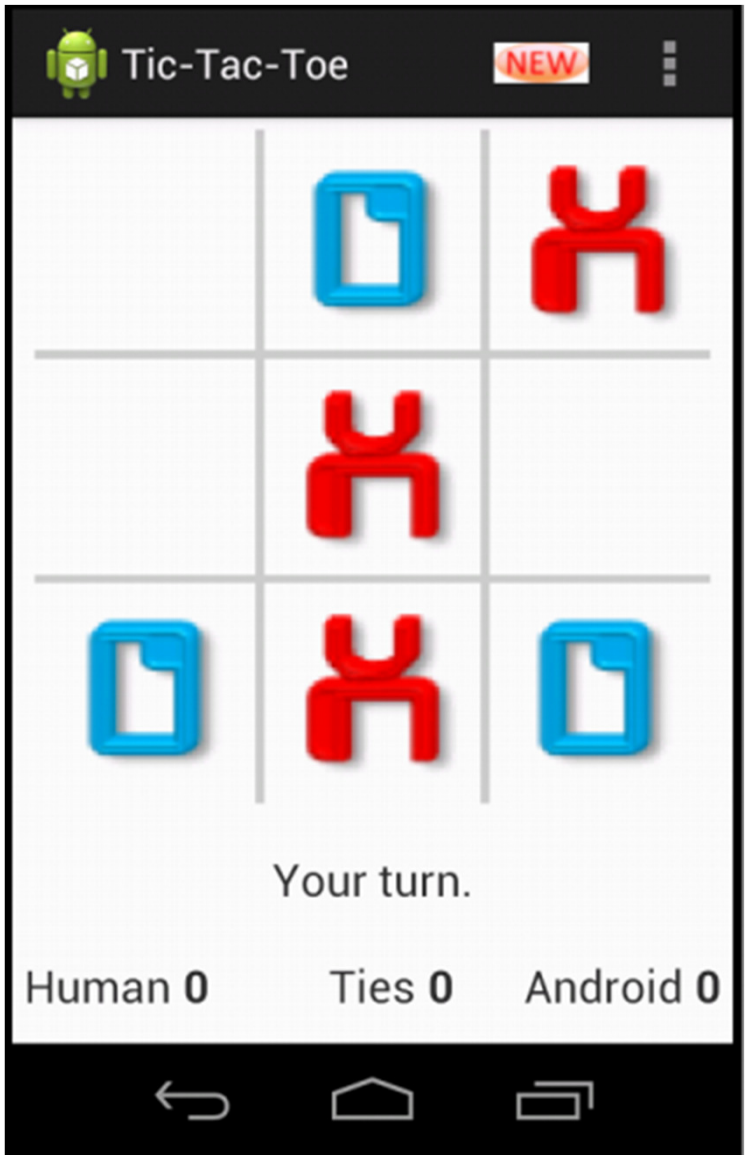
http://developer.android.com/design/patterns/gestures.html

Common Gestures

- Fling or flick gesture: similar to swipe or drag
- scroll/swipe/drag
  - user presses then moves finger in s steady motion before lifting finger
- fling or flick
  - user presses then moves finger in an accelerating motion before lifting

Dealing With Gestures

- To handle simple touch events create View.OnTouchListener for view
- Example from tutorial, screen press leads to player moving if it is their turn and they touch an open square



onTouchEvent

- passed a MotionEvent object with a large amount of data
- in tic tac toe tutorial you only used location of event (x and y)

final float	getHistoricalOrientation (int pos) `getHistoricalOrientation(int, int)` for the first pointer i
final void	getHistoricalPointerCoords (int pointerIndex, int pos, MotionEvent Populates a `MotionEvent.PointerCoords` object with historic
final float	getHistoricalPressure (int pos) `getHistoricalPressure(int, int)` for the first pointer inde
final float	getHistoricalPressure (int pointerIndex, int pos) Returns a historical pressure coordinate, as per `getPressure(i`
final float	getHistoricalSize (int pos) `getHistoricalSize(int, int)` for the first pointer index (ma
final float	getHistoricalSize (int pointerIndex, int pos) Returns a historical size coordinate, as per `getSize(int)`, that
final float	getHistoricalToolMajor (int pointerIndex, int pos) Returns a historical tool major axis coordinate, as per `getToolM`
final float	getHistoricalToolMajor (int pos) `getHistoricalToolMajor(int, int)` for the first pointer ind
final float	getHistoricalToolMinor (int pointerIndex, int pos) Returns a historical tool minor axis coordinate, as per `getToolM`
final float	getHistoricalToolMinor (int pos) `getHistoricalToolMinor(int, int)` for the first pointer ind
final float	getHistoricalTouchMajor (int pointerIndex, int pos) Returns a historical touch major axis coordinate, as per `getTouc`
final float	getHistoricalTouchMajor (int pos) `getHistoricalTouchMajor(int, int)` for the first pointer in
final float	getHistoricalTouchMinor (int pointerIndex, int pos)

Handling Common Gestures

- Instead of trying to decode gestures from the MotionEvent passed to on touch …
- Use the GestureDetector class
- Add a GestureDetector object to View
- override View.onTouhcEvent method to pass MotionEvent on to the GestureDetector.onTouchEvent method

Handling Common Gestures

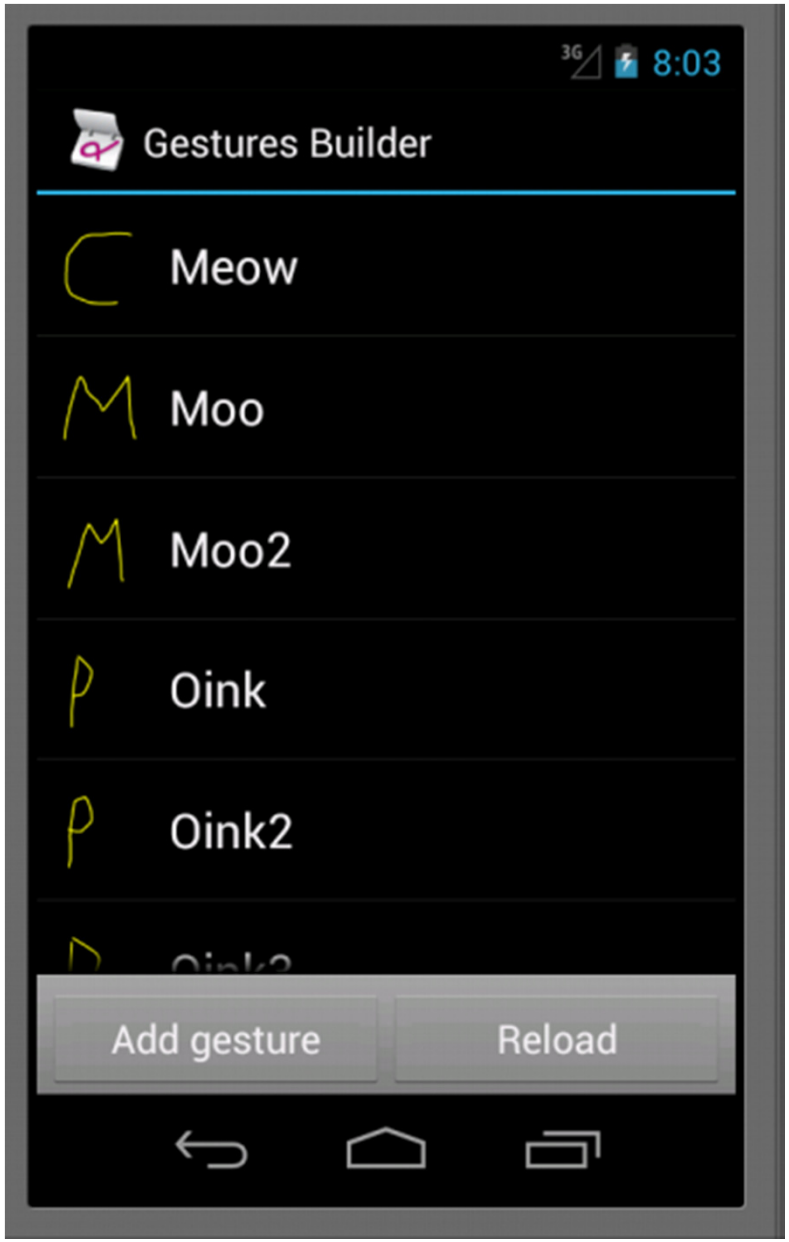
- create a GestureDetector.OnGestureListener (seveal gestures) or a GestureDetector.SimpleOnGestureListener (more gestures) and register it with the GesturerDetector
- callback methods for onDoubleTap, onLongPress, onScroll, onFling, onSingleTapConfirmed, many more

Complex Gestures

- Non standard gestures required lots of code to recognize
- Android 1.6 introduced new APIs to store, load, draw, and recognize gestures
- Gesture Builder app on emulator
  - emulator must include virtual SD card
  - allows creating set of gestures for your ppliaction

Complex Gestures

- Each gesture associated with name
- multiple gestures can have same name
  - variations on same gesture, better chance of recognizing
- Move gestures from emulator to application res/raw folder



Complex Gestures

- Recognizing gestures via a GestureOverlayView
- simple drawing board on top of view that shows and records user gestures
- When gesture complete GestureLibrary queried to see if gesture is recognized
- Predictions between entered gesture and those in the library

Animal Sounds App

Predictions

```
AnimalSounds      prediction score: 5.02052297579021, name: Oink2
AnimalSounds      prediction score: 11.698475110815773, name: Meow
AnimalSounds      prediction score: 1.4253241939996129, name: Oink3
AnimalSounds      prediction score: 1.708742452226205, name: Oink
AnimalSounds      prediction score: 1.778813340981308 7, name: Oink
Choreographer     Skipped 30 frames!  The application may be doing
AnimalSounds      prediction score: 1.597973912890255 3, name: Moo2
AnimalSounds      prediction score: 1.131260158503845 5, name: Moo
AnimalSounds      prediction score: 1.733056893468628, name: Meow
AnimalSounds      prediction score: 0.740482776019489 1, name: Moo
AnimalSounds      prediction score: 1.009555907026495 7, name: Moo2
AnimalSounds      prediction score: 1.408645869375701, name: Moo2
AnimalSounds      prediction score: 2.048106505538496, name: Oink3
AnimalSounds      prediction score: 3.078060118728627, name: Meow
AnimalSounds      prediction score: 2.932816689691991, name: Meow
AnimalSounds      prediction score: 1.792527999275177, name: Meow
AnimalSounds      prediction score: 1.816917660586996 6, name: Oink3
AnimalSounds      prediction score: 0.714336637312408 7, name: Moo
AnimalSounds      prediction score: 1.523282119075419 5, name: Oink
Choreographer     Skipped 32 frames!  The application may be doing
AnimalSounds      prediction score: 0.785716727687679 1, name: Moo
```

onCreate

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    mLibrary = GestureLibraries.fromRawResource(this, R.raw.gestures);
    if (!mLibrary.load()) {
        finish();
    }

    GestureOverlayView gestures
            = (GestureOverlayView) findViewById(R.id.gestures);
    gestures.addOnGesturePerformedListener(mGestureListener);

    createSoundPool();
}
```

Listener

```
@Override
public void onGesturePerformed(GestureOverlayView overlay,
        Gesture gesture) {
    // from http://android-developers.blogspot.com/2009/10/gestures-on-android-

    ArrayList<Prediction> predictions = mLibrary.recognize(gesture);

    // We want at least one prediction
    if (predictions.size() > 0) {
        Prediction prediction = predictions.get(0);

        Log.d(TAG, "prediction score: " + prediction.score + ", name: " + predi

        // We want at least some confidence in the result
        if (prediction.score > 3.0) {
            String name = prediction.name;
            if(name.contains("Moo"))
                mSounds.play(mSoundIDMap.get("Moo"), 1, 1, 1, 0, 1);
            else if(name.contains("Oink"))
                mSounds.play(mSoundIDMap.get("Oink"), 1, 1, 1, 0, 1);
            else if(name.contains("Meow"))
                mSounds.play(mSoundIDMap.get("Meow"), 1, 1, 1, 0, 1);
        }
    }
}
```