Getting Started with
Android Programming

The Android Community and David Read

Tech Valley Mobile Developers Network
February 1, 2012

My Background

- 25 years IT programming and design
- CTO at Blue Slate Solutions
- BPM, Data, Security, Machine Learning
- C, C++, Java, OWL, SPARQL, R
- Semantic Technology, Predictive Analytics
- SCJP, RHCE, PRPC, JRules, GSEC, CISSP
- Music Education, Piano, Voice

Agenda

- **Scope and Preconditions**
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- UI View
- Build It!
- Testing, Debugging and Uploading
- Q&A

What is “Android”

“Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.”

http://developer.android.com/guide/basics/what-is-android.html

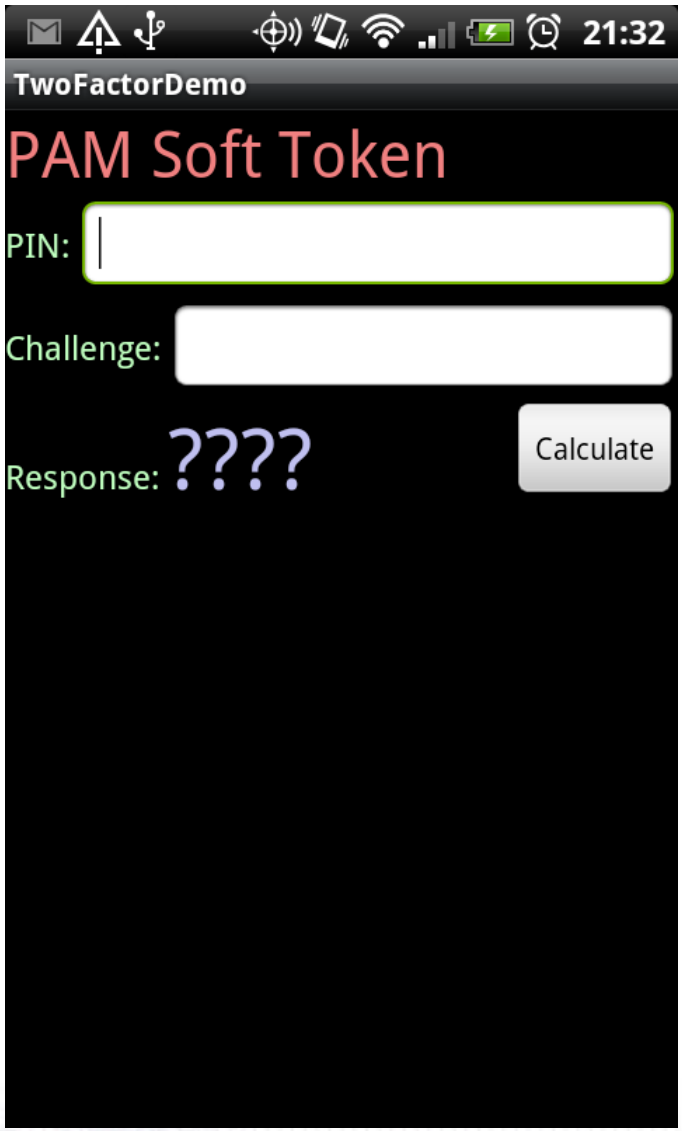
Geekic Overlook

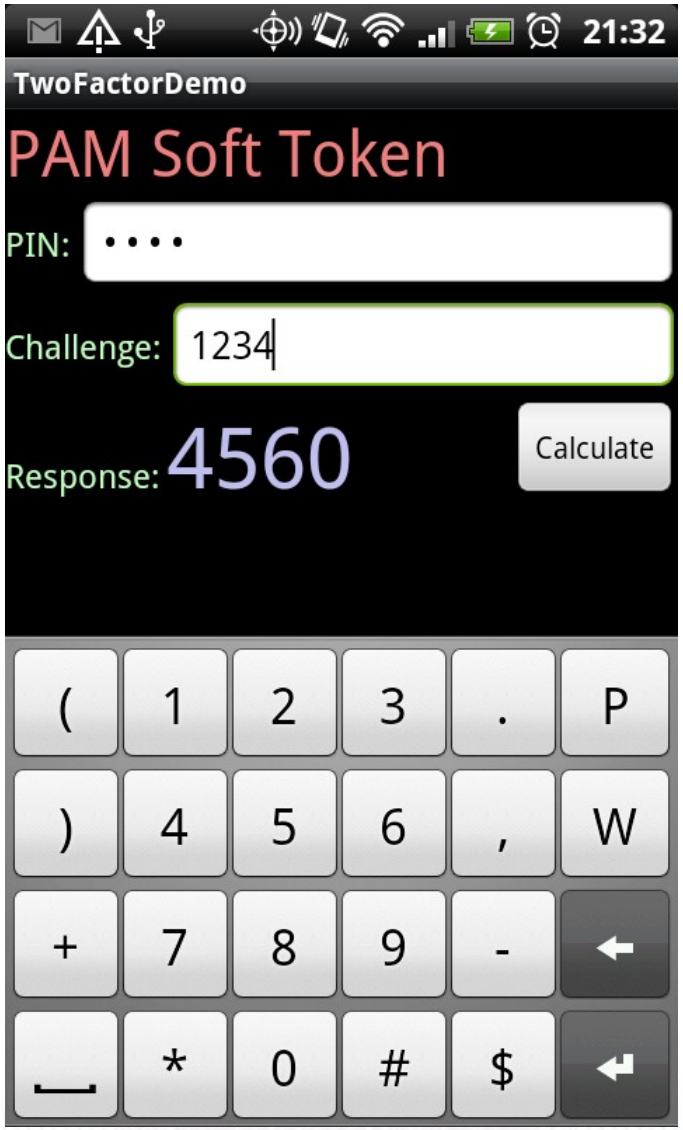
- Although the Java syntax is used and Java bytecode serves as an input for creating an Android application, the actual runtime environment is the Dalvik VM which does not support all the standard Java behaviors or libraries (like Swing)
- Geek points: Standard JVMs use a stack, the Delvik VM is register-based

Scope

- Android **Activity**
  - Usually programs for human interaction
  - Separate from **services** and **broadcast receivers**
- Introductory
  - 45 minutes - we won't get to see much beyond initial setup and methods
- Interactive
  - Ask questions, participate

The Sample Application





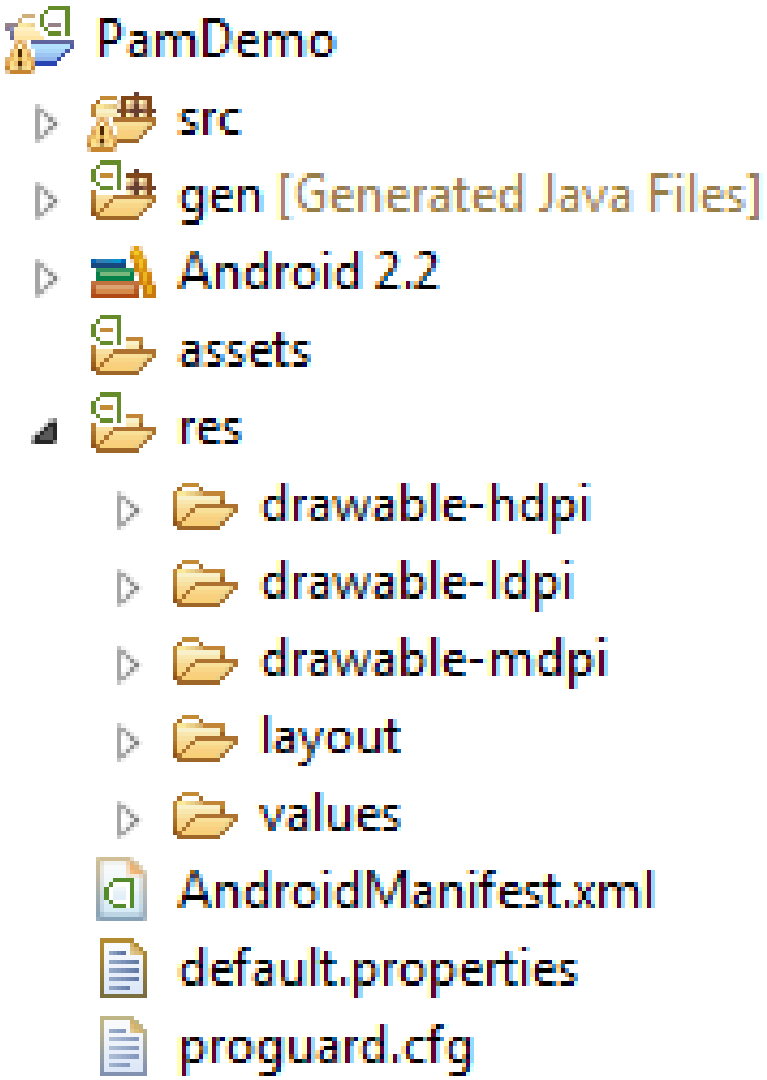
Preconditions

- Install Java 6 or 7
- Install Android SDK
  - latest is 4.0.3 as of January, 2012
- Unless you abhor productivity improvements, install Eclipse and the ADT plug-in
- Learn Java

Agenda

- Scope and Preconditions
- **Project: Directories and Files**
- Activity Lifecycle
- Lifecycle Methods
- UI View
- Build It!
- Testing, Debugging and Uploading
- Q&A

Project: Directory and Files



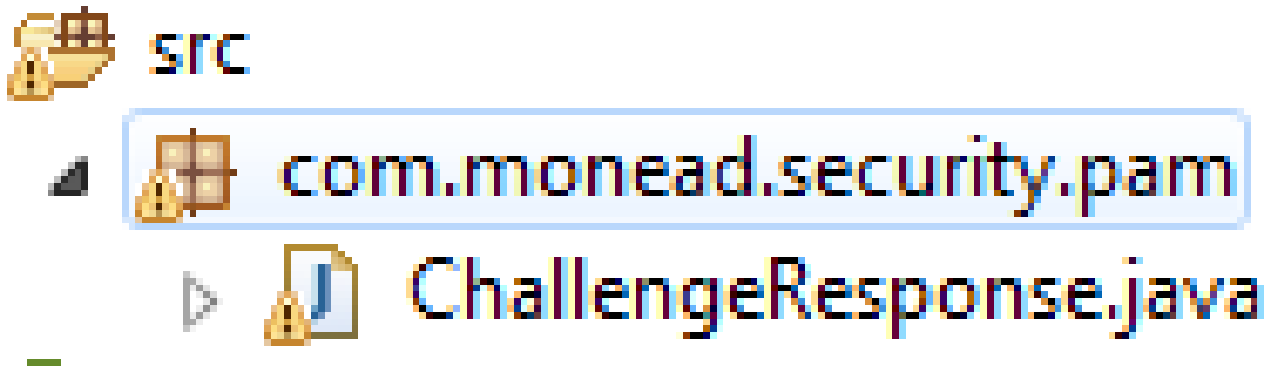
File: AndroidManifest.xml

- Configuration for the application
- Versioning, icon, initialization

```
<manifest package="com.monead.security.pam"
    android:versionCode="1" android:versionName="1.0">
  <application android:icon="@drawable/icon"
      android:label="@string/app_name">
    <activity android:name=".ChallengeResponse"
        android:label="@string/app_name">
      <intent-filter>...</intent-filter>
    </activity>
  </application>
  <uses-sdk android:targetSdkVersion="8"
      android:minSdkVersion="4"/>
    <uses-permission
        android:name=
          "android.permission.READ_PHONE_STATE" />
</manifest>
```

Directory: src

- Application source code
- Typical Java package structure

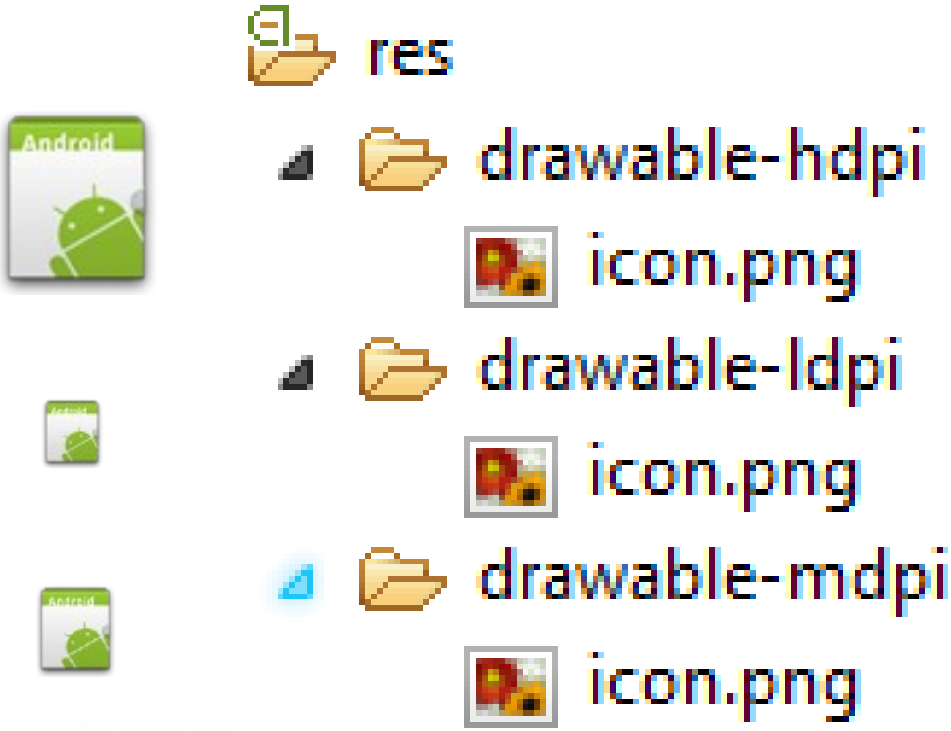


Directory: res/drawable-*

- Image files
- 3 directories to deal with various screen resolutions
- ldpi->low, mdpi->medium, hdpi->high
- Syntax to load an image allows the environment to determine the correct directory (ldpi, mdpi, hdpi)
- Image file name must match in each directory so that it can be found

Example of Launcher Icon File

- 3 resolutions for icon (measured in pixels):
- ldpi->36x36, mdpi->48x48, hdpi->72x72

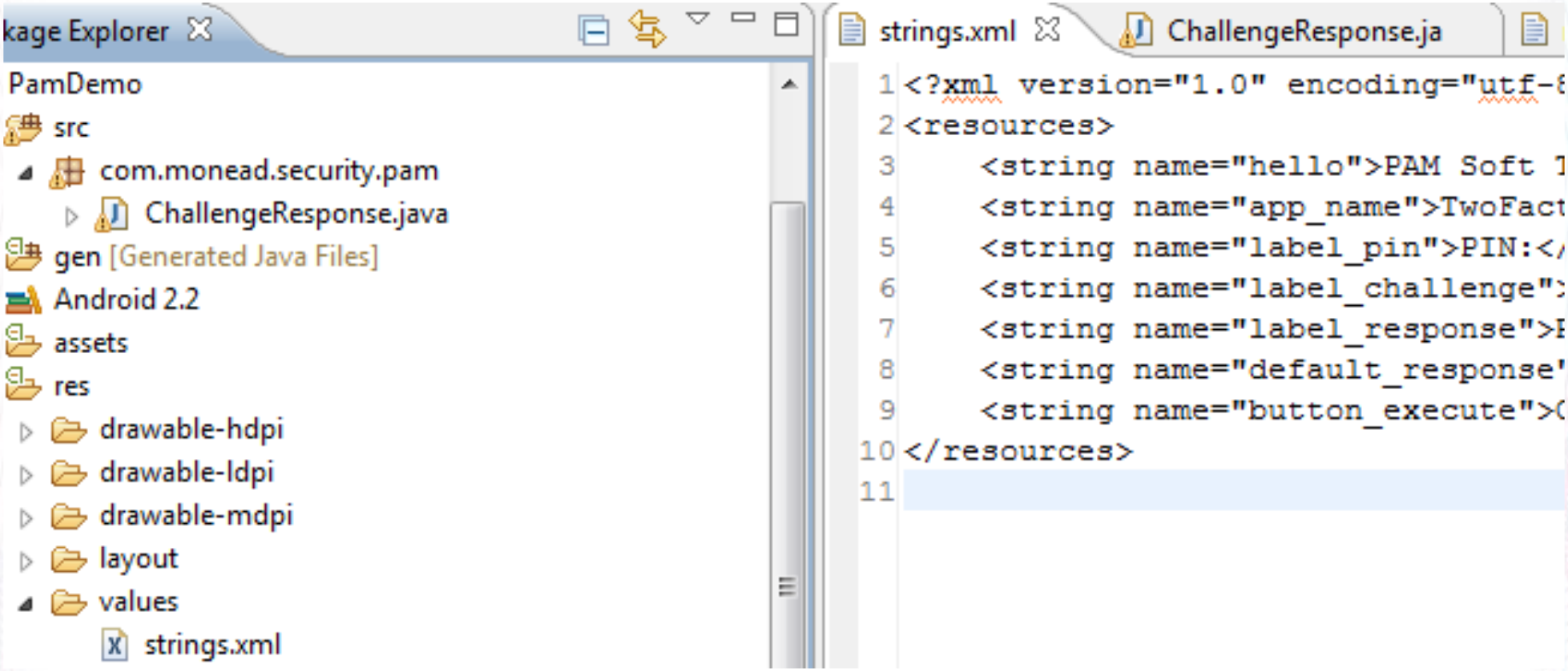


Directory: res/layout

- UI definitions
  - Layouts
  - Widgets

Directory: res/values

- Constants, supports internationalization



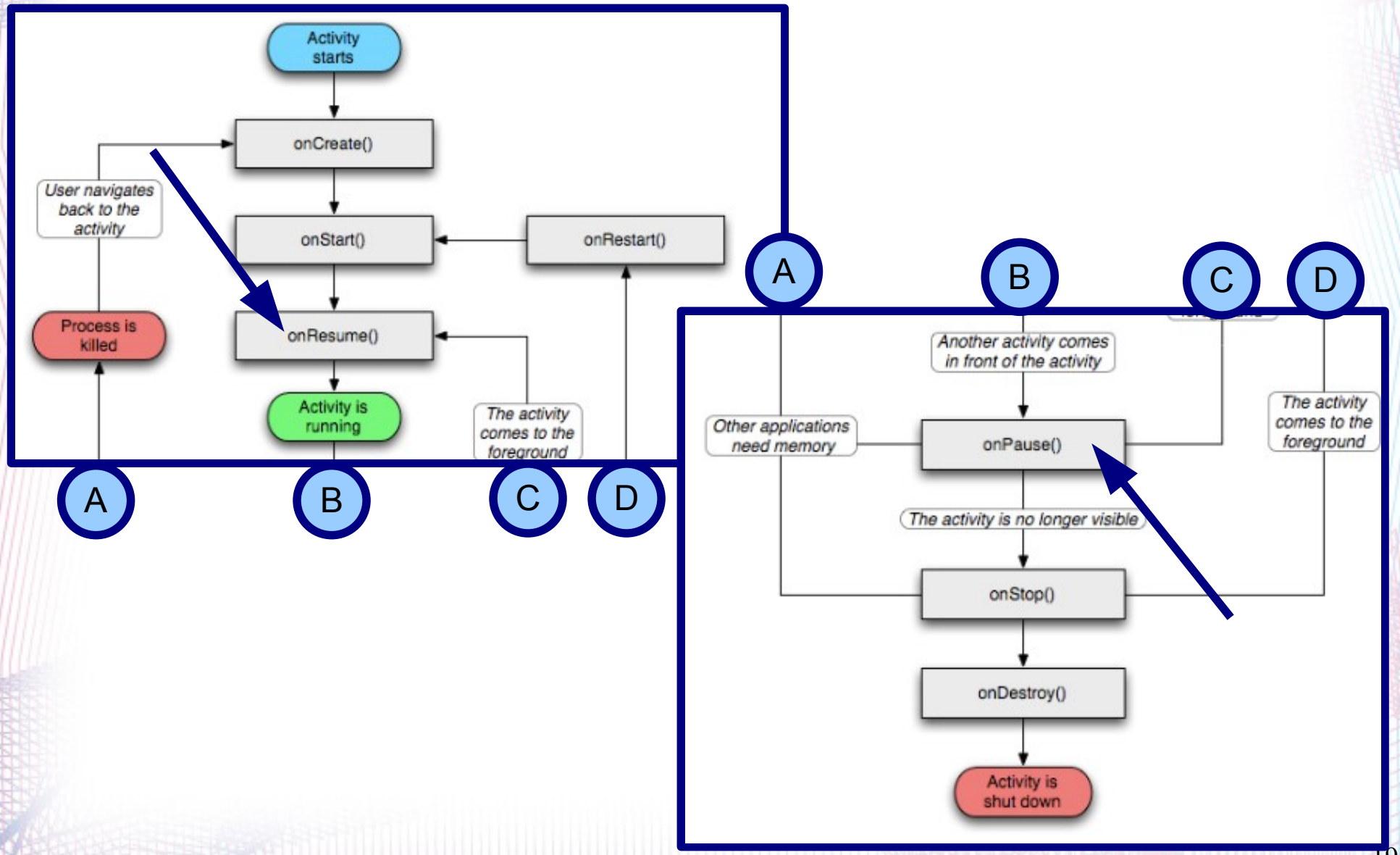
Agenda

- Scope and Preconditions
- Project: Directories and Files
- **Activity Lifecycle**
- Lifecycle Methods
- UI View
- Build It!
- Testing, Debugging and Uploading
- Q&A

Activity Lifecycle

- Guaranteed set of states through which the activity will pass from initial launch to termination
- Programmatically represented with methods that the programmer overrides to control the activity when it is in a specific state
- An activity implementation is a class that extends **android.app.Activity**

Activity Lifecycle Depiction



Agenda

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- **Lifecycle Methods**
- UI View
- Build It!
- Testing, Debugging and Uploading
- Q&A

Activity Lifecycle Methods

- **void onCreate(Bundle savedInstanceState)**
- void onStart()
- void onRestart()
- **void onResume()**
- **void onPause()**
- void onStop()
- void onDestroy()

onCreate(Bundle savedState)

```
/** Called when the activity
    is created */
@Override
public void onCreate(
    Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.main);

  setup();
}
```

onPause()

```
/** Called when activity loses
    foreground control */
public void onPause() {
   super.onPause();

   clearValues();
}
```

onResume()

```
/**
   * Called when activity
   * regains control
   */
public void onResume() {
   super.onResume();

   setForInput();
}
```

Agenda

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- **UI View**
- Build It!
- Testing, Debugging and Uploading
- Q&A

UI View

- Preferred approach to define the UI is to use XML files located in the **res/layout** directory
- Using the Eclipse plugin for Android development simplifies the development cycle since there is a build step required to turn the UI definitions into code so that you may programmatically access the widgets defined in the XML file

Part of Our Layout

```
<LinearLayout android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
  <TextView … />
  <LinearLayout android:orientation="horizontal"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content">
    <TextView android:id="@+id/label_pin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/label_pin"/>
    <EditText android:id="@+id/pin"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:password="true"
        android:inputType="phone"
        android:singleLine="true" />
  </LinearLayout>
</LinearLayout>
```

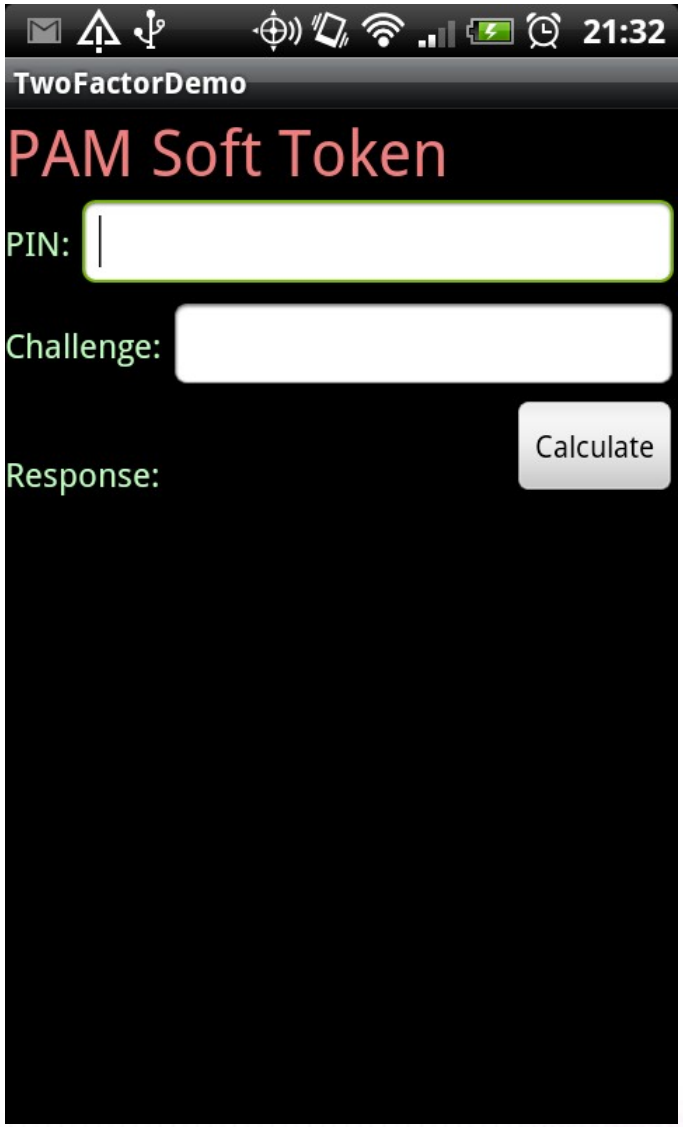
Code Using the Prior UI *(partial)*

```
private void clearValues() {
   EditText pin;
   EditText challenge;
   TextView response;

   pin = (EditText) findViewById(R.id.pin);
   challenge = (EditText)
       findViewById(R.id.challenge);
   response = (TextView)
       findViewById(R.id.response);

   pin.setText("");
   challenge.setText("");
   response.setText("");
}
```

Resulting Screen



Internationalization

- Define strings in a resource file
- Directory determines language
  - res/values : default
  - res/values-*ISO-LANG-CODE*
  - res/values-*ISO-LANG-CODE_COUNTRY*
- Examples
  - res/values-es/strings.xml
  - res/values-es_MX/strings.xml

res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">PAM Soft Token</string>
  <string name=
      "app_name">TwoFactorDemo</string>
  <string name="label_pin">PIN:</string>
  <string name=
      "label_challenge">Challenge:</string>
  <string name=
      "label_response">Response:</string>
  <string name=
      "default_response">\?\?\?\?</string>
</resources>
```

res/values-es/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name=
      "hello">PAM testigo lógico</string>
  <string name=
      "app_name">DosFactoresDeDemo</string>
  <string name="label_pin">NIP:</string>
  <string name=
      "label_challenge">Desafío:</string>
  <string name=
      "label_response">Respuesta:</string>
  <string name=
      "default_response">\?\?\?\?</string>
</resources>
```

Get a Value – In A View

```
<LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
  <TextView android:id="@+id/label_challenge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#c0ffc0"
    android:textSize="16sp"
    android:text="@string/label_challenge"/>
```

Get a Value - Programatically

```
catch (Throwable throwable) {
  Log.e("ChallengeResponse",
    "Failed to calculate response",
    throwable);

  response.setText(
      getResources().getString(
      R.string.default_response));
}
```

Agenda

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- UI View
- **Build It!**
- Testing, Debugging and Uploading
- Q&A

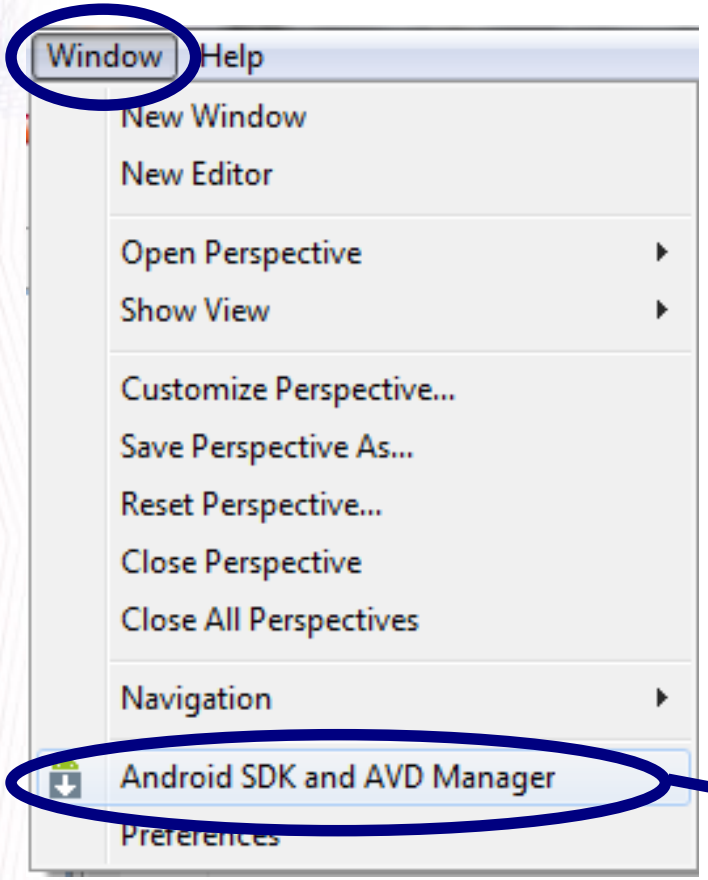
Agenda

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- UI View
- Build It!
- **Testing, Debugging and Uploading**
- Q&A

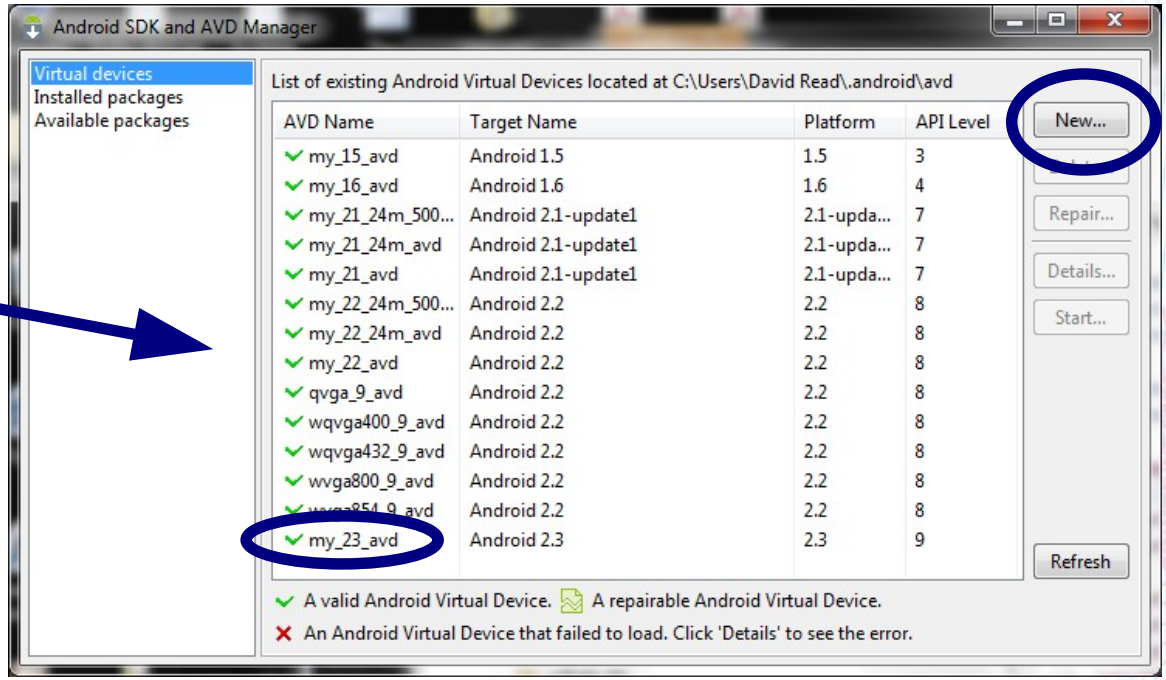
Testing

- The Android SDK provides emulators that can mimic a variety of Android versions, screen dimensions and phone behaviors
- Runs slower than on a real phone
  - So don't get too worried if your application feels sluggish in the emulator
- Provides support for all the features and lifecycle events including persisting state, writing local files, etc
- From Eclipse it launches automatically when you run the application

Create an Emulator



- Must be uniquely named
- Each may have different configurations
  - (e.g. Android version, memory, SD card)

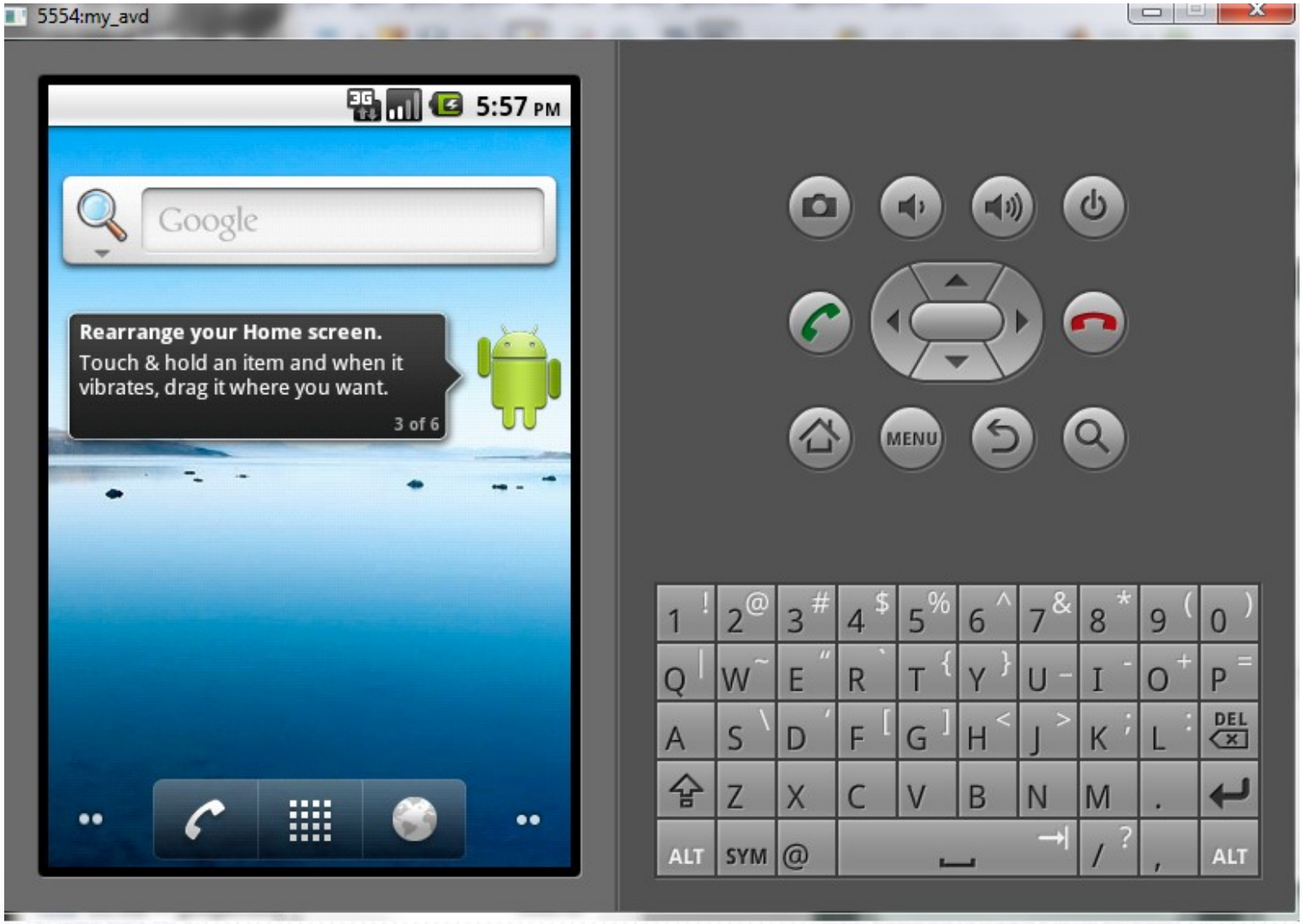


Start Emulator with Options

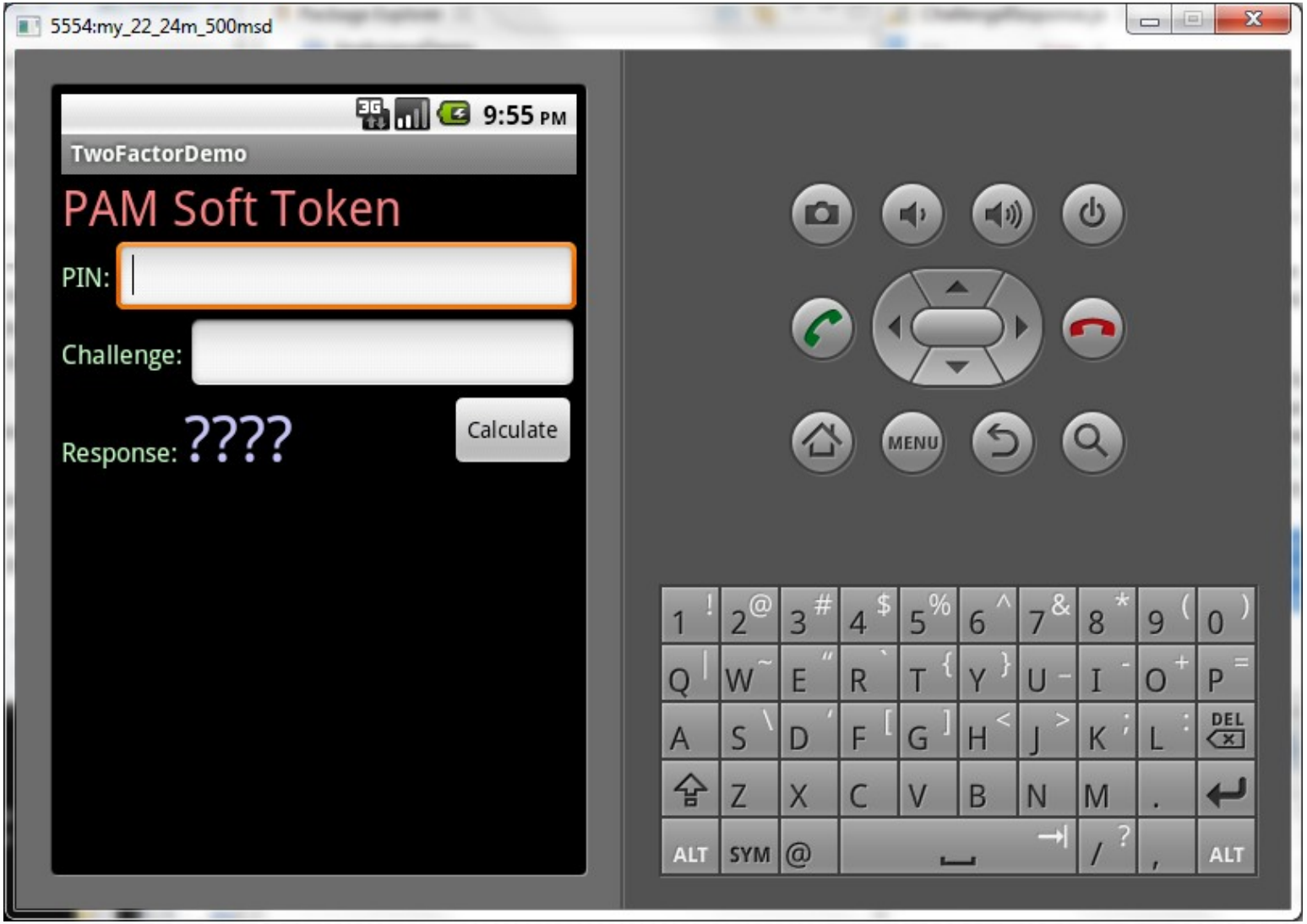
- Emulator Named: **my_23_avd**
- Options: Spanish/Spain

```
start "Droid 2.3 HVGA es ES" /B /D \usr\android-sdk-windows\tools emulator -avd my_23_avd -prop persist.sys.language=es -prop persist.sys.country=ES
```

Emulator at Startup



Emulator Running an App



Debugging

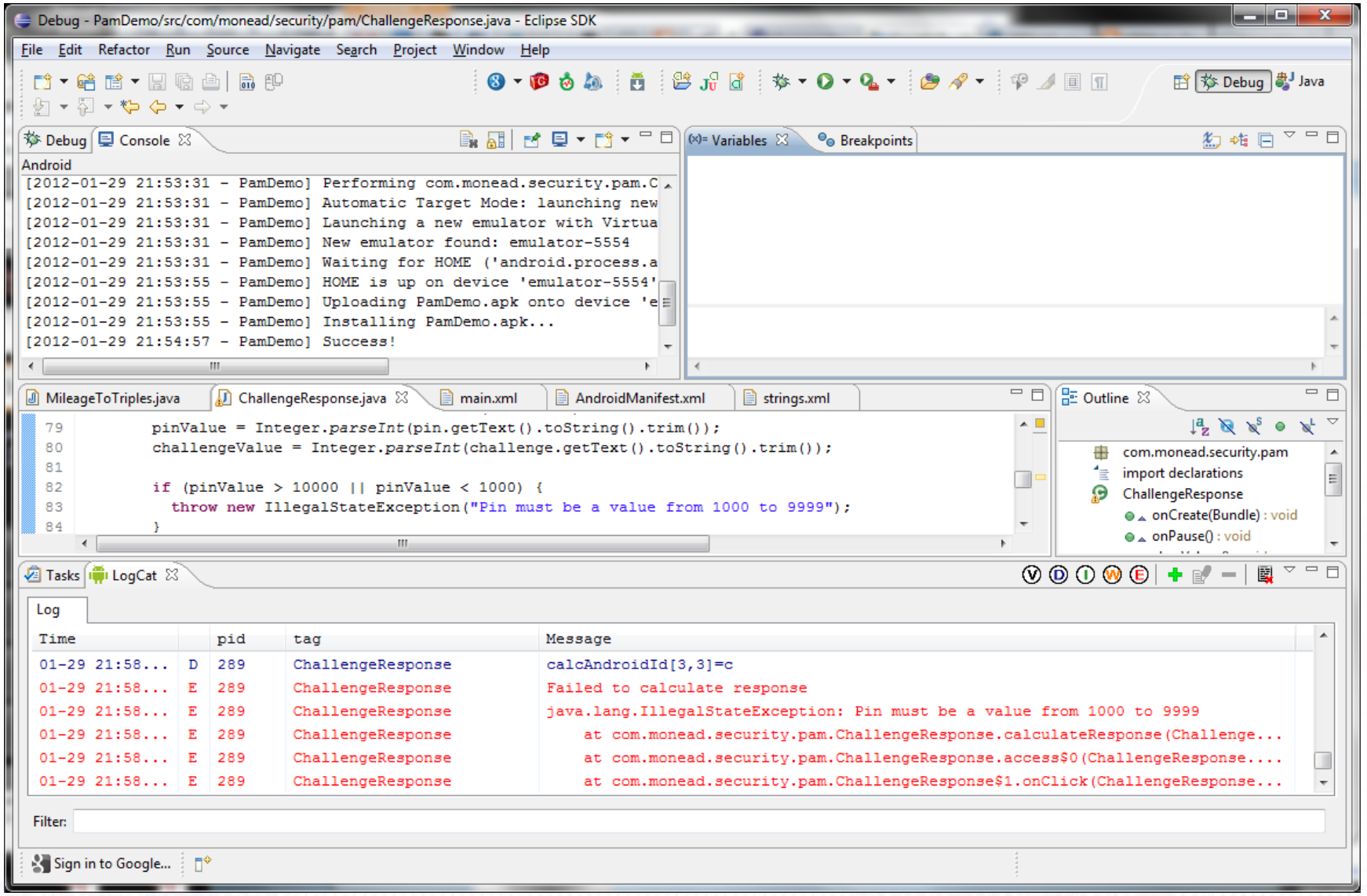
- Eclipse debug view is helpful
  - Can debug using phone or emulator
- Use of logging is essential
- Environment provides a logger accessed with the Log class' static methods:
  - v(), d(), i(), w(), e()
  - Two or three parameters: an application “tag” identifying the source of the message, a text message and an optional Throwable

Sample Code with Logging

```
catch (Throwable throwable) {
  Log.e("ChallengeResponse",
    "Failed to calculate response",
    throwable);

  response.setText(
      getResources().getString(
      R.string.default_response));
}
```

Debug View


Debug - PamDemo/src/com/monead/security/pam/ChallengeResponse.java - Eclipse SDK
File Edit Refactor Run Source Navigate Search Project Window Help
Debug Java
Debug Console
Android
[2012-01-29 21:53:31 - PamDemo] Performing com.monead.security.pam.C
[2012-01-29 21:53:31 - PamDemo] Automatic Target Mode: launching new
[2012-01-29 21:53:31 - PamDemo] Launching a new emulator with Virtua
[2012-01-29 21:53:31 - PamDemo] New emulator found: emulator-5554
[2012-01-29 21:53:31 - PamDemo] Waiting for HOME ('android.process.a
[2012-01-29 21:53:55 - PamDemo] HOME is up on device 'emulator-5554'
[2012-01-29 21:53:55 - PamDemo] Uploading PamDemo.apk onto device 'e
[2012-01-29 21:53:55 - PamDemo] Installing PamDemo.apk...
[2012-01-29 21:54:57 - PamDemo] Success!
Variables Breakpoints
MileageToTriples.java ChallengeResponse.java main.xml AndroidManifest.xml strings.xml
79 pinValue = Integer.parseInt(pin.getText().toString().trim());
80 challengeValue = Integer.parseInt(challenge.getText().toString().trim());
81
82 if (pinValue > 10000 || pinValue < 1000) {
83 throw new IllegalStateException("Pin must be a value from 1000 to 9999");
84 }
Outline
com.monead.security.pam
import declarations
ChallengeResponse
onCreate(Bundle) : void
onPause() : void
Tasks LogCat
Log
Time pid tag Message
01-29 21:58... D 289 ChallengeResponse calcAndroidId[3,3]=c
01-29 21:58... E 289 ChallengeResponse Failed to calculate response
01-29 21:58... E 289 ChallengeResponse java.lang.IllegalStateException: Pin must be a value from 1000 to 9999
01-29 21:58... E 289 ChallengeResponse at com.monead.security.pam.ChallengeResponse.calculateResponse(Challenge...
01-29 21:58... E 289 ChallengeResponse at com.monead.security.pam.ChallengeResponse.access$0(ChallengeResponse....
01-29 21:58... E 289 ChallengeResponse at com.monead.security.pam.ChallengeResponse$1.onClick(ChallengeResponse...
Filter:
Sign in to Google...

Android App Store

- Must create an Android Developer Account
  - $25 one-time fee
  - Create public/private key to sign all of your applications
- To sell your application must also create a Google Merchant Account
  - SSN/Tax Id
  - Fee schedule $0.30 + 2.9% down to 1.9% based on monthly sales $

Assets for An Application

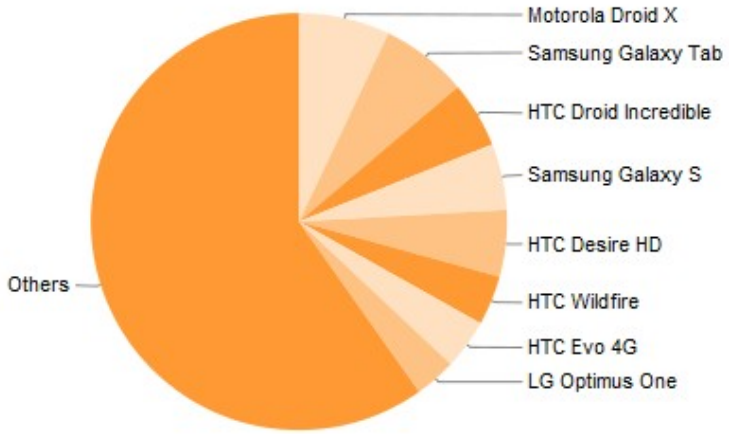
- Screenshots
- High-res Icon
- Promotional icons and videos
- Title and description
  - Internationalization supported
- Application type and category
  - e.g. Games, Brain&Puzzle

Build the Release

- Use the ant “release” task
  - Compiles, aligns (Dalvik VM, remember?) and signs (with your private key)
- Upload to the Android Market using your account
  - https://market.android.com/publish/Home
- Add release notes and publish

Variety of Market Statistics

Device



com.monead.games.android.sequence	
1 Motorola Droid X	7.1% (22)
2 Samsung Galaxy Tab	6.8% (21)
3 HTC Droid Incredible	5.1% (16)
4 Samsung Galaxy S (GT-I9000)	5.1% (16)
5 HTC Desire HD	5.1% (16)
6 HTC Wildfire	3.9% (12)
7 HTC Evo 4G	3.9% (12)
8 LG Optimus One	3.2% (10)
9 Samsung Galaxy S (SCH-I500)	2.9% (9)
10 Motorola Droid	2.6% (8)

Country

com.monead.games.android.sequence		Market all apps	
1 United States	55.6% (173)	United States	57.1%
2 United Kingdom	5.8% (18)	South Korea	9.7%
3 Canada	2.9% (9)	Japan	6.4%
4 Italy	2.9% (9)	United Kingdom	4.3%
5 Australia	2.6% (8)	France	2.7%
6 Brazil	1.9% (6)	Germany	2.5%
7 Germany	1.9% (6)	Taiwan	1.3%

Agenda

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- UI View
- Build It!
- Testing and Debugging
- **Q&A**

Remember Trick 1

```
<EditText android:id="@+id/pin"
    android:layout_width="fill_parent"
  android:layout_height="wrap_content"
      android:password="true"
      android:inputType="phone"
      android:singleLine="true" />
```

Remember Trick 2

```
new Handler()).postDelayed(new Runnable() {
  public void run() {
    EditText pin = (EditText)
        findViewById(R.id.pin);
    pin.requestFocus();
    pin.dispatchTouchEvent(MotionEvent.obtain(
        SystemClock.uptimeMillis(),
        SystemClock.uptimeMillis(),
        MotionEvent.ACTION_DOWN, 0, 0, 0));
    pin.dispatchTouchEvent(MotionEvent.obtain(
        SystemClock.uptimeMillis(),
        SystemClock.uptimeMillis(),
        MotionEvent.ACTION_UP, 0, 0, 0));
    }
  }, 200);
```

Go Forth and Code

- **Thank you for attending!**
- Download software and start developing
  - developer.android.com
- Tons of online support and articles
- Questions and pointers appreciated
- www.monead.com
- David.S.Read@gmail.com
- **Sparql Droid:** http://tinyurl.com/7fqzdk9

References

- http://www.oracle.com/technetwork/java/javase/downloads/index.html
- http://www.eclipse.org/
- http://developer.android.com/index.html
- http://developer.android.com/sdk/index.html
- http://developer.android.com/sdk/eclipse-adt.html