Android Programming

An Introduction

The Android Community and David Read

For the CDJDN
April 21, 2011

My Journey?

- 25 years IT programming and design
- FORTRAN->C->C++->Java->C#->Ruby
- Unix->VMS->DOS->Windows->Linux
- 6σ->SCJP->RHCE->GSEC->CISSP
- Violin->Clarinet->Piano->Voice
- American->Cheddar->Muenster->Jarlsberg

Our Journey?

- **Scope and Preconditions**
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- UI View
- UI Input
- State
- 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

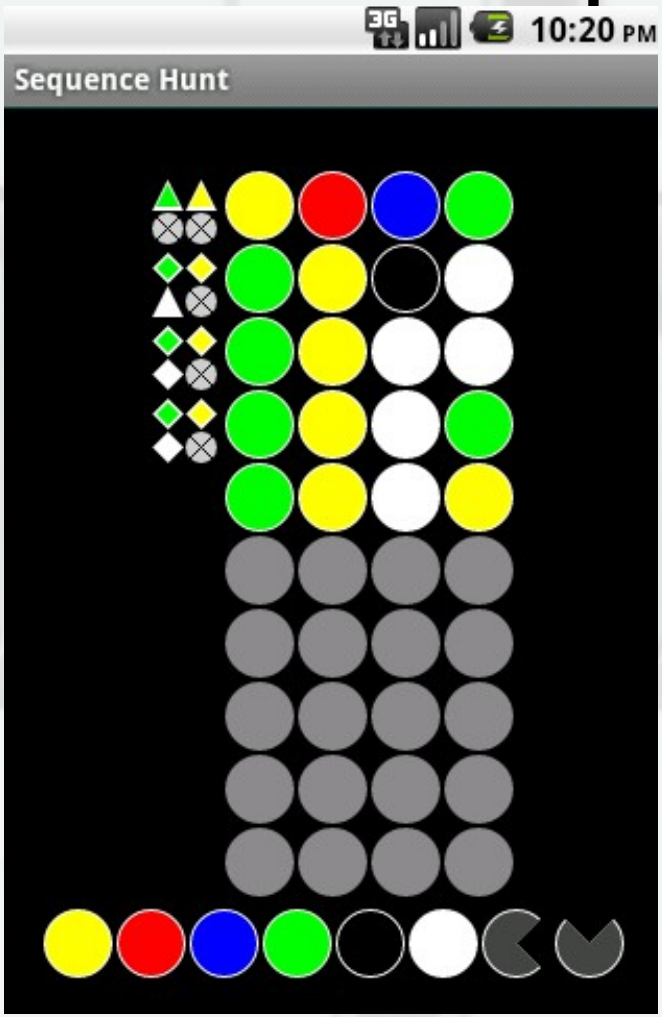
Geekic Overlook

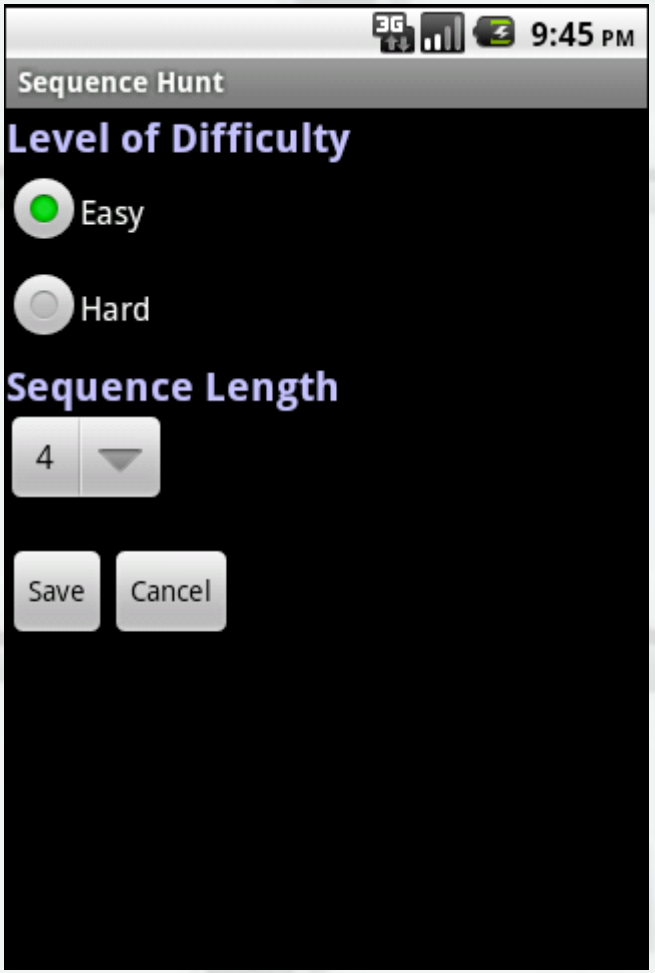
- Although the Java syntax is used and Java bytecode serves as an input for creating an Android application, the actual runtime environment is the Dalvik VM which does not support all the standard Java behaviors or libraries (like Swing)
- Geek points: Standard JVMs use a stack, the Delvik VM is register-based

Scope

- Android **Activity**
  - Usually programs for human interaction
  - Separate from **services** and **broadcast receivers**
- Introductory
  - Even though we have an hour or two, we won't explore every feature available or even half, quarter, eighth, …
- Interactive
  - Ask questions, participate

The Sample Application





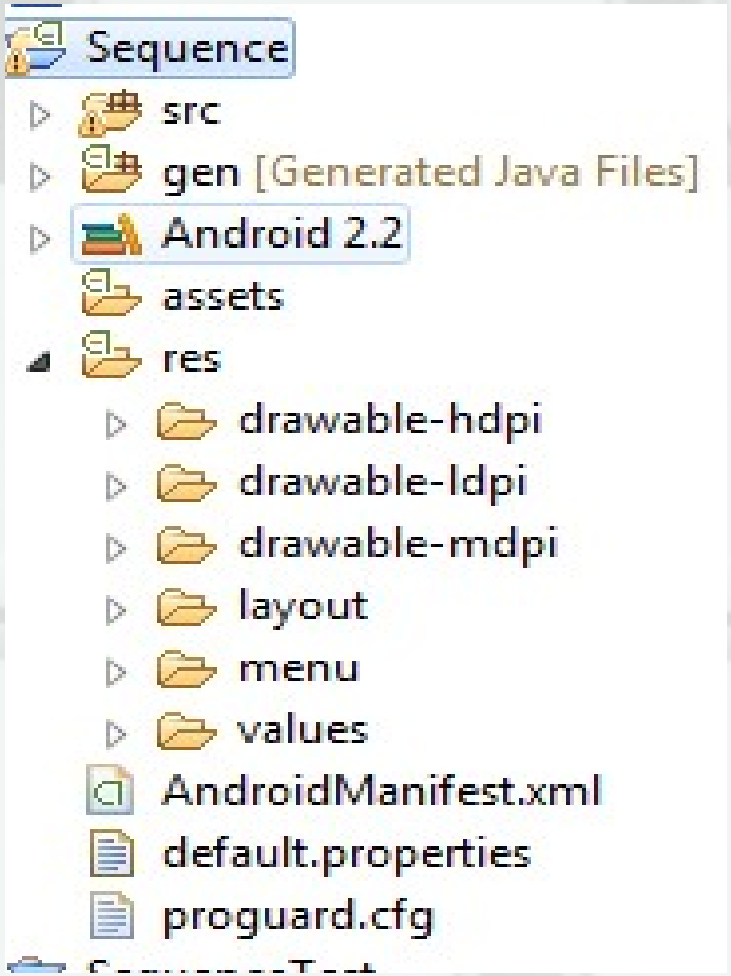
Preconditions

- Install Java 6
- Install Android SDK
  - latest is 3.0 as of 4/21/2011
- Unless you abhor productivity improvements, install Eclipse and the ADT plug-in
- Learn Java

Our Journey?

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

Project: Directory and Files



- Project
- src
- gen
- assets
- res
  - drawable-hdpi
  - drawable-mdpi
  - drawable-ldpi
  - layout
  - menu
  - values

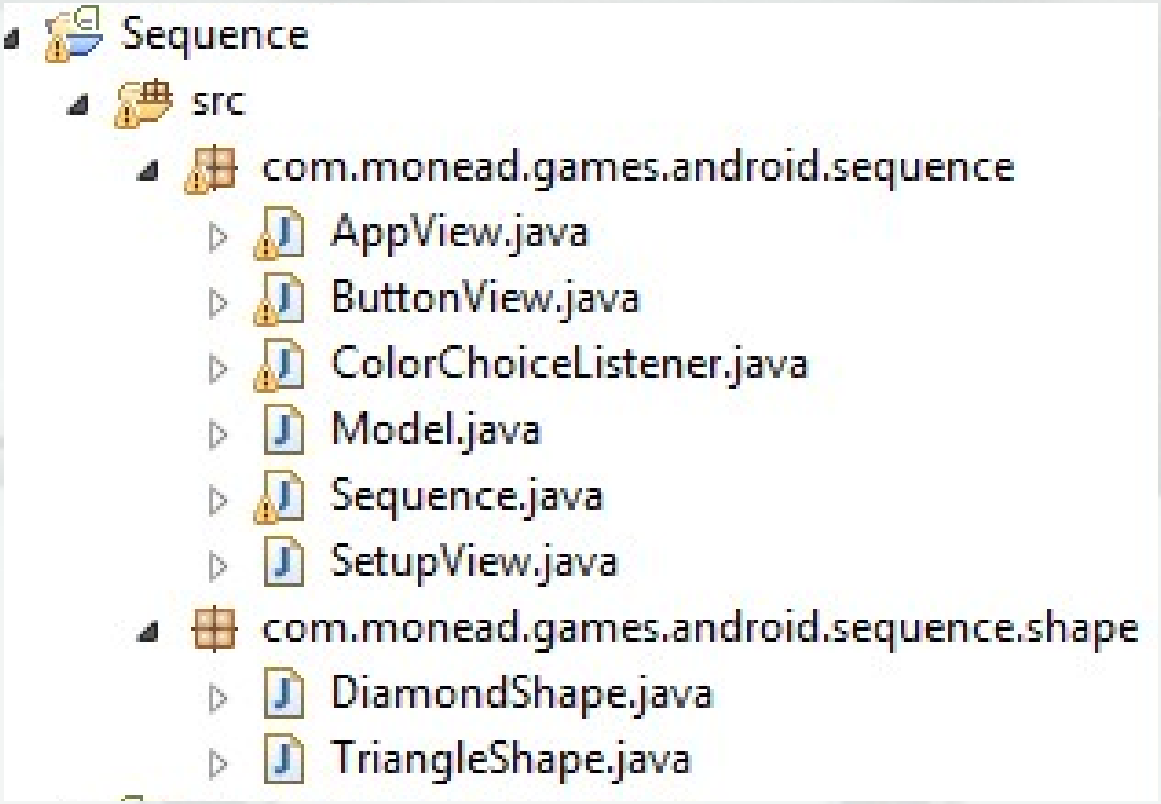
File: AndroidManifest.xml

- Configuration for the application
- Versioning, icon, initialization

```
<manifest package="com.monead.games.android.sequence"
 android:versionName="01.06" android:versionCode="17">
    <application android:icon="@drawable/launch_seq"
         android:label="@string/title_app_name">
        <activity android:name=".Sequence"
         android:label="@string/title_app_name">
            <intent-filter>...</intent-filter>
        </activity>
    </application>
    <uses-sdk android:targetSdkVersion="8"
         android:minSdkVersion="4">
    </uses-sdk>
</manifest>
```

Directory: src

- Application source code

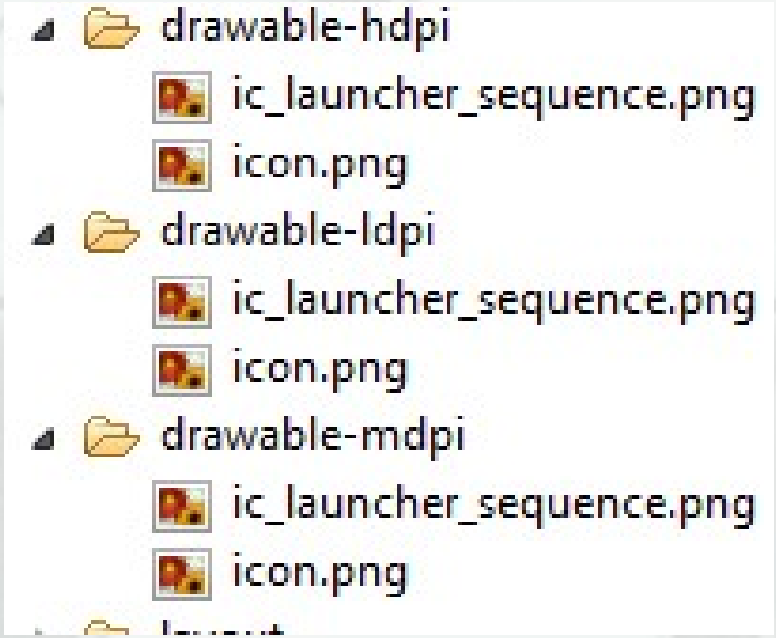


Directory: res/drawable-*

- 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

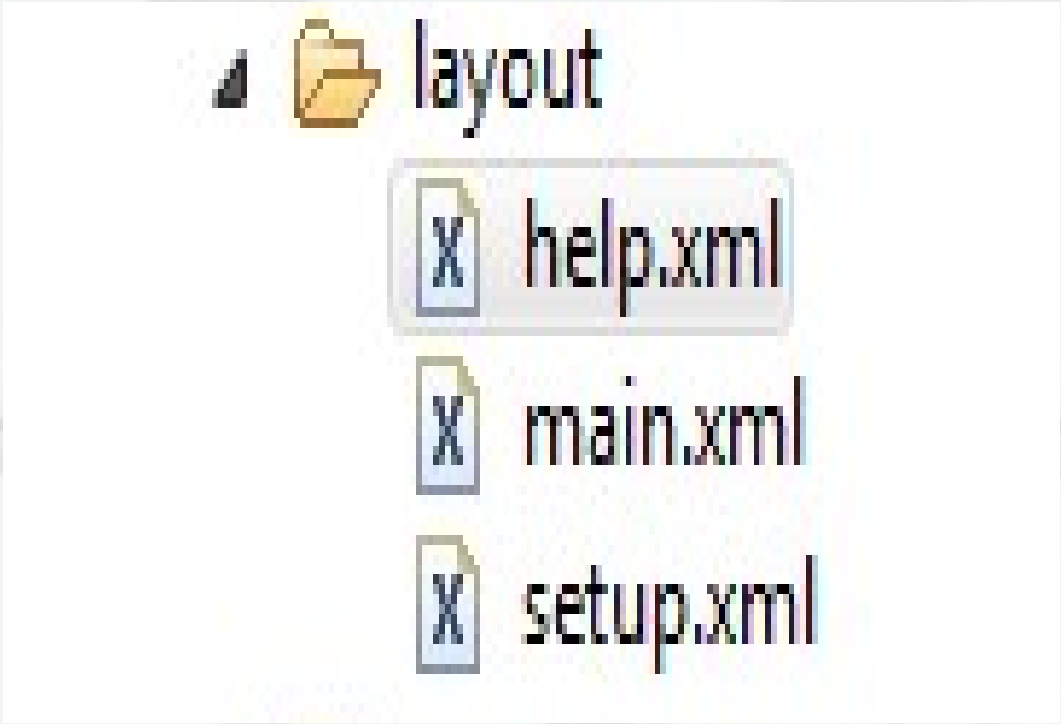
Example of Launcher Icon File

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



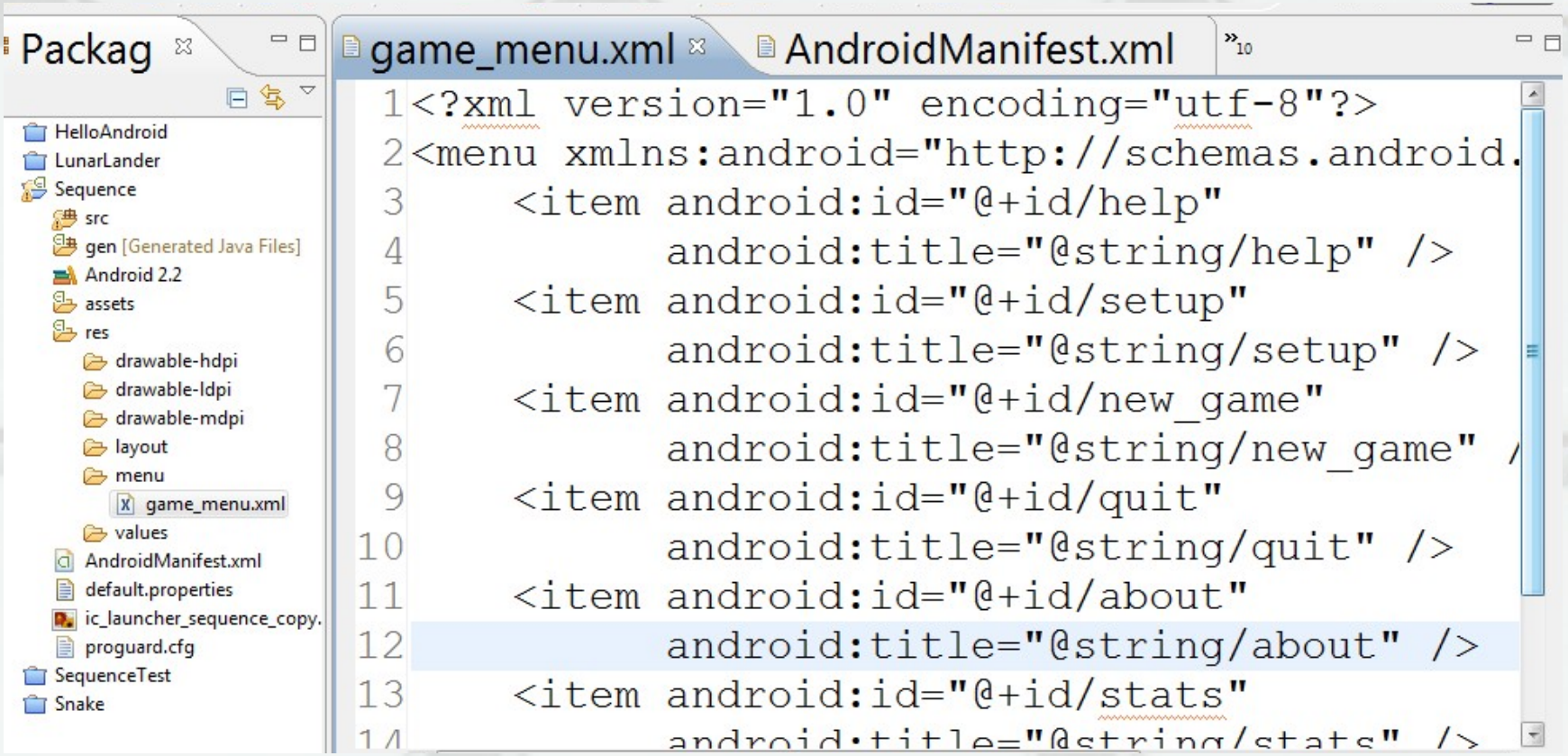
Directory: res/layout

- UI definitions
- Layouts
- Widgets



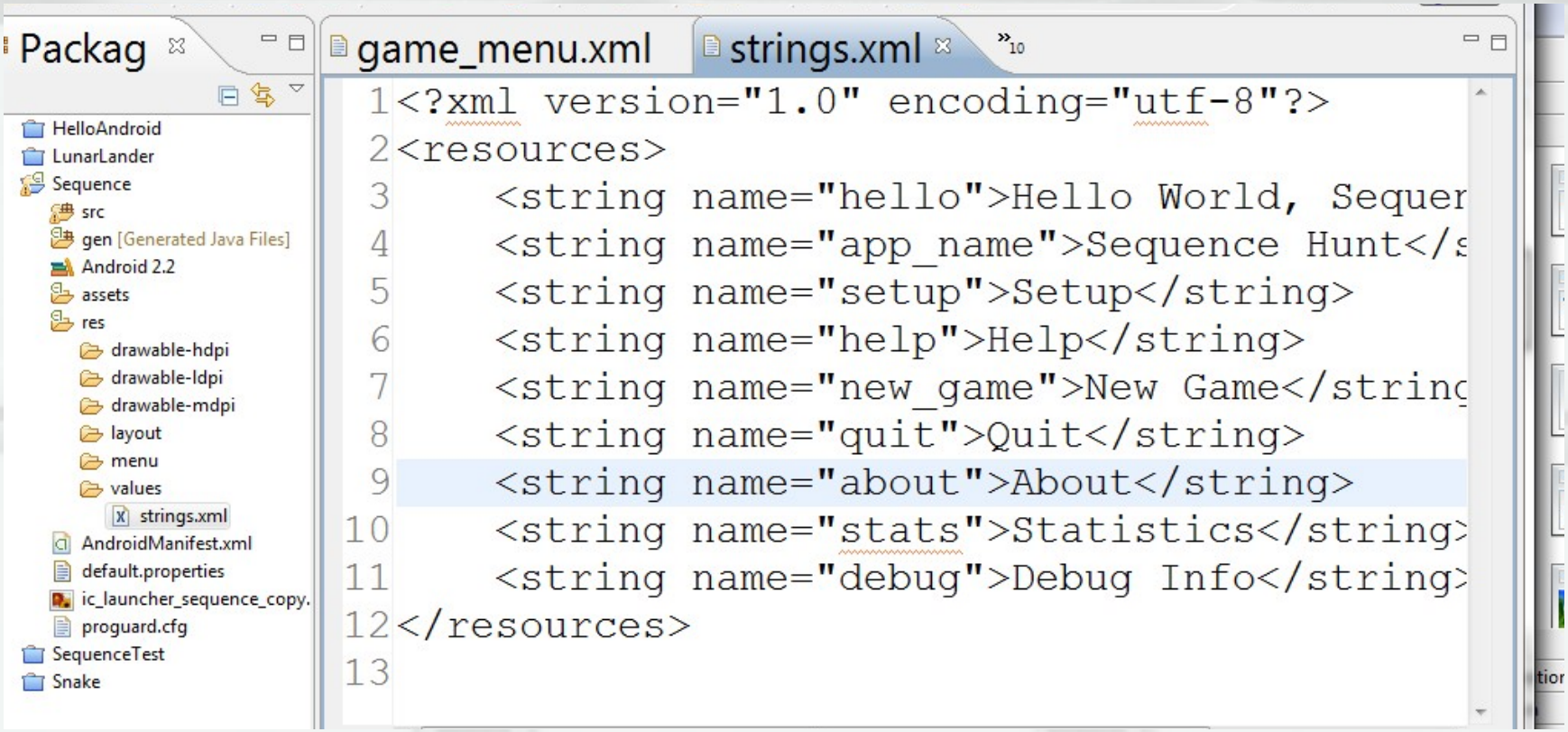
Directory: res/menu

- Defines menus accessible to the user



Directory: res/values

- Constants, supports internationalization



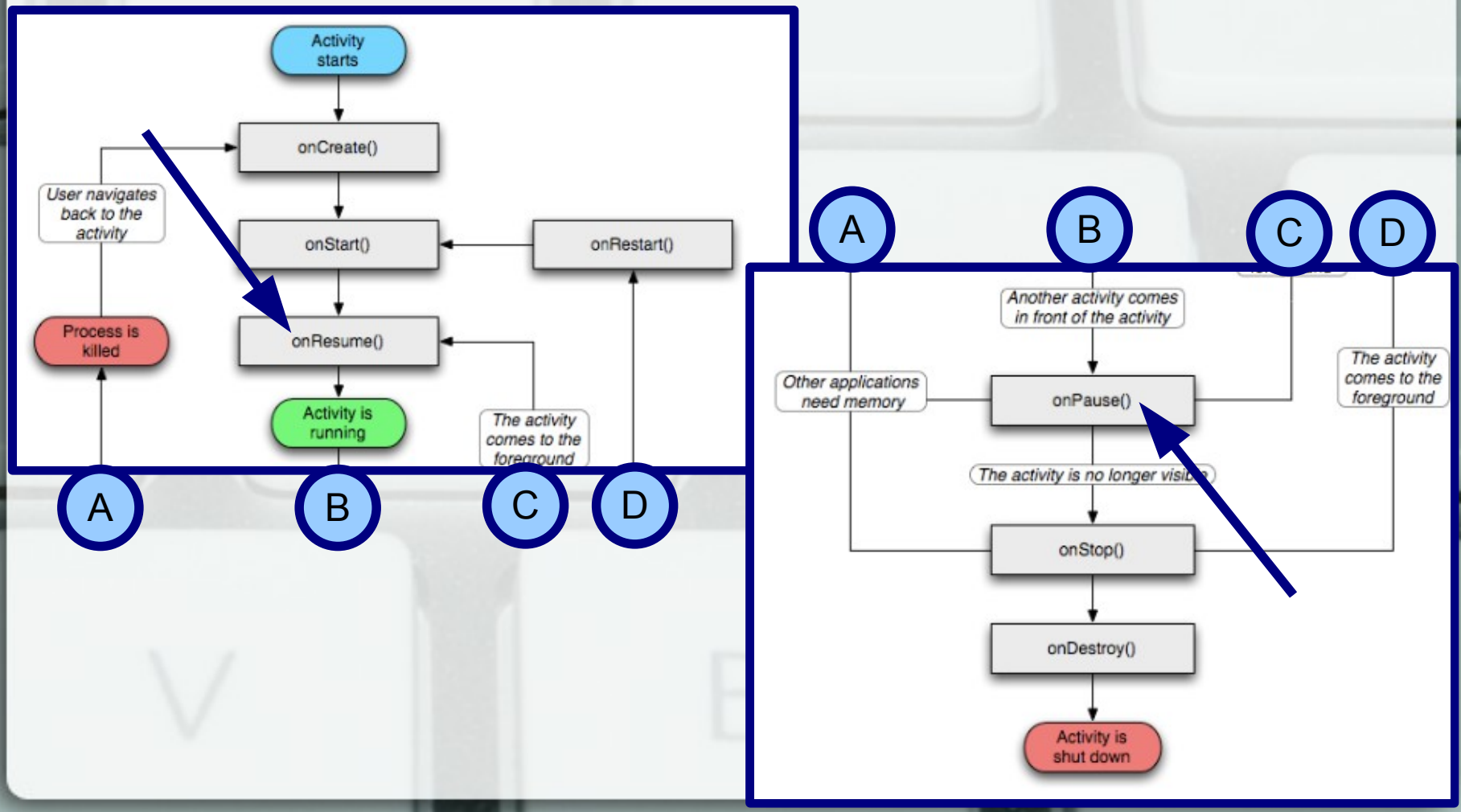
Our Journey?

- Scope and Preconditions
- Project: Directories and Files
- **Activity Lifecycle**
- Lifecycle Methods
- UI View
- UI Input
- State
- 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



Our Journey?

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- **Lifecycle Methods**
- UI View
- UI Input
- State
- 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 at Activity instance creation
@Override
protected void onCreate(Bundle savedState) {
   super.onCreate(savedState);
   gameBoard = new SequenceGameBoard(this);
   loadModel();
   gameBoard.setOnTouchListener(this);
   if (firstUse()) {
      setupForFirstUse();
   } else {
      displayGameboard();
   }
}
```

onPause()

```
// Called by environment if the
// app loses foreground control
@Override
protected void onPause() {
   super.onPause();
   setGameBoardNotVisible();
   saveModel();
   saveGameStatistics();
}
```

onResume()

```
// Called when an existing app
// instance regains fg control
@Override
protected void onResume() {
   super.onResume();
   loadModel();
}
```

Explore Some Source Code

- We'll switch to Eclipse and look at an Activity using some lifecycle methods

Our Journey?

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- **UI View**
- UI Input
- State
- 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

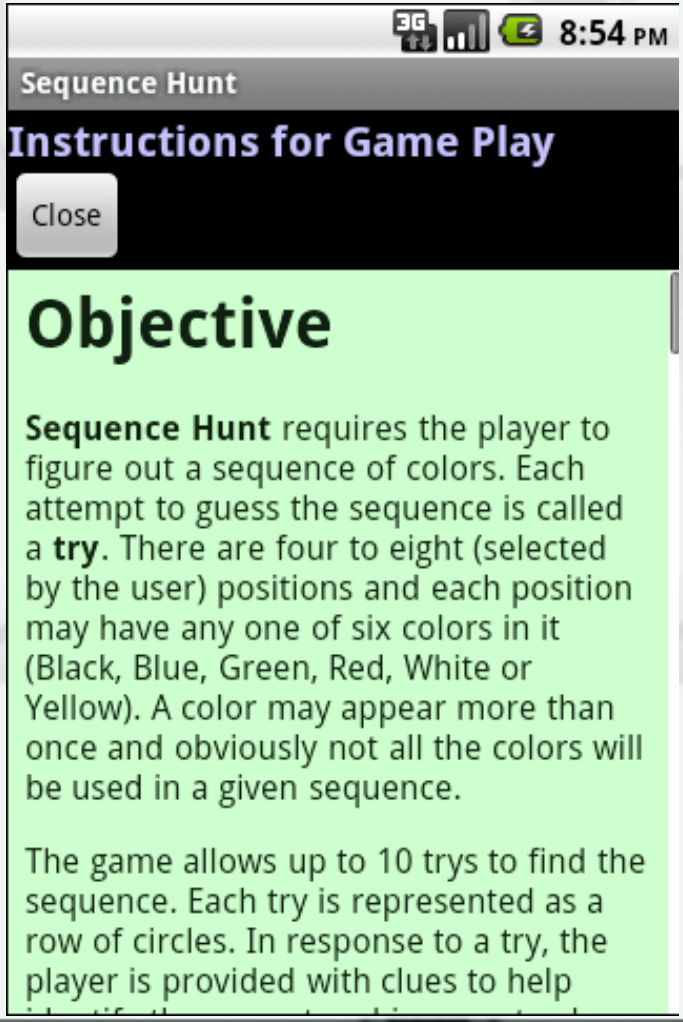
Simple Layout

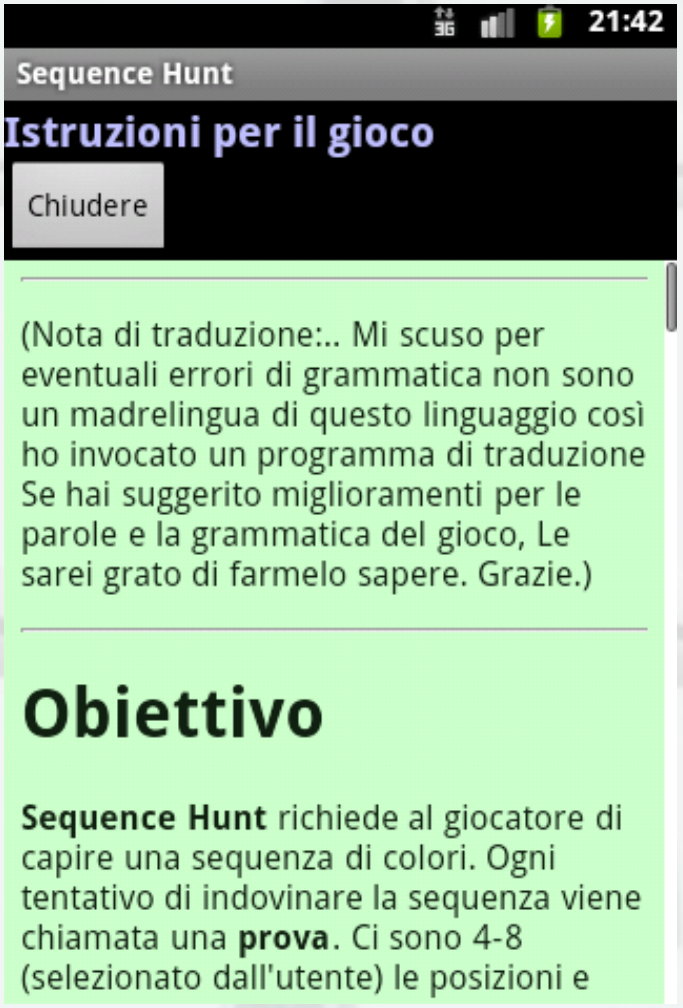
```
<!-- res/layout/help.xml -->
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/screen"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:focusable="true">
  <TextView ...>
  <Button android:id="@+id/button_close"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="@string/button_close" />
  <WebView android:id="@+id/instructions"
          android:layout_width="match_parent"
          android:layout_height="match_parent" />
</RelativeLayout>
```

Code Using the Prior UI *(partial)*

```
// This method is in the Activity class
private void showHelpScreen() {
   String url;
   setContentView(R.layout.help);
   url = "file:///android_asset/help/in.html";
   WebView instructions = (WebView)
       findViewById(R.id.instructions);
   instructions.loadUrl(url);
   ((Button) findViewById(R.id.button_close))
       .setOnClickListener(helpDoneClick);
}
```

Resulting Screen





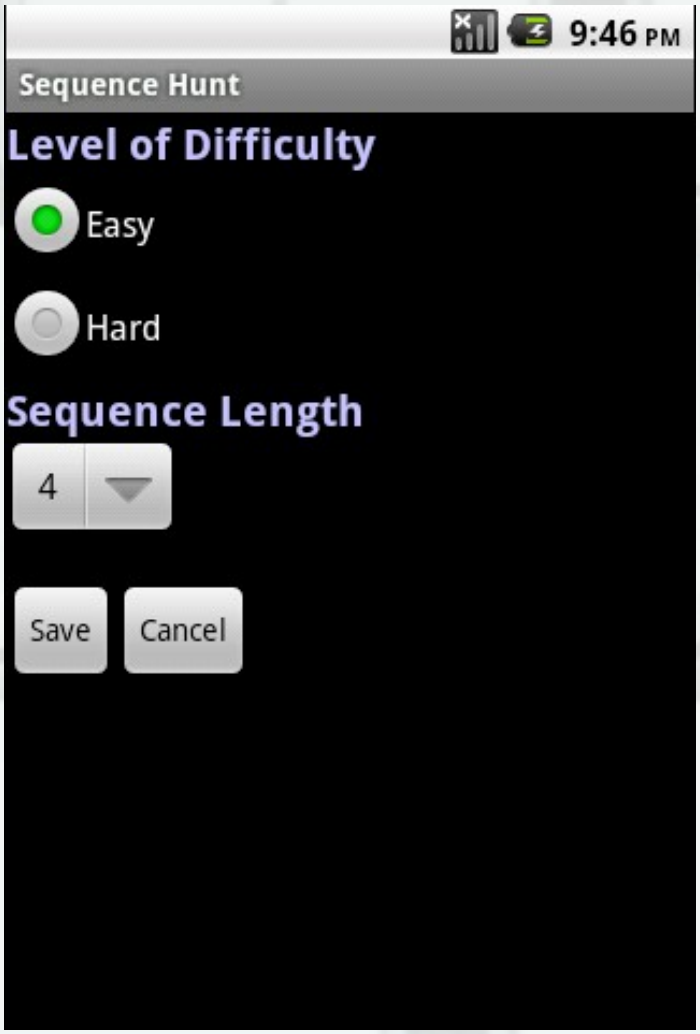
Layout with Multiple Widgets

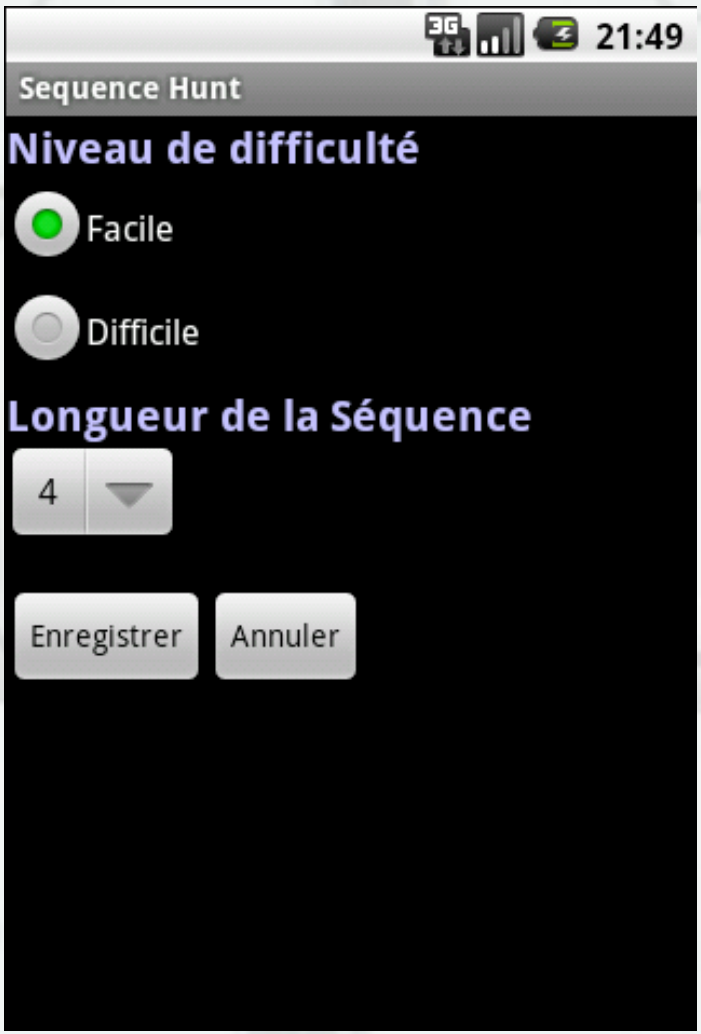
```
<!-- res/layout/setup.xml -->
<LinearLayout … android:orientation="vertical">
  <TextView android:id="@+id/label" … />
  <RadioGroup ...>
      <RadioButton android:id="@+id/radio_easy"
          android:text="Easy" />
      <RadioButton android:id="@+id/radio_hard"
          android:text="Hard" />
    </RadioGroup>
  <LinearLayout … android:orientation="horizontal"
    <Button android:id="@+id/button_save" …
            android:text="Save" />
    <Button android:id="@+id/button_cancel" …
            android:text="Cancel" />
  </LinearLayout>
</LinearLayout>
```

Code Using Prior UI *(partial)*

```
private void setupScreen() {
    setContentView(R.layout.setup);
    RadioButton easy = (RadioButton)
     findViewById(R.id.radio_easy);
    RadioButton hard = (RadioButton)
     findViewById(R.id.radio_hard);
    ((Button)
     findViewById(R.id.button_save))
      .setOnClickListener(setupSave);
    easy.setChecked(true);
}
```

Resulting Screen





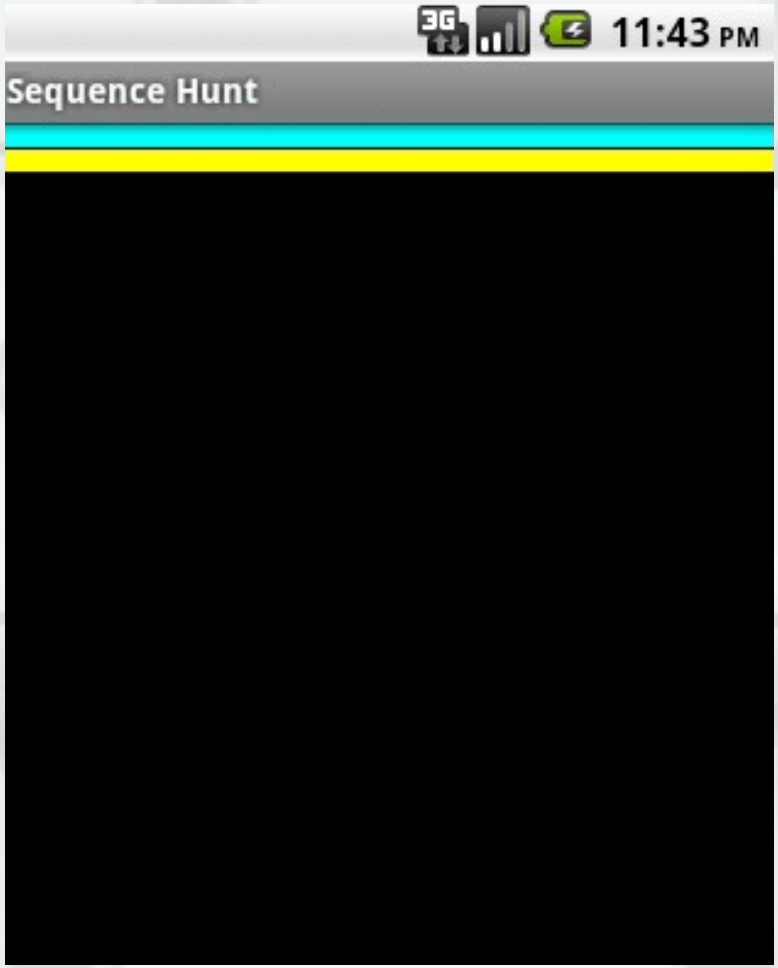
Programmatic UI

- Can define the view by extending the **android.view.View** class
- Override the **onDraw(Canvas canvas)** method
- Receives a Canvas instance, which represents the screen
- Use invalidate() method to force the view to redraw itself on the device

Drawing on the Canvas

```
ShapeDrawable drawable;
drawable = new ShapeDrawable(
  new RectShape());
drawable.getPaint()
  .setColor(Color.CYAN);
drawable.setBounds(0,0,getWidth(),10);
drawable.draw(canvas);
drawable.getPaint()
  .setColor(Color.YELLOW);
drawable.setBounds(0,11,getWidth(),20);
drawable.draw(canvas);
```

Resulting Screen



Creating Your Own Shapes

- Extend
  android.graphics.drawable.shapes.Shape
- Override the **draw()** method
  - You are given Canvas and Paint instances
- **android.graphics.Path** class is handy for connecting a set of line segments

Drawing a Diamond

```
@Override
public void draw(Canvas canvas,
    Paint paint) {
  Path path;
  path = new Path();
  path.moveTo(getWidth()/2, 0);
  path.lineTo(0, getHeight()/2);
  path.lineTo(getWidth()/2,getHeight());
  path.lineTo(getWidth(),
      getHeight()/2);
  path.lineTo(getWidth()/2, 0);
  canvas.drawPath(path, paint);
}
```

Resulting Shape

Menu

- Defined as XML resource
- Lifecycle methods in Activity
  - onCreateOptionsMenu(Menu m)
  - onOptionsMenuSelected(MenuItem mi)

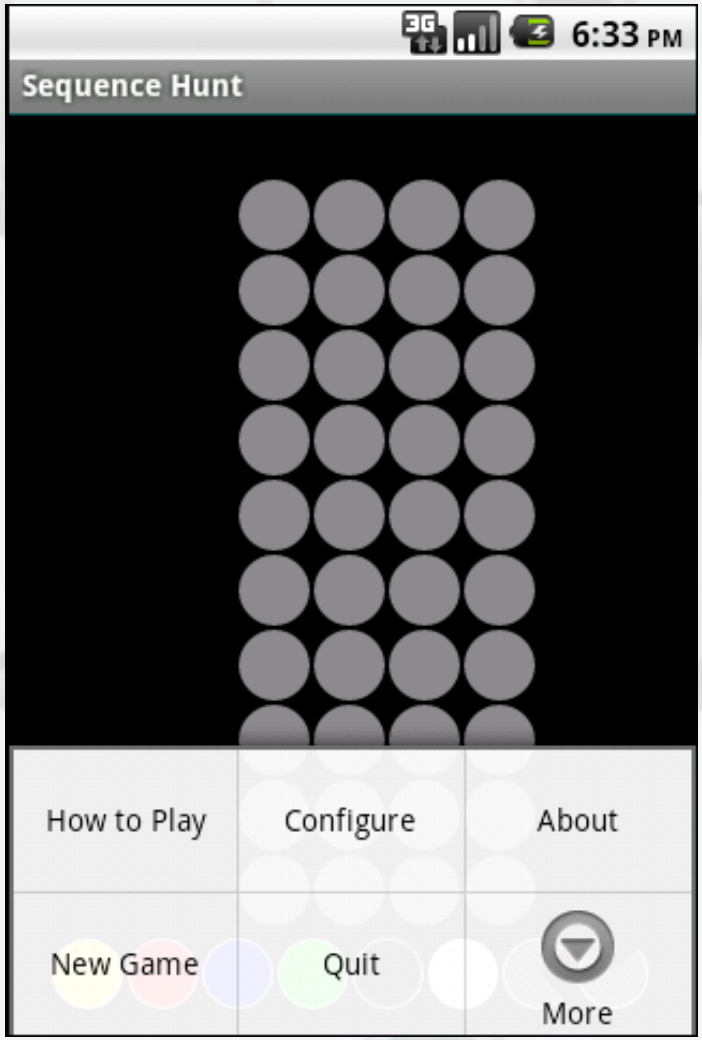
Game Menu Definition

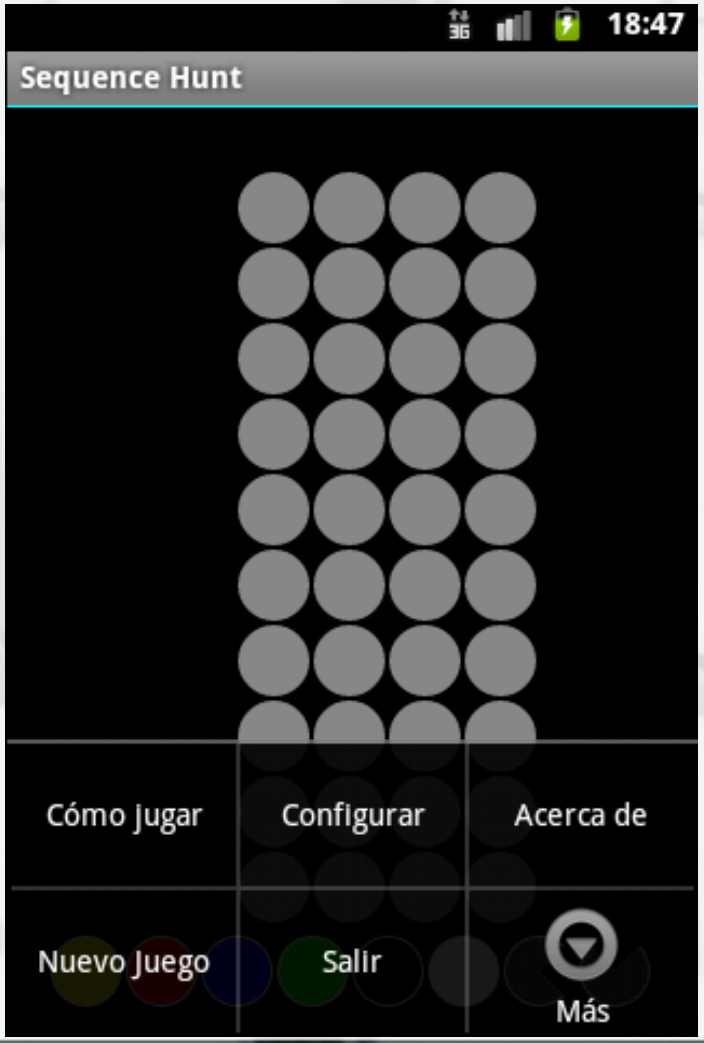
```
<!-- res/menu/game_menu.xml -->
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android=
      "http://schemas.android.com/apk/res/android">
    <item android:id="@+id/how_to_play"
      android:title=
      "@string/menu_how_to_play"></item>
    <item android:id="@+id/setup"
      android:title=
      "@string/menu_setup"></item>
</menu>
```

onCreateOptionsMenu()

```
@Override
public boolean onCreateOptionsMenu(
    Menu menu) {
  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.game_menu, menu);
  return true;
}
```

Menu Screenshot





onOptionsItemSelected()

```
@Override
public boolean onOptionsItemSelected(
    MenuItem item) {
  switch (item.getItemId()) {
    case R.id.how_to_play:
      showHelpScreen();
      return true;
  default:
    return super.onOptionsItemSelected(item);
  }
}
```

Dialog

- Typical modal dialog for forcing an interaction
  - Notifying the user
  - Choosing configuration options
- Default view allows scrollable text area and a close button

Dialog Lifecycle Methods

- onCreateDialog(int id)
  - Used to setup the dialog's structure and events (buttons and listeners)
- onPrepareDialog(int id, Dialog d, Bundle b)
  - Used to set runtime information each time the dialog is shown.
  - Pre-version 8 (prior to Android 2.2)
    - onPrepareDialog(int id, Dialog d)

onCreateDialog() Example

```
@Override
protected Dialog onCreateDialog(int id) {
 AlertDialog.Builder builder; Dialog dialog;
 switch (id) {  case DIALOG_INFO:
   builder = new AlertDialog.Builder(this);
   builder.setMessage("The Message").
   setCancelable(true).setNeutralButton("No",
   new DialogInterface.OnClickListener() {
     public void onClick(DialogInterface
       Dialog, int id){ dialog.cancel(); }});
      dialog = builder.create(); break;
  } return dialog;
}
```

onPrepareDialog() Example

```
@Override
protected void onPrepareDialog(int id,
    Dialog dialog, Bundle bundle) {
  switch (id) {
    case DIALOG_INFO:
      StringBuffer info = new StringBuffer();
      for (String detail:gameBd.getRtInfo()){
        info.append(detail);
      }
      ((AlertDialog)dialog)
        .setMessage(info.toString()); break;
  }
}
```

onPrepareDialog() Backward Compatibility

```
@Override
protected void onPrepareDialog(
    int id, Dialog dialog) {
  if (android.os.Build.VERSION.SDK_INT < 8) {
    onPrepareDialog(id, dialog, null);
  }
}
```

Dialog Screenshot

Internationalization

- Define strings in a resource file
- Directory determines language
  - res/values : default
  - res/values-LANGUAGE
  - res/values-LANGUAGE_COUNTRY
- Examples
  - res/values-es/strings.xml
  - res/values-es_MX/strings.xml

res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- en -->
<resources>
    <string name="title_app_name">
        Sequence Hunt</string>
    <string name="title_welcome">
        Welcome to Sequence Hunt!</string>
    <string name="title_how_to_play">
        Instructions for Game Play</string>
</resources>
```

res/values-de/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- de -->
<resources>
    <string name="title_app_name">
        Sequence Hunt</string>
    <string name="title_welcome">
        Willkommen in Sequence Hunt!</string>
    <string name="title_how_to_play">
        Anleitung für das Spiel</string>
</resources>
```

Get a Value – In A View

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout ...>
  <TextView android:id="@+id/label"
     android:text="@string/title_welcome" />
      <ScrollView ...><LinearLayout ...>
 <TextView
    android:id="@+id/welcome_message_initial"
    android:text="@string/title_welcome"/>
      </LinearLayout>  </ScrollView>
 </LinearLayout>
```

Get a Value - Programatically

```
((AlertDialog)dialog).setMessage(
    getResources().getString(
      R.string.message_win)
    + String.format(
      getResources().getString(
      R.string.message_playing_time),
  Formatter.getInstance().formatTimer(
    gameBoard.getModel().getElapsedTime()))
    + getResources().getString(
    R.string.question_play_again));
```

Our Journey?

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

UI Input

- Typical input sources
  - Keyboard
  - Touch Screen
- Event-based Paradigm
  - Register to listen for device events

Keyboard

- Could be physical or virtual
  - No difference for developer
- Activity will receive the events using callback methods
  - Override to process input
    - onKeyDown()
    - onKeyUp()
    - onKeyLongPress()
    - onKeyMultiple()

onKeyDown() Example

```
@Override
public boolean onKeyDown(int code, KeyEvent evt) {
  if (code == KeyEvent.KEYCODE_ENTER) {
     gameBoard.notifyTry();
   } else if (code == KeyEvent.KEYCODE_DEL) {
     gameBoard.notifyDeleteChoice();
   } else {
     return super.onKeyDown(code, evt);
   }
  return true;
}
```

Touch Screen Input

- Must implement an interface
  - android.view.View.OnTouchListener
- Declares one method
  - onTouch(View v, MotionEvent event)
- Register to receive touch events
- Method defined on View class
  - setOnTouchListener(OnTouchListener l)

onTouch() Example in Activity

```
@Override
public boolean onTouch(View v, MotionEvent evt) {
  boolean p;
  if (v instanceof OnTouchListener) {
    p = ((OnTouchListener)v).onTouch(v, evt);
    if (p) {
      if (gameBd.getModel().isWinner()) {
        showDialog(DIALOG_WIN);
      }
    }
    return true;
  }
  return false;
}
```

MotionEvents

- How was the screen touched
- Constants for decoding the events through the lifecycle of a gesture
- Examples
  - ACTION_DOWN
  - ACTION_MOVE
  - ACTION_UP
  - ACTION_CANCEL

onTouch() Using MotionEvent

```
public boolean onTouch(View view,
    MotionEvent event) {
  if (event.getAction() ==
      MotionEvent.ACTION_DOWN) {
    processLocation(event.getX(),
        event.getY());
    invalidate();
    return true;
  }
  return false;
}
```

Our Journey?

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

Persisting State

- Shared Preferences
  - Store primitive data in key-value pairs
- Internal Storage
  - Store arbitrary data on the device memory
- External Storage
  - Store data on shared external storage
- SQLite Databases
  - Store structured data in a private database
- Network Connection
  - Store data on network-accessible device

Shared Preferences

- Simple key-value pairs
- **getSharedPreferences()** available on the **Activity** class (inherited from **Context**)
- Supply a **file name** and a **privacy mode** for data visibility
  - MODE_PRIVATE
  - MODE_WORLD_READABLE
  - MODE_WORLD_WRITEABLE

Setting SharedPreferences Value

```
SharedPreferences settings =
   getSharedPreferences(
   “Sequence.preferences”,
   MODE_PRIVATE);
SharedPreferences.Editor editor =
   settings.edit();
editor.putBoolean(“ModeHard”,
   true);
editor.commit();
```

Getting SharedPreferences Value

```
SharedPreferences settings =
    getSharedPreferences(
    "Sequence.preferences",
    MODE_PRIVATE);

someBean.setModeHard(
    settings.getBoolean(
    "ModeHard", false));
```

Internal Storage

- Files associated with application
- You may expose files to other applications
- **openFileInput()**, **openFileOutput()** available on the Activity instance (inherited from Context)
- Just like SharedPreferences, you supply a **file name** and, for ***output***, a **privacy mode**

Write to an Internal File

```
ObjectOutputStream out = null;
try {
  out = new
    ObjectOutputStream(
      openFileOutput("SeqModel.ser",
        MODE_PRIVATE));
  out.writeObject(someBean.getModel());
}
finally {
  if (out != null) {
    out.close();
  }
}
```

Read from an Internal File

```
ObjectInputStream in = null;
try {
   in = new ObjectInputStream(
      openFileInput("SeqModel.ser"));
   Model model = (Model) in.readObject();
}
finally {
   if (in != null) {
      in.close();
   }
}
```

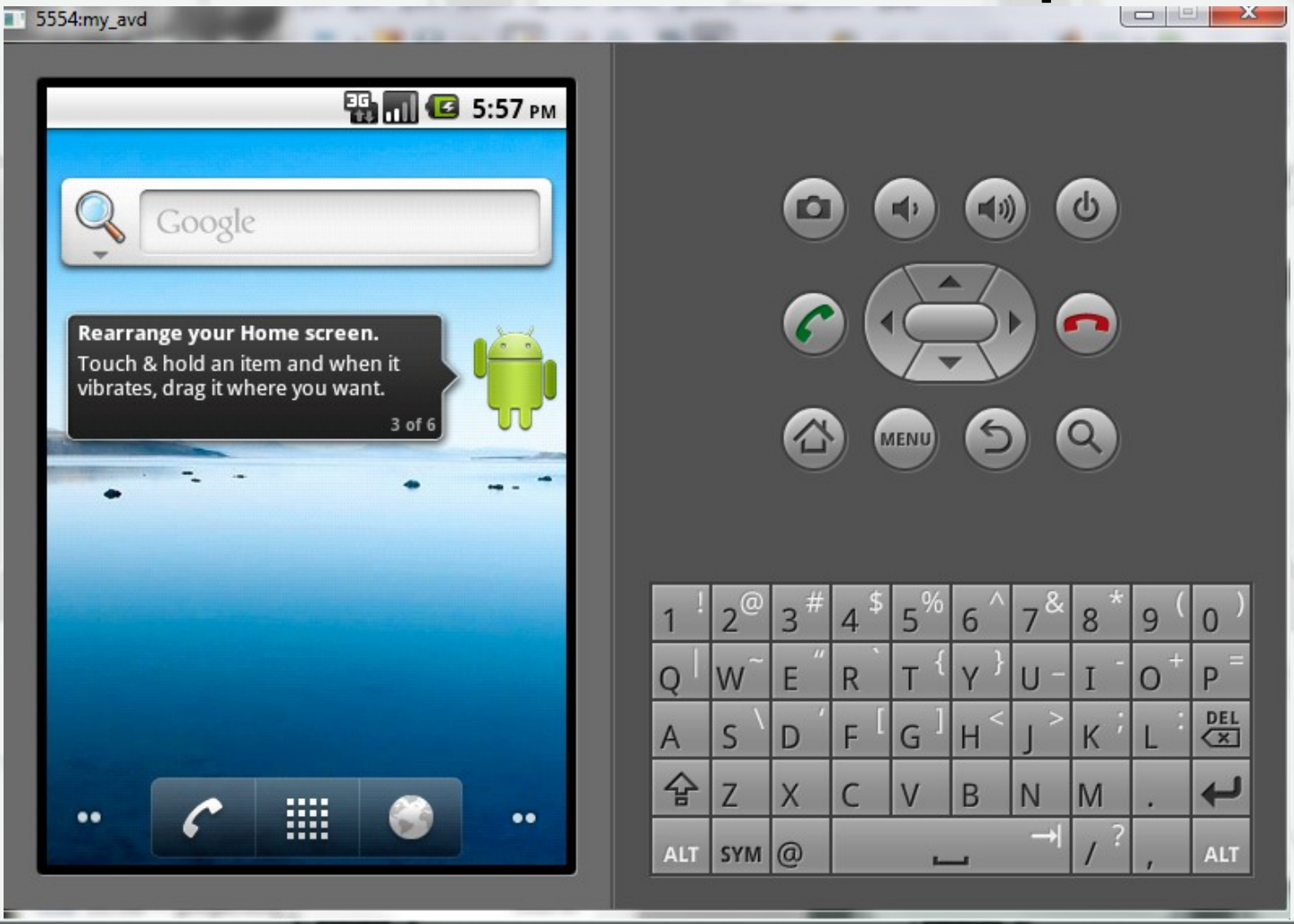
Our Journey?

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

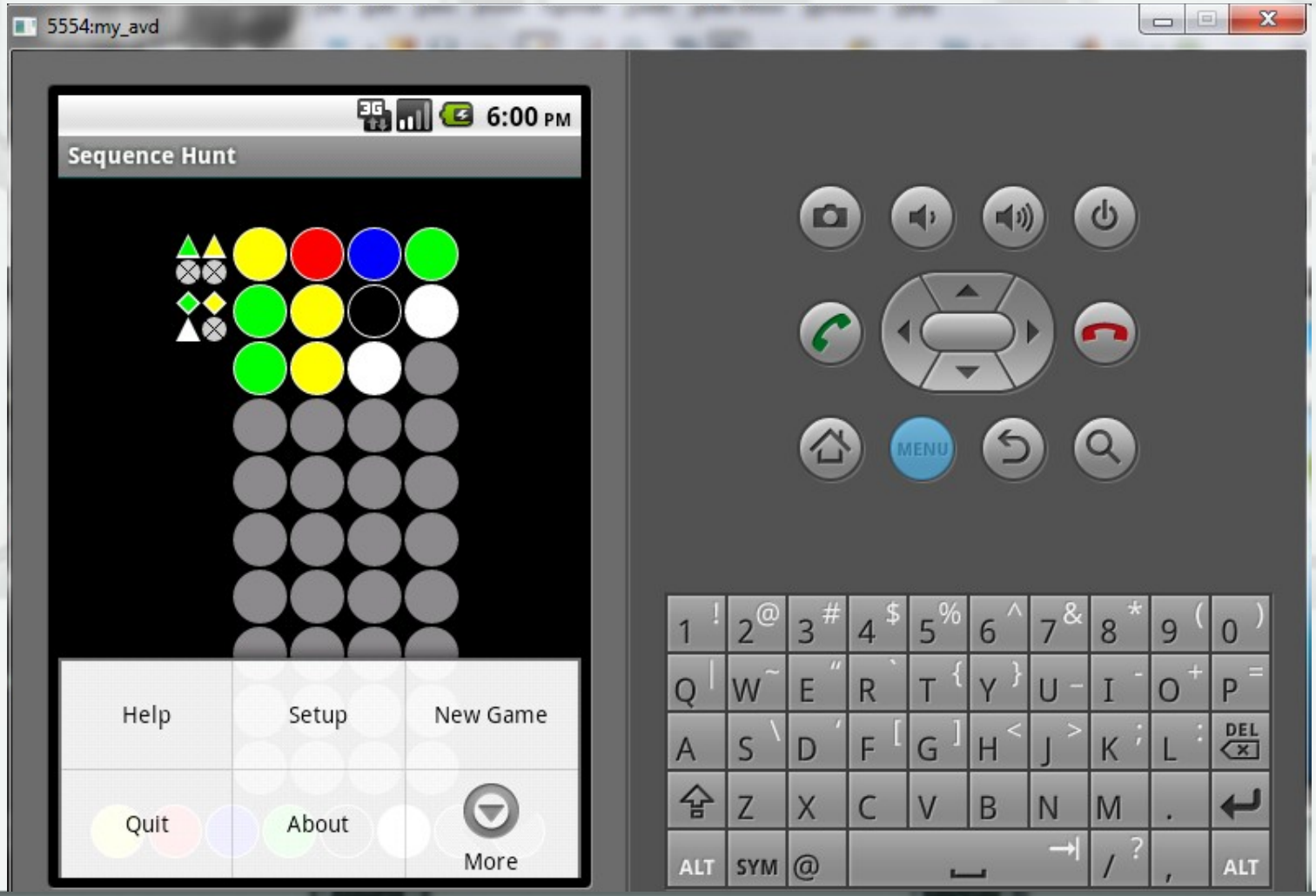
Testing

- 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

Emulator at Startup



Emulator Running an App



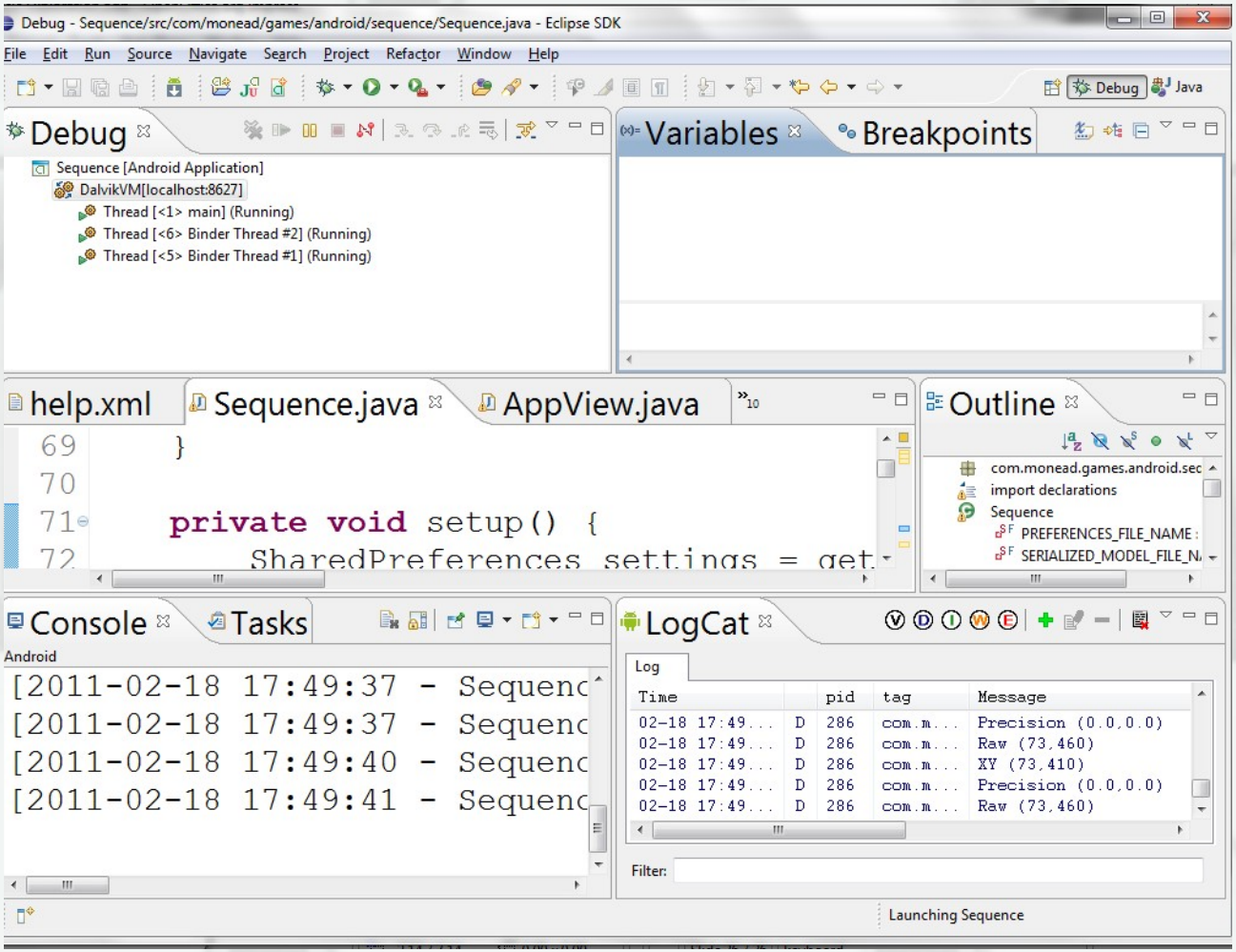
Debugging

- 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

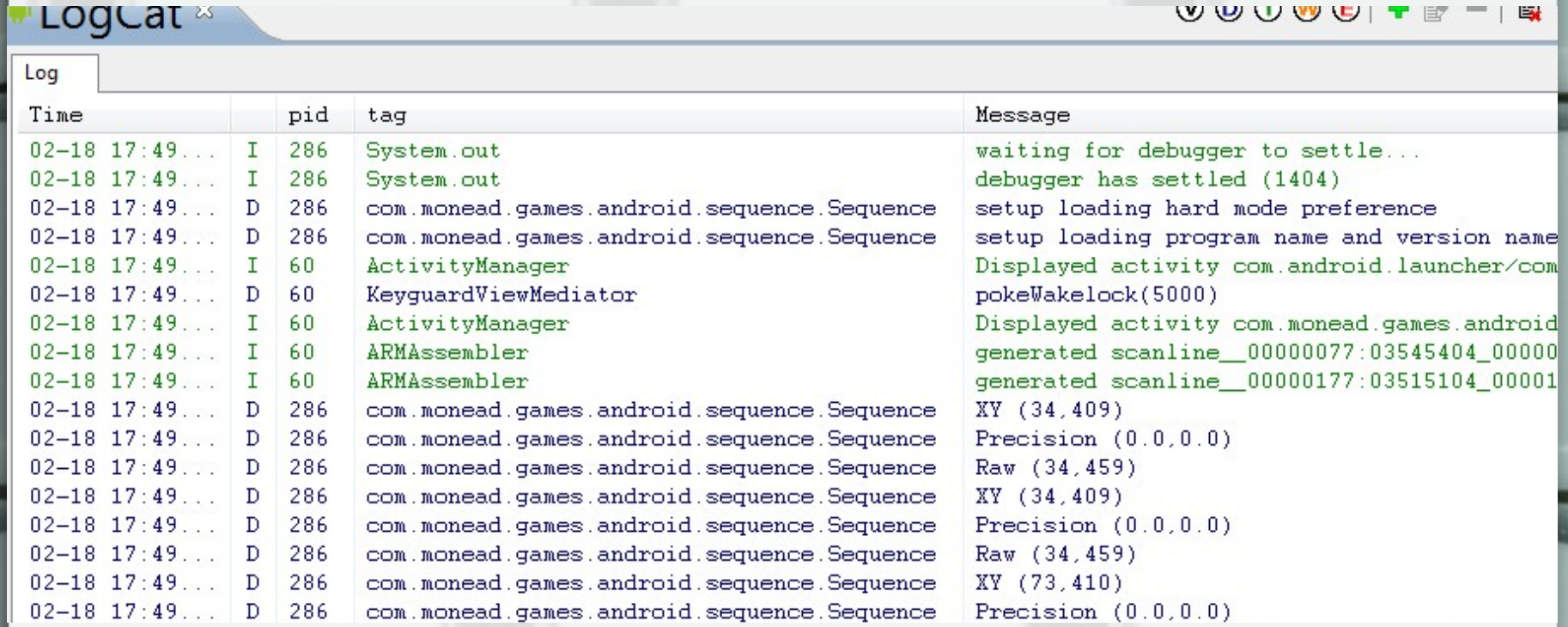
Sample Code with Logging

```
try {
   PackageInfo pi = GetPackageManager()
      .getPackageInfo(
      "com.monead.games.android.sequence", 0);
   progName = getPackageManager()
      .getApplicationLabel(
      getApplicationInfo()).toString();
   progVersion = pi.versionName;
}
catch (Throwable throwable) {
 Log.e(tag, "Unable to …", throwable);
}
Log.d(tag, progName + "/" + progVersion);
}
```

Debug View


Debug - Sequence/src/com/monead/games/android/sequence/Sequence.java - Eclipse SDK
File Edit Run Source Navigate Search Project Refactor Window Help
Debug Java
Debug
Sequence [Android Application]
DalvikVM[localhost:8627]
Thread [<1> main] (Running)
Thread [<6> Binder Thread #2] (Running)
Thread [<5> Binder Thread #1] (Running)
Variables
Breakpoints
help.xml
Sequence.java
AppView.java
69 }
70
71 private void setup() {
72 SharedPreferences settings = get
Outline
com.monead.games.android.sec
import declarations
Sequence
PREFERENCES_FILE_NAME
SERIALIZED_MODEL_FILE_N
Console
Tasks
Android
[2011-02-18 17:49:37 - Sequenc
[2011-02-18 17:49:37 - Sequenc
[2011-02-18 17:49:40 - Sequenc
[2011-02-18 17:49:41 - Sequenc
LogCat
Log
Time pid tag Message
02-18 17:49... D 286 com.m... Precision (0.0,0.0)
02-18 17:49... D 286 com.m... Raw (73,460)
02-18 17:49... D 286 com.m... XY (73,410)
02-18 17:49... D 286 com.m... Precision (0.0,0.0)
02-18 17:49... D 286 com.m... Raw (73,460)
Filter:
Launching Sequence

Log Entries


LogCat
Log
Time | | pid | tag | Message
02-18 17:49... I 286 System.out waiting for debugger to settle...
02-18 17:49... I 286 System.out debugger has settled (1404)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence setup loading hard mode preference
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence setup loading program name and version name
02-18 17:49... I 60 ActivityManager Displayed activity com.android.launcher/com
02-18 17:49... D 60 KeyguardViewMediator pokeWakelock(5000)
02-18 17:49... I 60 ActivityManager Displayed activity com.monead.games.android
02-18 17:49... I 60 ARMAssembler generated scanline__00000077:03545404_00000
02-18 17:49... I 60 ARMAssembler generated scanline__00000177:03515104_00001
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence XY (34,409)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence Precision (0.0,0.0)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence Raw (34,459)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence XY (34,409)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence Precision (0.0,0.0)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence Raw (34,459)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence XY (73,410)
02-18 17:49... D 286 com.monead.games.android.sequence.Sequence Precision (0.0,0.0)

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

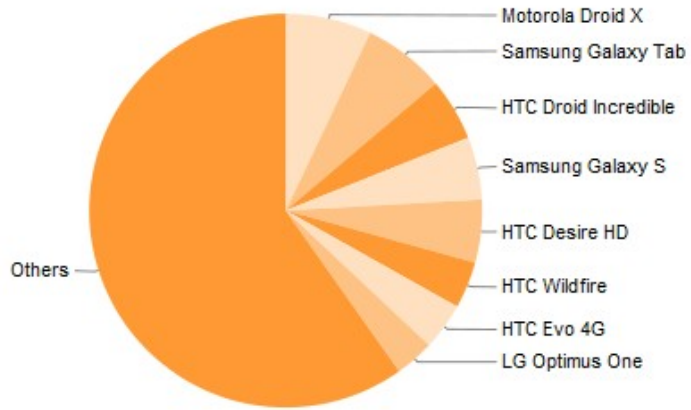
- Screenshots
- High-res Icon
- Promotional icons and videos
- Title and description
  - Internationalization supported
- Application type and category
  - e.g. Games, Brain&Puzzle

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%

Our Journey?

- Scope and Preconditions
- Project: Directories and Files
- Activity Lifecycle
- Lifecycle Methods
- UI View
- UI Input
- State
- Testing and Debugging
- **Q&A**

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
- https://github.com/DaveRead/SequenceHunt

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