Beginning Java for Android
Session 2: Android— a handheld robot

Izzy Johnston

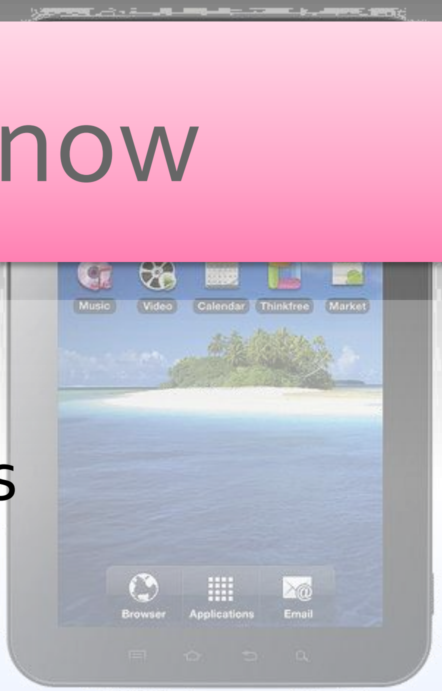
izzycjohnston@gmail.com

@izzy_johnston

What is Android?

- Software Stack
  - Operating System
  - Middleware
  - Applications
- Founded in 2003, bought by Google in 2005
- Open source Software Development Kit
  - SDK to the initiated

What you need to know



- Multiple devices
  - OS and device are not synonymous
  - Phones, Tablets, Nook e-reader
- Apps written in Java
- Can call external APIs
  - Webkit, Geo-referencing, OpenGL

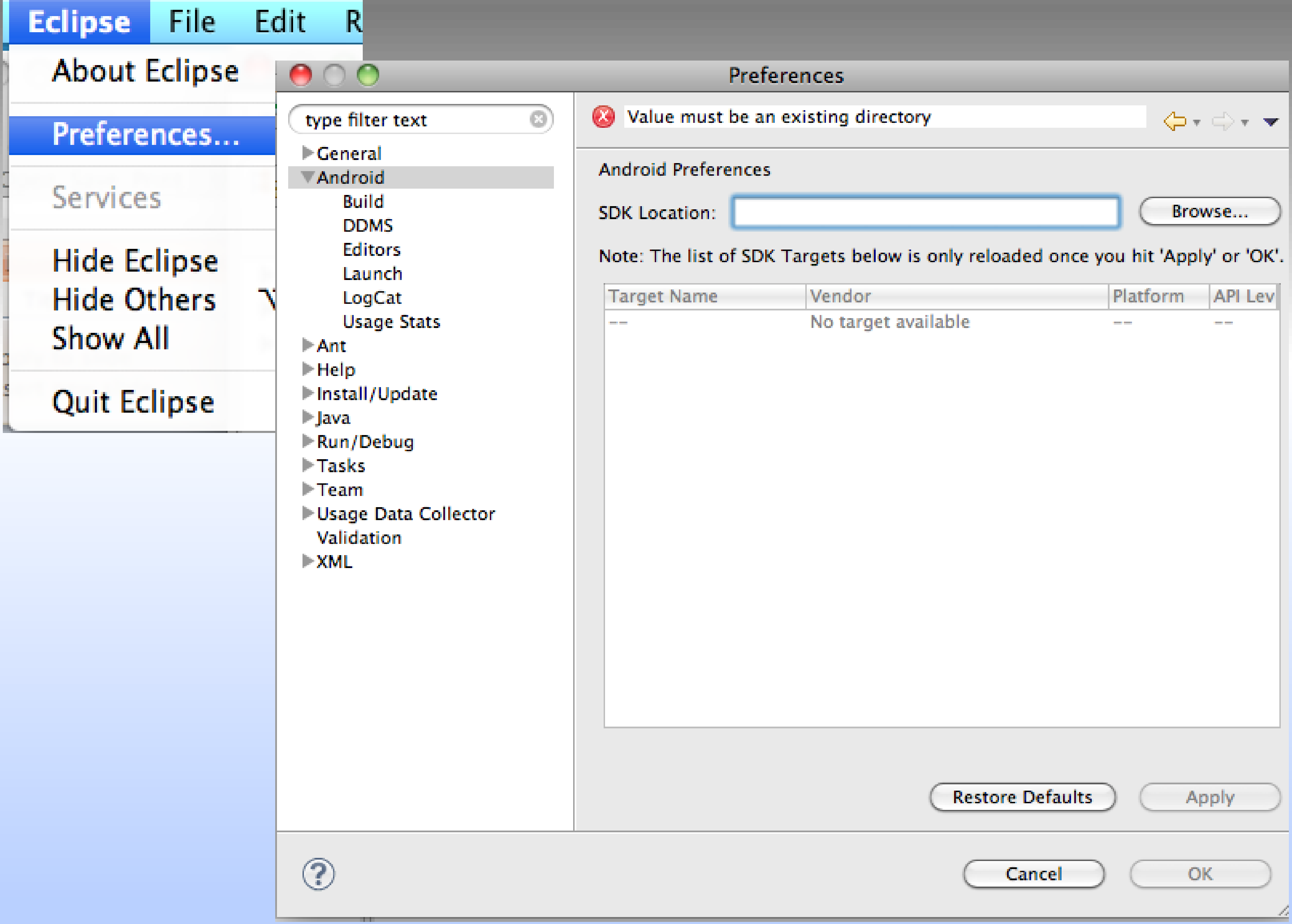
Getting Started

Tell Eclipse that you will be using the Android SDK

Create a New Android project

Using the Android SDK is like using specific lego blocks.

Fancier ones. Like Technic.



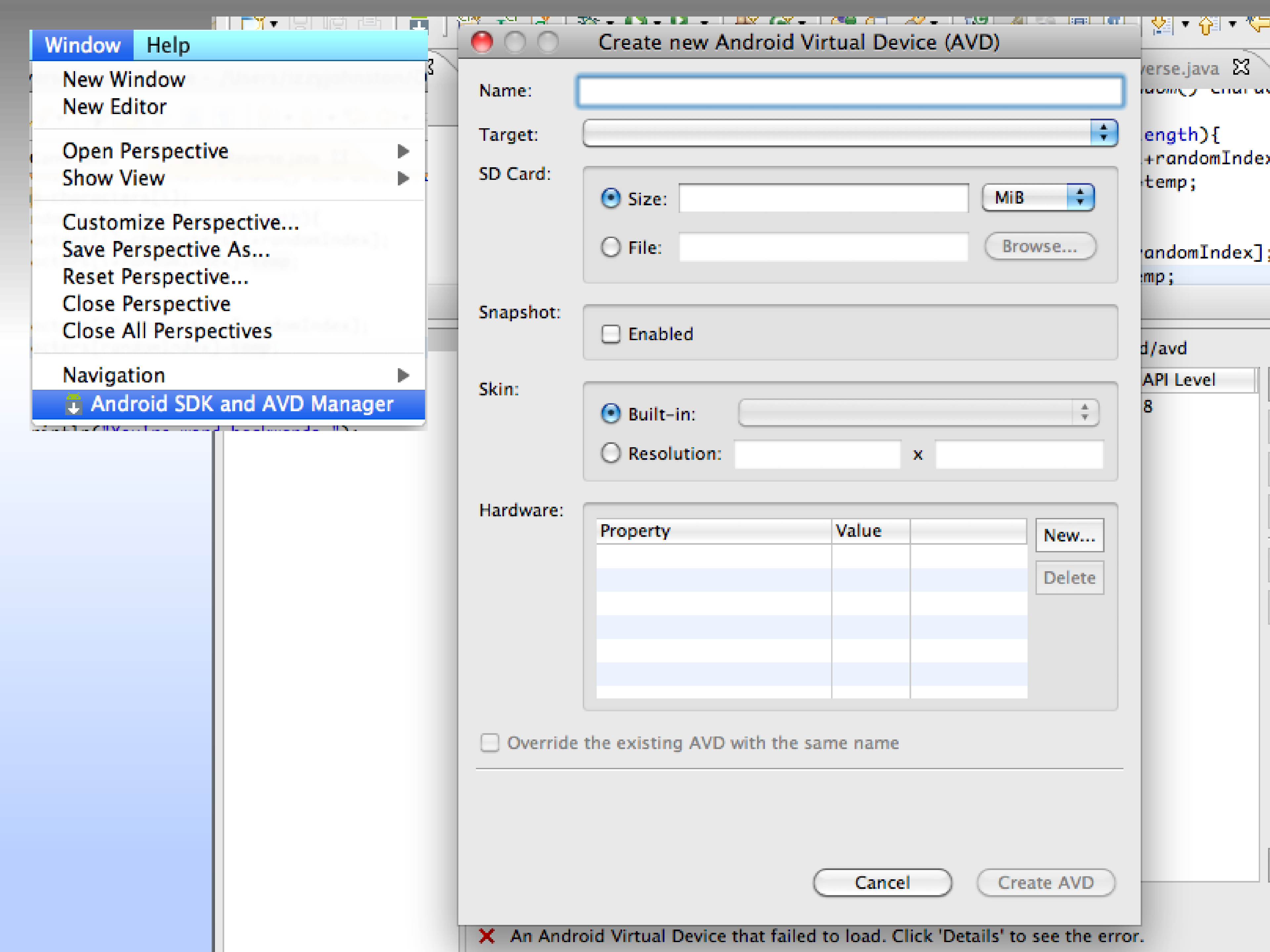
Value must be an existing directory

Android Preferences

SDK Location:

Note: The list of SDK Targets below is only reloaded once you hit 'Apply' or 'OK'.

Target Name	Vendor	Platform	API Lev
--	No target available	--	--



Name:

Target:

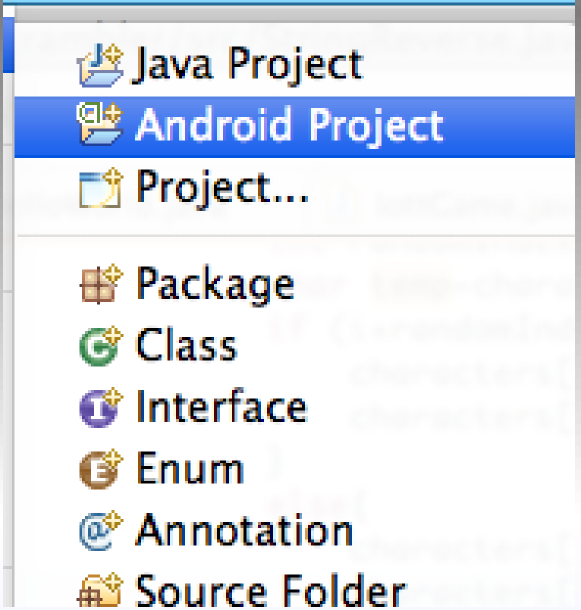
SD Card: ● Size: MiB ○ File:

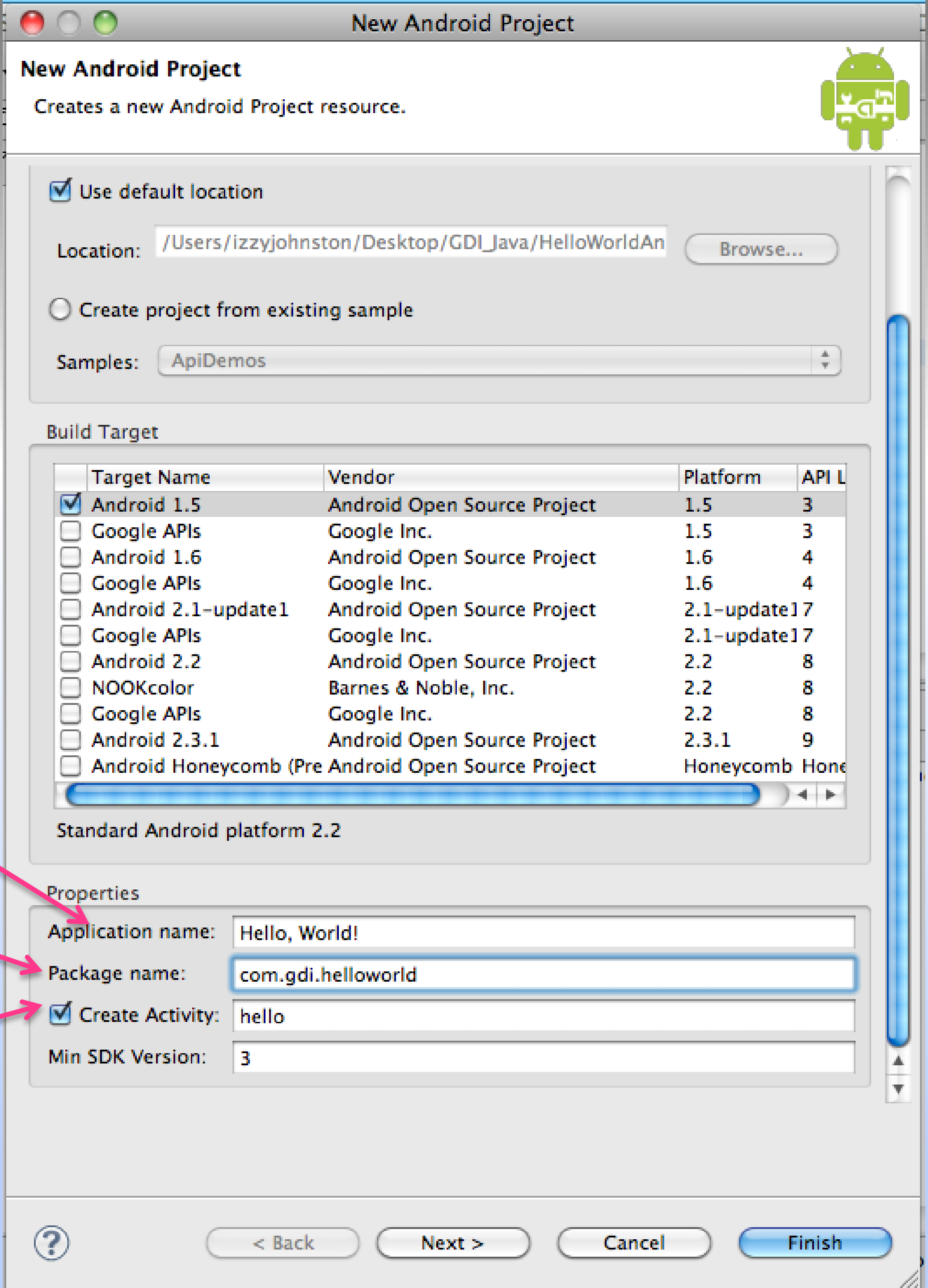
Snapshot: □ Enabled

Skin: ● Built-in: ○ Resolution: x

□ Override the existing AVD with the same name

An Android Virtual Device that failed to load. Click 'Details' to see the error.





New Android Project

New Android Project

Creates a new Android Project resource.

☑ Use default location

Location: /Users/izzyjohnston/Desktop/GDI_Java/HelloWorldAn

○ Create project from existing sample

Samples: ApiDemos

Build Target

	Target Name	Vendor	Platform	API L
☑	Android 1.5	Android Open Source Project	1.5	3
□	Google APIs	Google Inc.	1.5	3
□	Android 1.6	Android Open Source Project	1.6	4
□	Google APIs	Google Inc.	1.6	4
□	Android 2.1-update1	Android Open Source Project	2.1-update1	7
□	Google APIs	Google Inc.	2.1-update1	7
□	Android 2.2	Android Open Source Project	2.2	8
□	NOOKcolor	Barnes & Noble, Inc.	2.2	8
□	Google APIs	Google Inc.	2.2	8
□	Android 2.3.1	Android Open Source Project	2.3.1	9
□	Android Honeycomb (Pre	Android Open Source Project	Honeycomb	Hone

Standard Android platform 2.2

Properties

Application name: Hello, World!

Package name: com.gdi.helloworld

☑ Create Activity: hello

Min SDK Version: 3

Application Name—what it will say by the icon

Package—unique identifier of whole app

Activity—automatically generated class

File Structure

- src—contains the guts, all classes of app
- gen-generated Java files
- Build SDK
- assets—non-Java files
- res—layouts, images, etc.
- AndroidManifest.xml—ever important permissions

Your First Android App

```
package com.gdi.helloworld;

import android.app.Activity;
import android.os.Bundle;

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

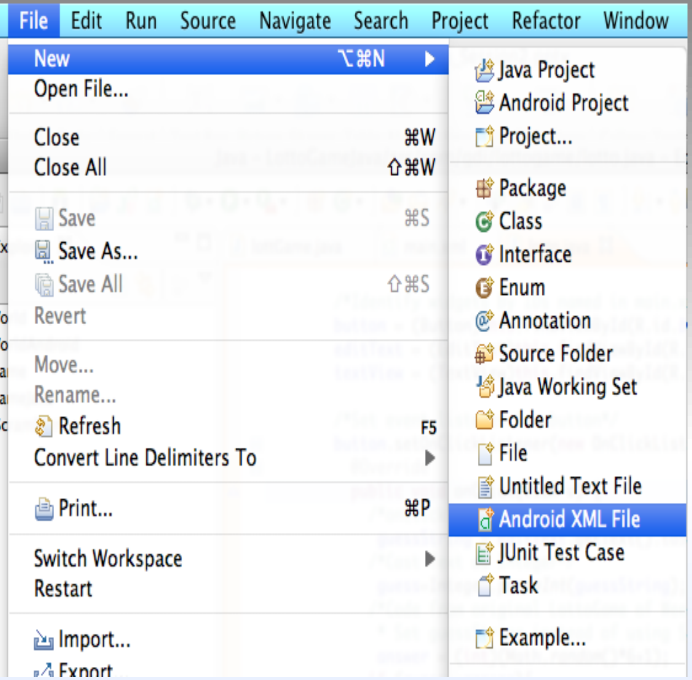
Android Manifest File

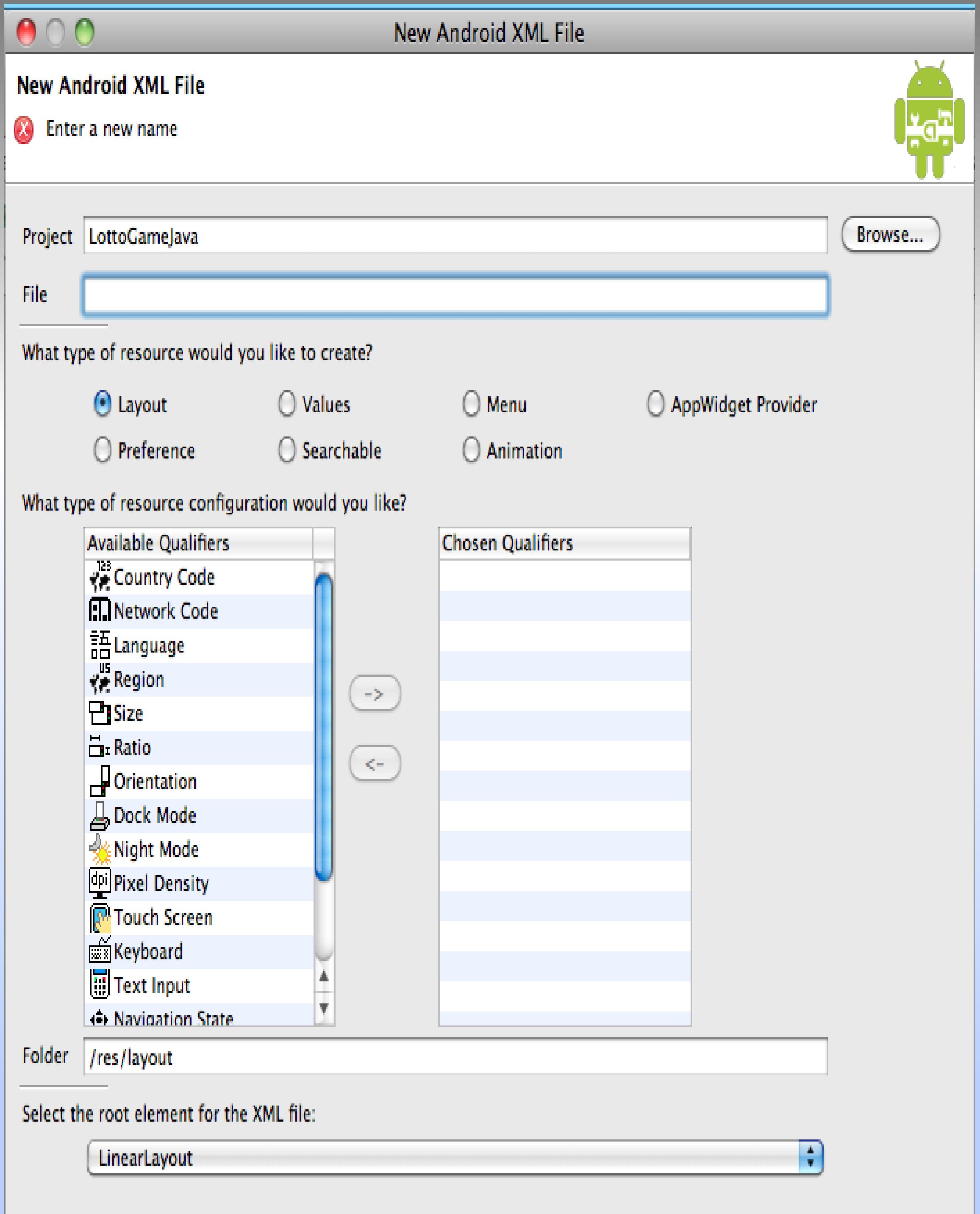
```
<?xml version="1.0" encoding="utf-8" ?>
   <manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.gdi.helloworld"
      android:versionCode="1"
     android:versionName="1.0">
     <uses-sdk android:minSdkVersion="3" />
     <application android:icon="@drawable/icon"
   android:label="@string/app_name">
       <activity android:name=".hello"  android:label="@string/app_name">
        <intent-filter>
          <action android:name="android.intent.action.MAIN" />
          <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
    </activity>
   </application>

</manifest>
```

Layouts

- Absolute Layout
  - pixel by pixel
- Relative Layout
  - relative to other objects
- Table Layout
  - simple table
- Linear Layout
  - either horizontal or vertical

All layouts can be nested inside each other




New Android XML File

New Android XML File

Enter a new name

Project: LottoGameJava

File:

What type of resource would you like to create?

● Layout ○ Values ○ Menu ○ AppWidget Provider
○ Preference ○ Searchable ○ Animation

What type of resource configuration would you like?

Available Qualifiers: Country Code, Network Code, Language, Region, Size, Ratio, Orientation, Dock Mode, Night Mode, Pixel Density, Touch Screen, Keyboard, Text Input, Navigation State

Chosen Qualifiers:

Folder: /res/layout

Select the root element for the XML file: LinearLayout

Showing Information to the User

- Views
  - TextView
  - ImageView
  - SurfaceView
- Attributes and values
  - android:background="*#ffffff*"
  - android:text="*@string/hello*"
  - android:text="*@string/hello*"

Adding an ImageView

```
<ImageView
   android:layout_height="wrap_content"
    android:src="@drawable/android"
   android:id="@+id/imageView1"
   android:layout_width="fill_parent">
</ImageView>
```

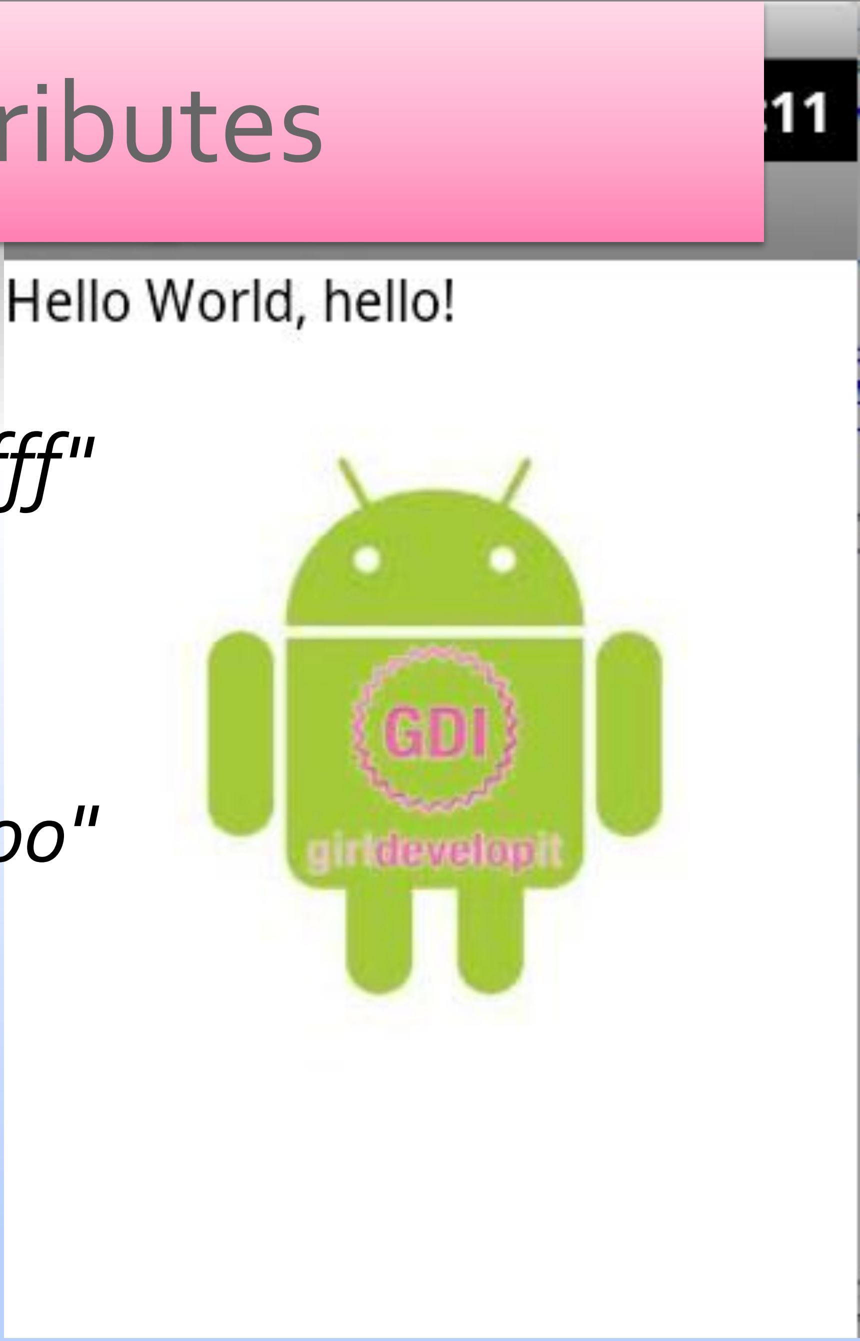
Adding attributes

LinearLayout:

android:background=*"#ffffff"*

TextView:

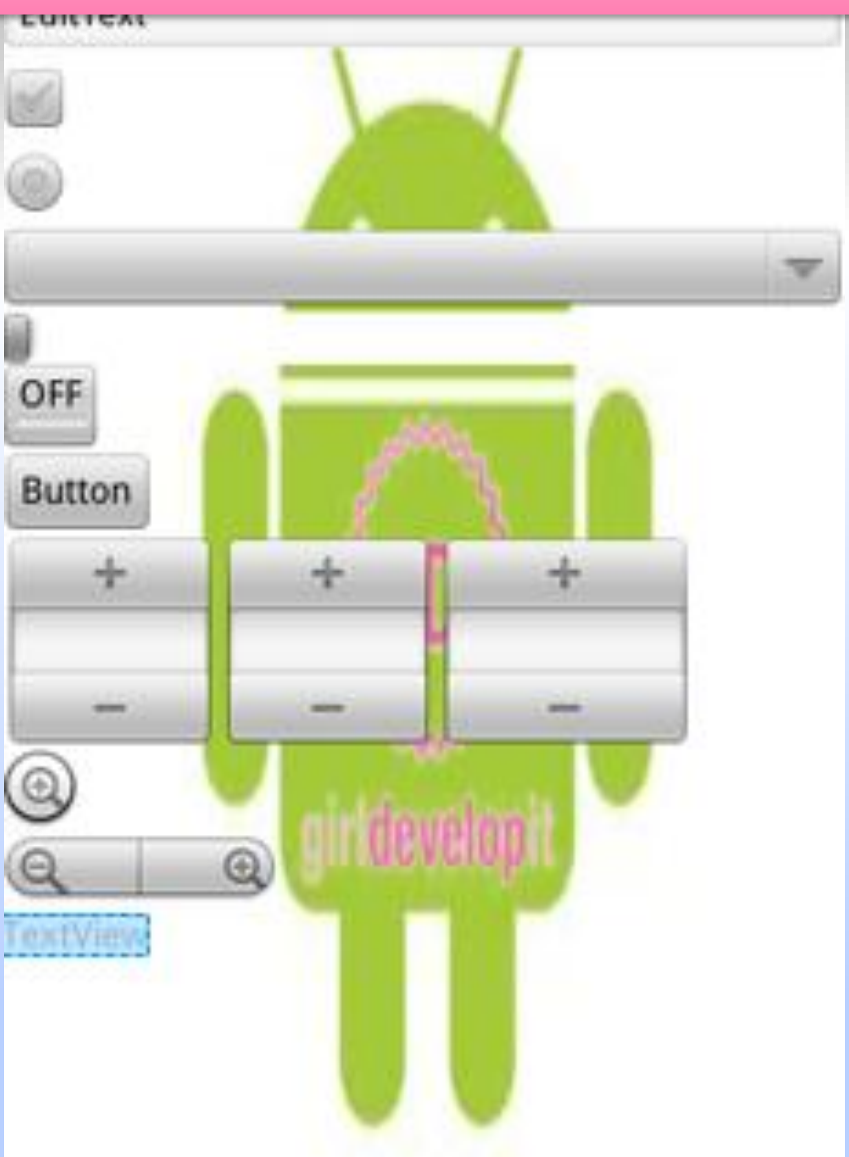
android:textColor=*"#000000"*



Hello World, hello!

Getting data from the user

- EditText
- CheckBox
- RadioButton
- Spinner
- Slider
- ToggleButton
- Button
- DatePicker
- Zoom
- ZoomControl



LottoGame for Android!

- Create new Android Project
- Name it LottoGameAndroid
- package com.gdi.lottogame
- Choose a name for the App and the first class
- Add a TextEdit, a Button, and a TextView
- Add the property to LinearLayout
  - android:background="*@drawable/android*"

Layout for LottoGame

```
<?xml version="1.0" encoding="utf-8"?>
    <LinearLayout
    xmlns:android=http://schemas.android.com/apk/res/android
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/android">
<EditText android:text="EditText"  android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText1"></EditText>
<Button android:text="Button"  android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"></Button>
<TextView android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"></TextView>
</LinearLayout>
```

Connecting to Widgets

Above public class{

```
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
```

Above @Override

```
Button button;
EditText editText;
TextView textView;
```

Event Listeners

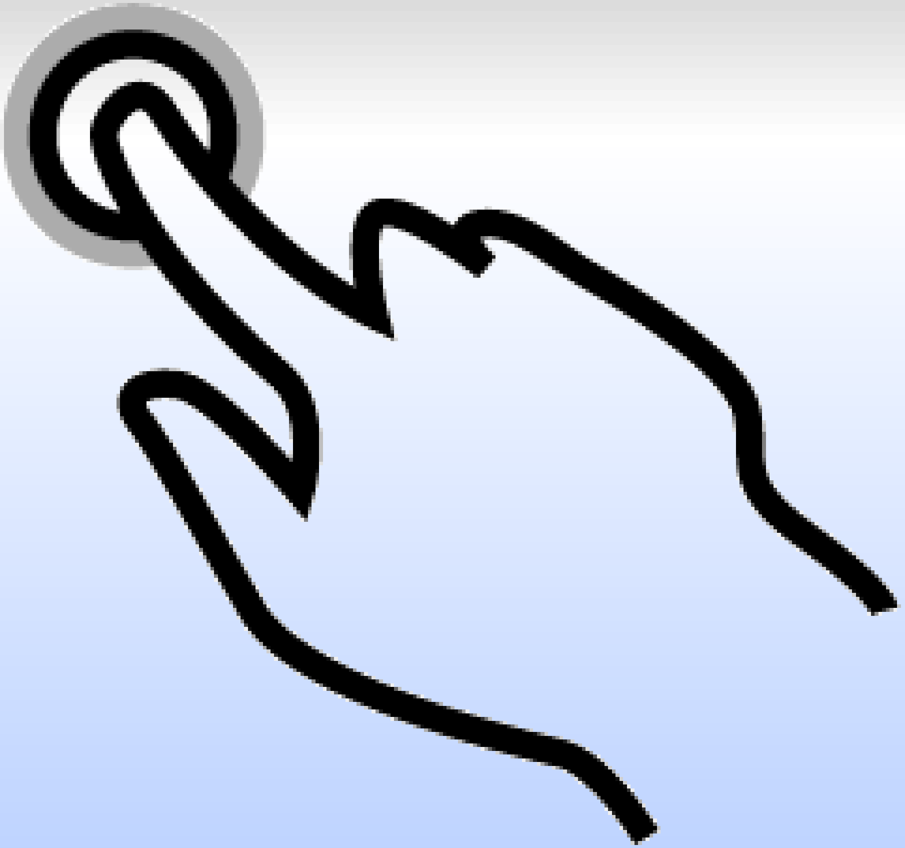
onClick()

onDrag()

onFocus()

onLongClick()

onScale()



Event Listeners for LottoGame

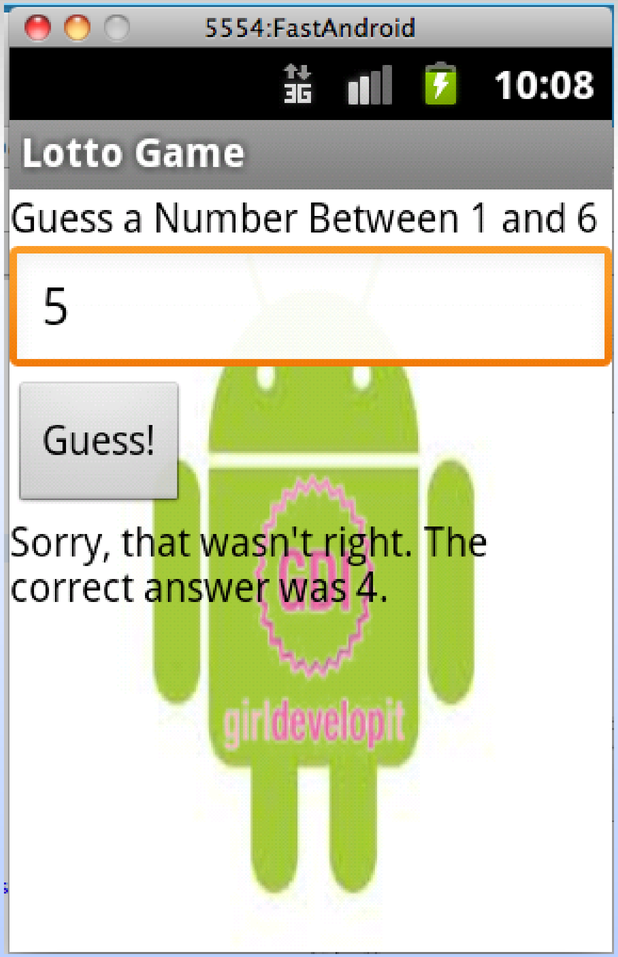
```
import android.view.View;
import android.view.View.OnClickListener;

 button = (Button)this.findViewById(R.id.button1);
      button.setOnClickListener(new OnClickListener() {
       @Override
      public void onClick(View v) {

       }
    });
```

Getting and Setting Text

```
editText = (EditText)this.findViewById(R.id.editText1);
textView = (TextView)this.findViewById(R.id.textView1);

onClick(View v){
guessString = editText.getText().toString();
textView.setText(guessString);

guessString = editText.getText().toString();
guess=Integer.parseInt(guessString);
guess=guess-6;
guessString=Integer.toString(guess);
textView.setText(guessString);
```

Using LottoGame from Last week

```
guessString="Congratulations!
    You guessed right!";

guessString="Sorry, that
    wasn't right. The correct
    answer was " + answer+".";

textView.setText(guessString);
```



Guess a Number Between 1 and 6

5

Sorry, that wasn't right. The correct answer was 4.

Homework

Use what you learned about getting user data today to create a version of StringScrambler for the Android

Bonus: Play with other widgets that get user data

Questions?