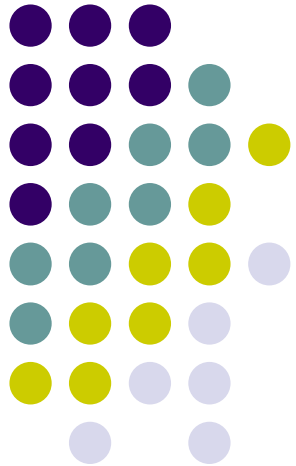
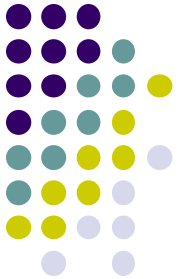
CS 528 Mobile and Ubiquitous Computing

Lecture 2b: WebView

Emmanuel Agu





WebView Widget

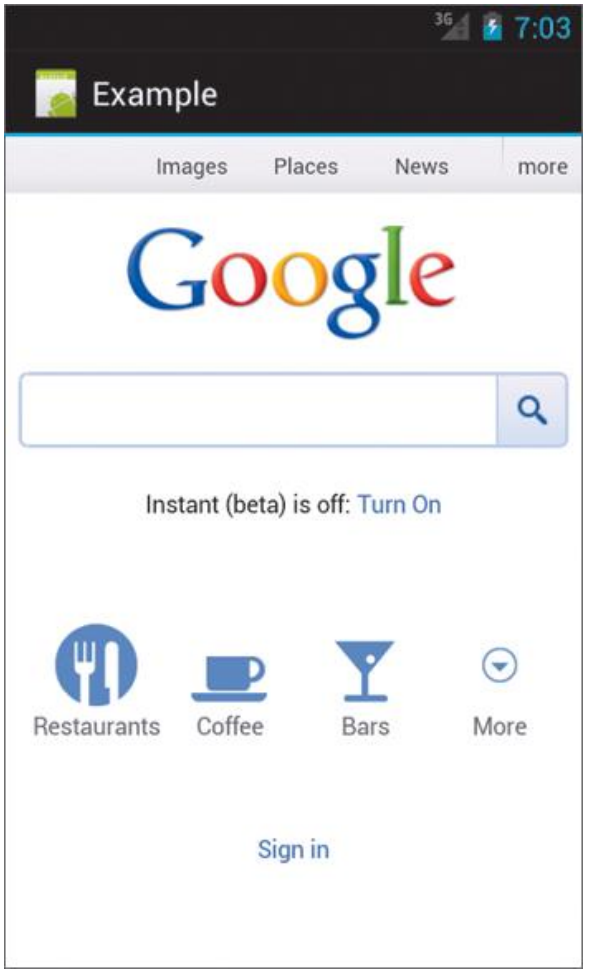
WebView Widget

- A View that displays web pages
  - Can be used for creating your own web browser
  - OR just display some online content inside your app
- Since Android 4.4, webviews rendering using:
  - Chromium open source project, engine used in Google Chrome browser (http://www.chromium.org/)

- Webviews on earlier Android versions supported webkit, which is used many web browsers including Safari

WebView Widget Functionality

- Supports HTML5, CSS3 and JavaScript
- Navigate previous URLs (back and forward)
- zoom in and out
- perform searches
- Can also:
  - Embed images in page
  - Search page for string
  - Deal with cookies



WebView Example

- Simple app to view and navigate web pages
- XML code (e.g in res/layout/main.xml)

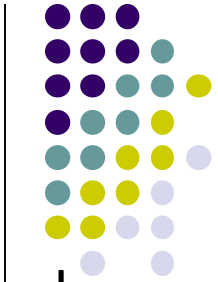
```
<?xml version="1.0" encoding="utf-8"?>
<WebView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
/>
```

WebView Activity

- In onCreate, use loadURL to load website
- If website contains Javascript, enable Javascript
- loadUrl( ) can also load files on Android local filesystem (file://)

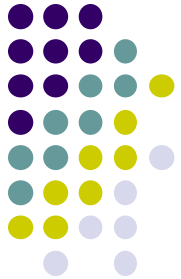
```
public class HelloWebView extends Activity {

    private WebView mWebView;

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

        mWebView = (WebView) findViewById(R.id.webview);
        mWebView.getSettings().setJavaScriptEnabled(true);
        mWebView.loadUrl("http://m.utexas.edu");
    }
}
```

WebView: Request Internet Access



- Request **permission to use Internet** in AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="scottm.examples"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" />

    <uses-permission android:name="android.permission.INTERNET" />
```



Android UI Design Example

GeoQuiz App
Reference: Android Nerd Ranch, pgs 1-32

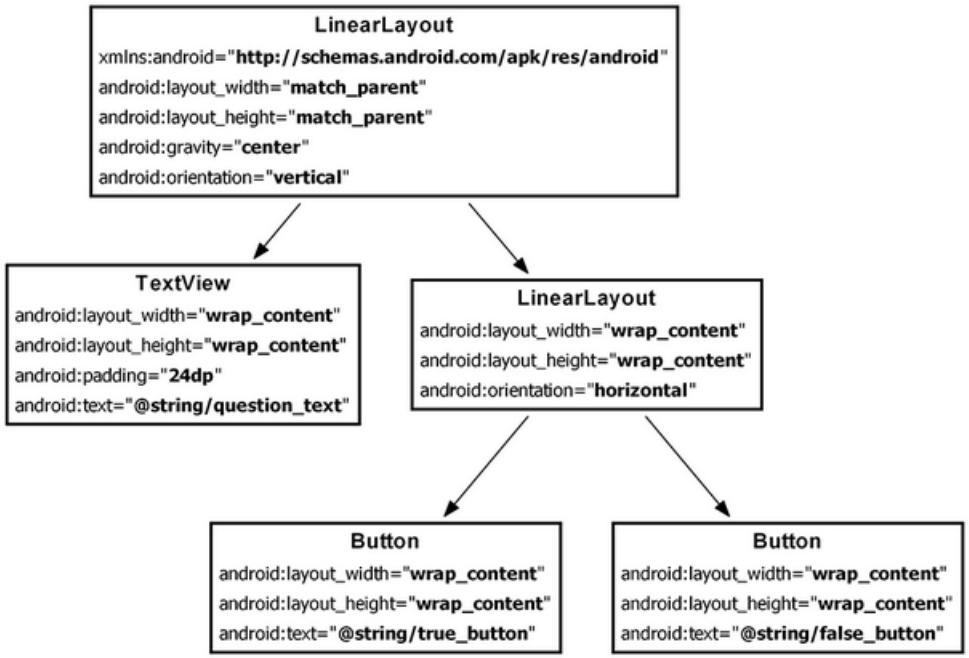
- App presents questions to test user's knowledge of geography
- User answers by pressing **True** or **False** buttons
- How to get this book?

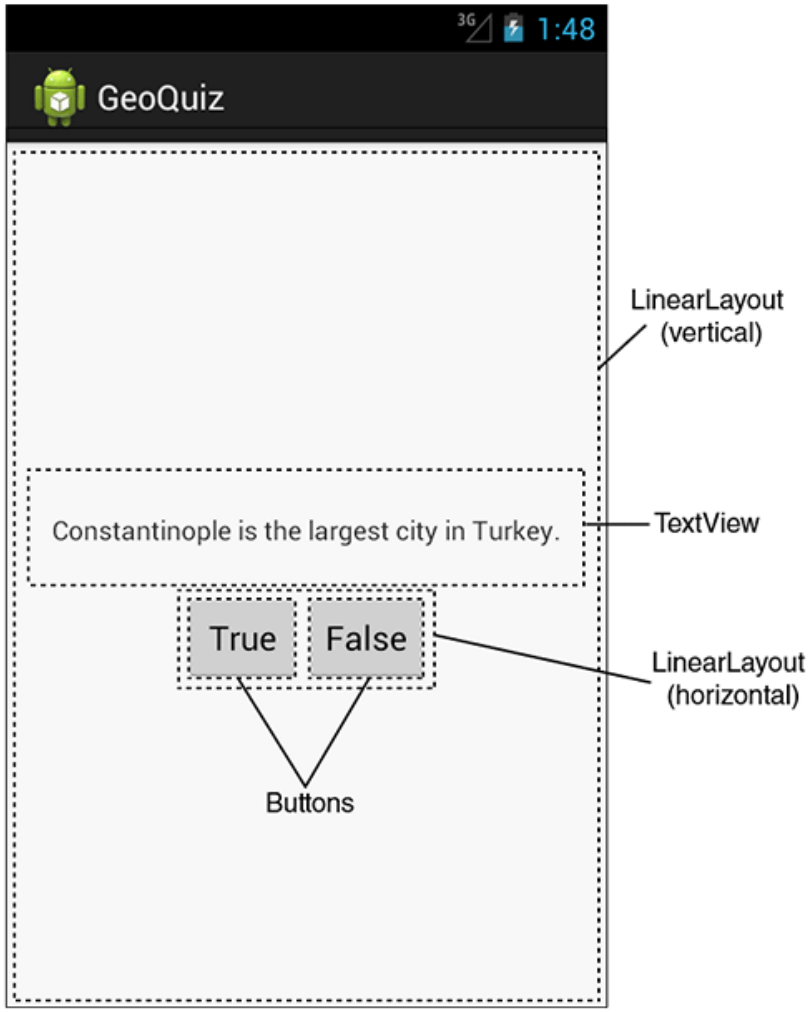
GeoQuiz App

- 2 main files:
  - **activity_quiz.xml:** to format app screen
  - **QuizActivity.java:** To present question, accept True/False response
- **AndroidManifest.xml** lists all app components, auto-generated

GeoQuiz: Plan Out App Widgets

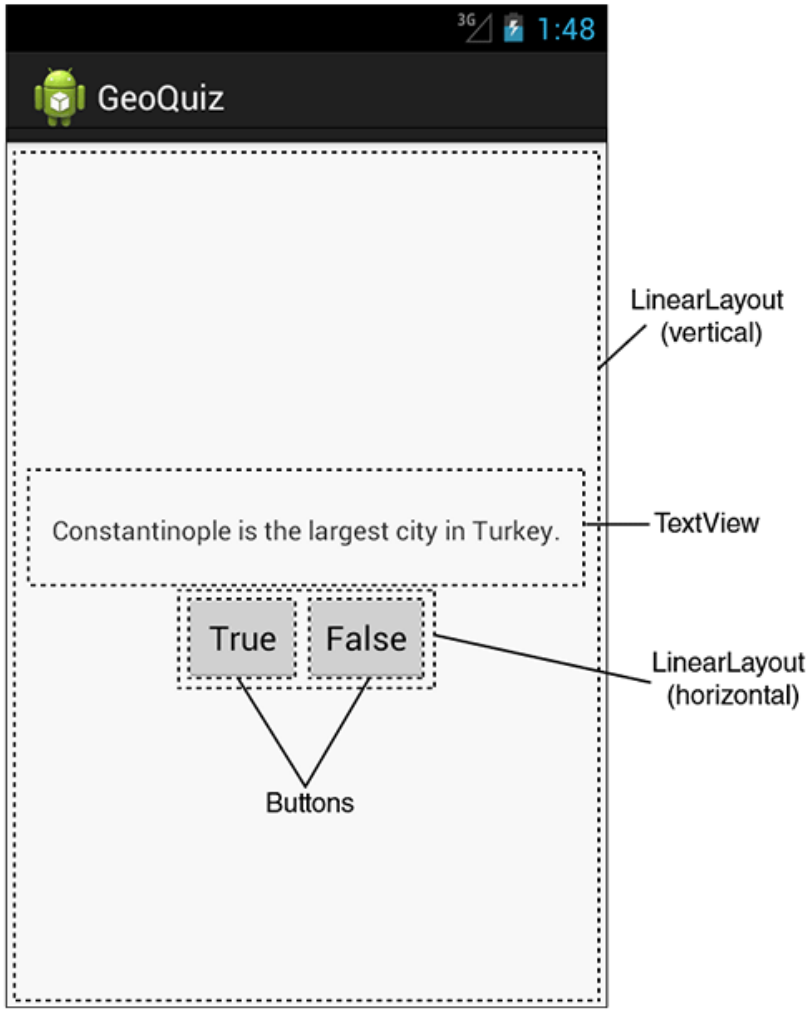
- 5 Widgets arranged hierarchically





GeoQuiz: activity_quiz.xml File listing

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:gravity="center"
  android:orientation="vertical" >

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="24dp"
    android:text="@string/question_text" />

  <LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/true_button" />

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/false_button" />

  </LinearLayout>

</LinearLayout>
```

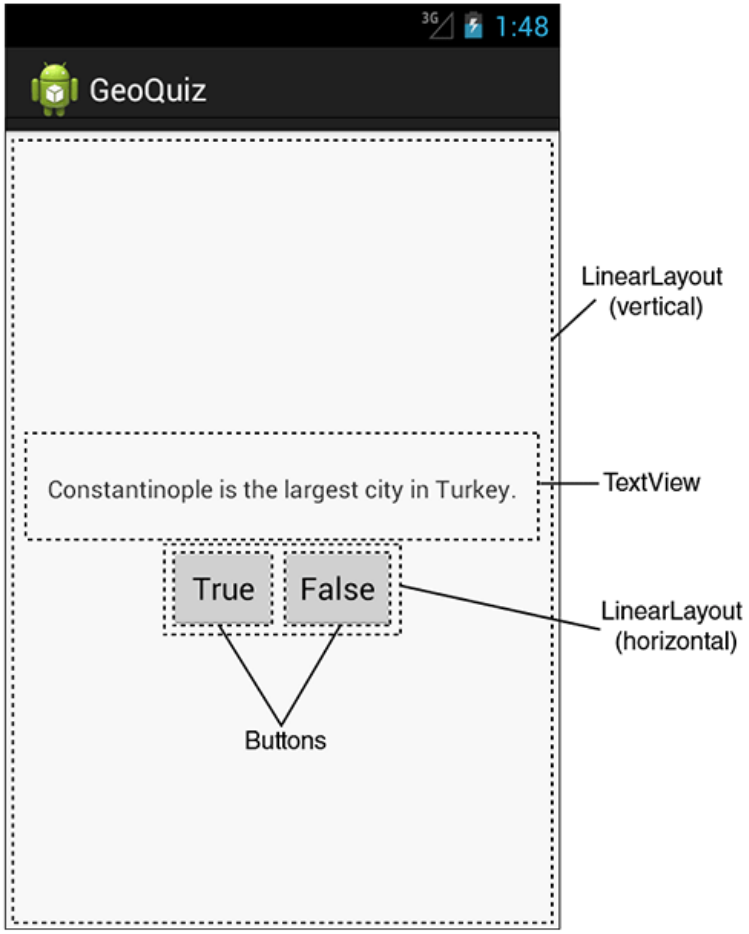


GeoQuiz: strings.xml File listing

- Define all strings app will use
  - Question: “Constantinople is.. “
  - True
  - False

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

  <string name="app_name">GeoQuiz</string>
  <string name="hello_world">Hello, world!</string>
  <string name="question_text">Constantinople is the largest city in
Turkey.</string>
  <string name="true_button">True</string>
  <string name="false_button">False</string>
  <string name="menu_settings">Settings</string>

</resources>
```

strings.xml



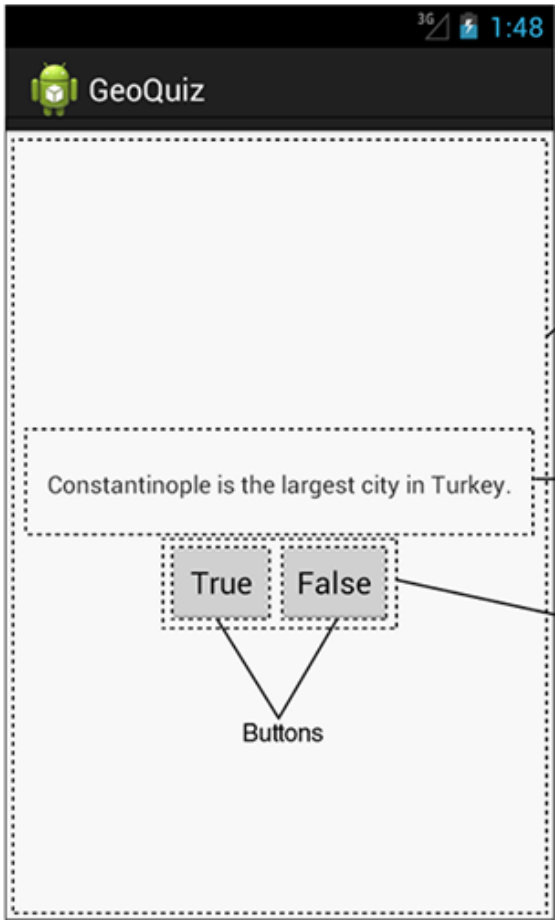
QuizActivity.java

- Initial QuizActivity.java code

```
package com.bignerdranch.android.geoquiz;

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

public class QuizActivity extends Activity {

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

onCreate Method is called once Activity is created

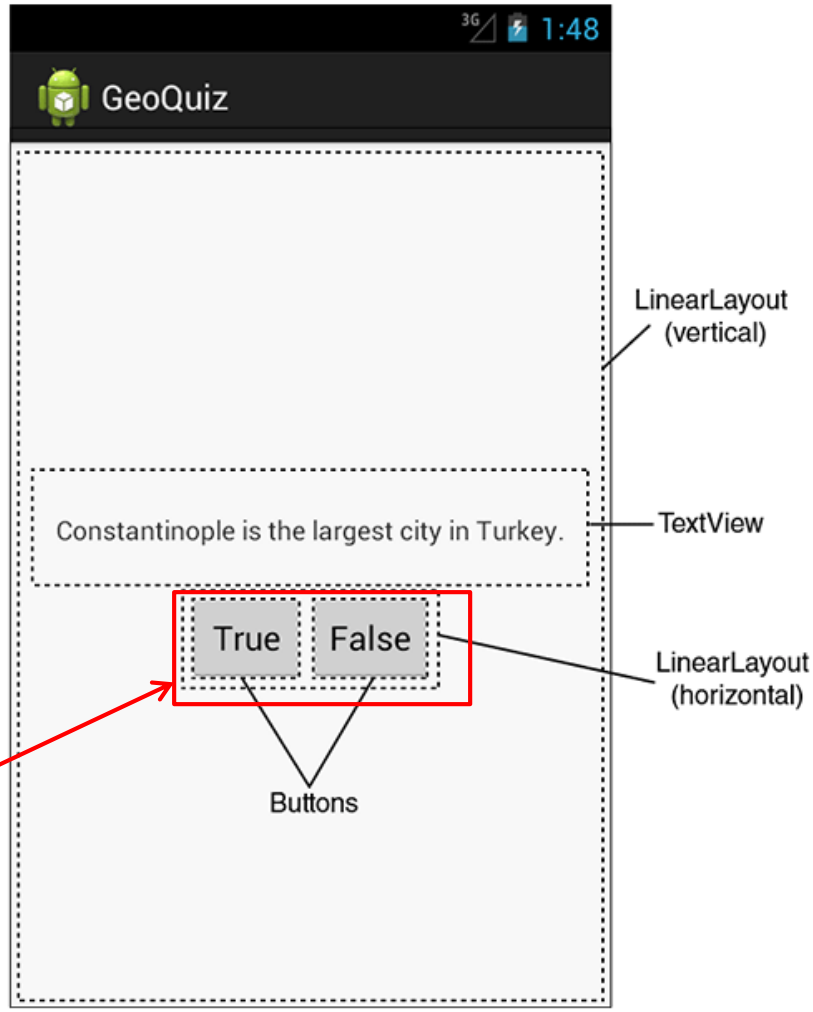
specify layout XML file (**activity_quiz.xml**)



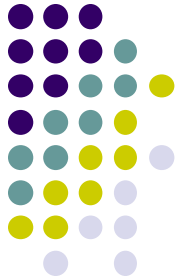
- Would like java code to respond to True/False buttons being clicked

Responding to True/False Buttons in Java

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

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="24dp"
    android:text="@string/question_text" />

  <LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
      android:id="@+id/true_button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/true_button" />

    <Button
      android:id="@+id/false_button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/false_button" />

  </LinearLayout>

</LinearLayout>
```



Write code in Java file to specify app's response when True/False buttons are clicked

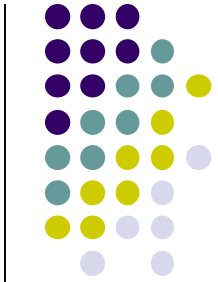


2 Alternative Ways to Respond to Button Clicks

1. In XML: set android:onClick attribute (already seen this!!)

2. In java create a ClickListener object, override onClick method
   - typically done with anonymous inner class

Recall: Approach 1: Responding to Button Clicks



- May want Button press to trigger some action
- How?

1. In XML file (e.g. Activity_my.xml), set android:onClick attribute to specify method to be invoked

2. In Java file (e.g. MainActivity.java) declare method/handler to take desired action

Activity_my.xml

```
<Button
  android:onClick="someMethod"
  ...
/>
```

MainActivity.java

```
public void someMethod(View theButton) {
  // do something useful here
}
```

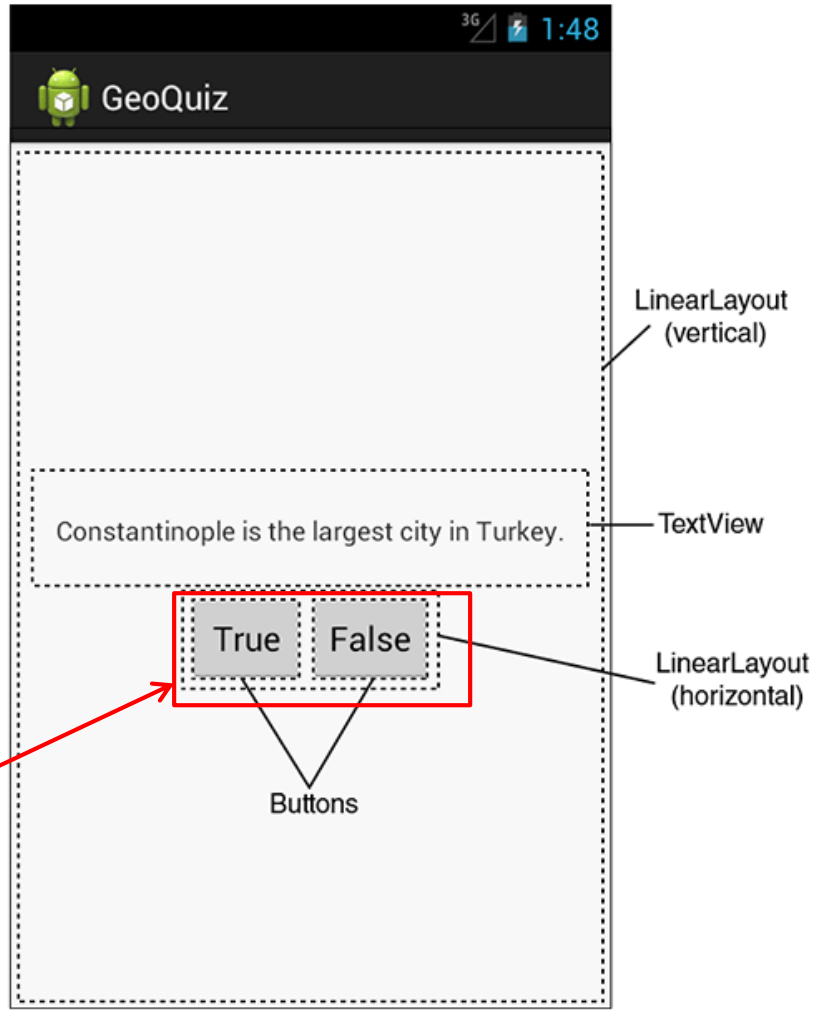
Approach 2: Create a ClickListener object, override onClick

- First, get reference to Button in our Java file. How?

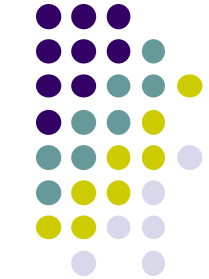
```
<Button
  android:id="@+id/true_button"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/true_button" />
```

```
<Button
  android:id="@+id/false_button"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/false_button" />
```

Need reference to Buttons



R.Java Constants



- During compilation, XML resources (drawables, layouts, strings, views with IDs, etc) are assigned constants
- Sample R.Java file
- In Java file, can refer to these resources using their constants

Interfaces grouping the constants.

Constants referring to XML resource.

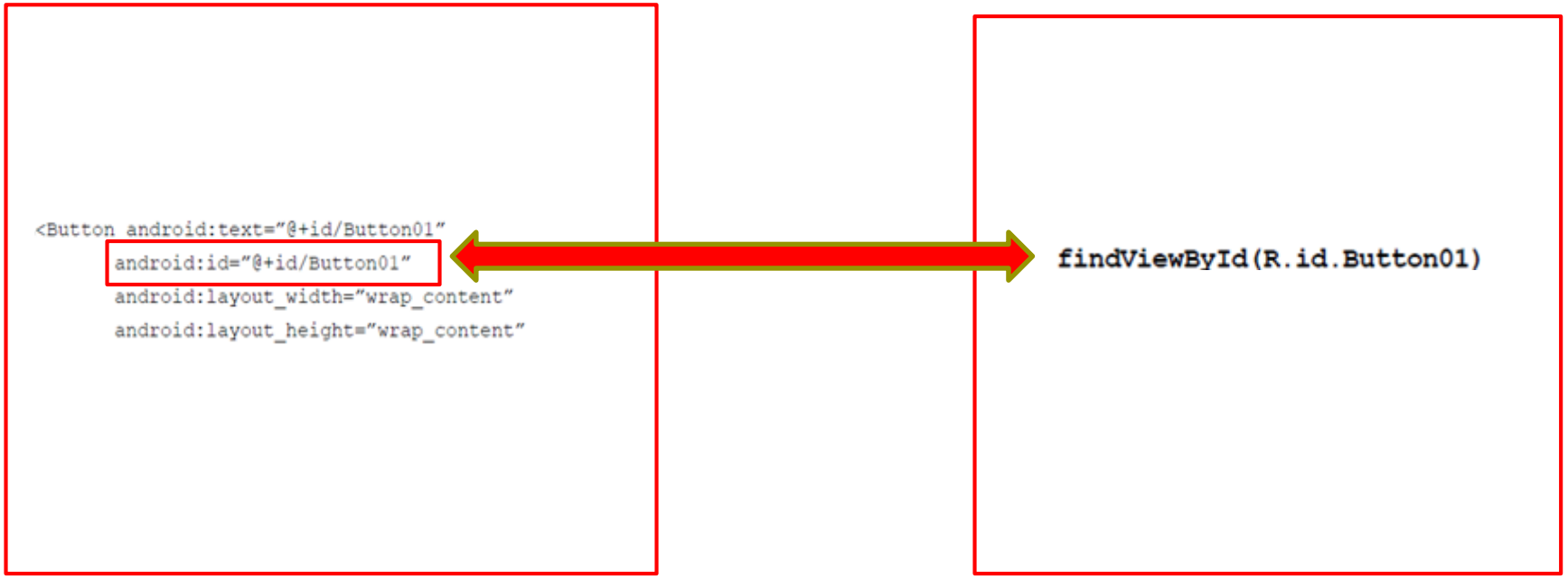
```
public final class R {
        public static final class attr {}
        public static final class drawable {
                public static final int icon=0x7f020000;
        }
        public static final class id {
                public static final int Button01=0x7f050000;
        }
        public static final class layout {
                public static final int main=0x7f030000;
        }
        public static final class string {
                public static final int app_name=0x7f040001;
                public static final int haiku=0x7f040000;
                public static final int love_button_text=0x7f040002;
        }
}
```

Referencing Widgets by ID

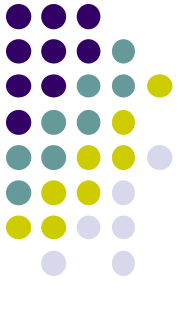
- To reference a widget in Java code, use **findviewbyID** need its **android:id**
- Use **findviewbyID**

In XML file, give the widget/view an ID i.e. assign android:id

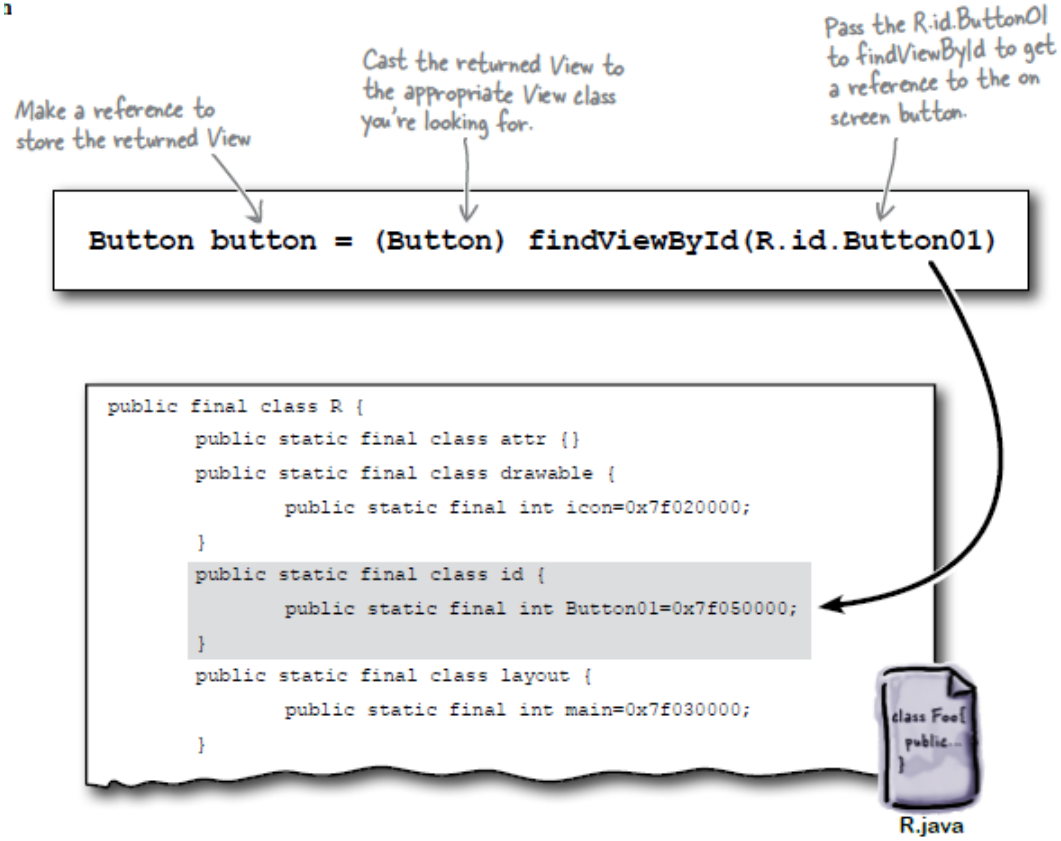
In java file, to reference/manipulate view/widget use its ID to find it (call findviewbyID( ))



Getting View References



- Argument of **findViewById** is constant of resource
- A generic view is returned (not subclasses e.g. buttons, TextView), so needs to cast



QuizActivity.java: Getting References to Buttons

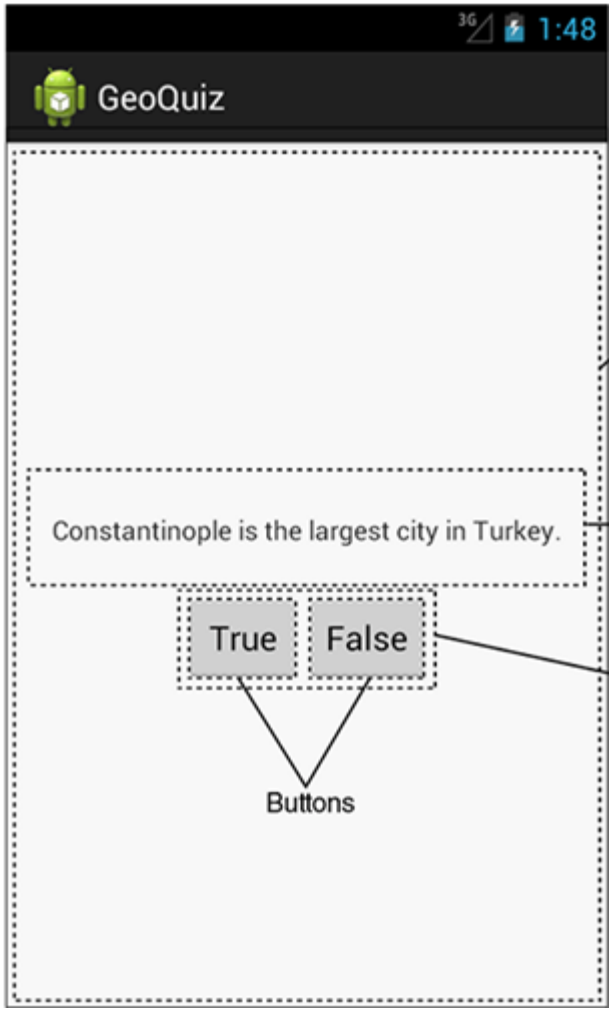
- To get reference to buttons in java code

```
public class QuizActivity extends Activity {

    private Button mTrueButton;
    private Button mFalseButton;

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

        mTrueButton = (Button)findViewById(R.id.true_button);
        mFalseButton = (Button)findViewById(R.id.false_button);
    }

...
}
```

Declaration in XML

```
<Button
  android:id="@+id/true_button"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/true_button" />
```

```
<Button
  android:id="@+id/false_button"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/false_button" />
```



QuizActivity.java: Setting Listeners

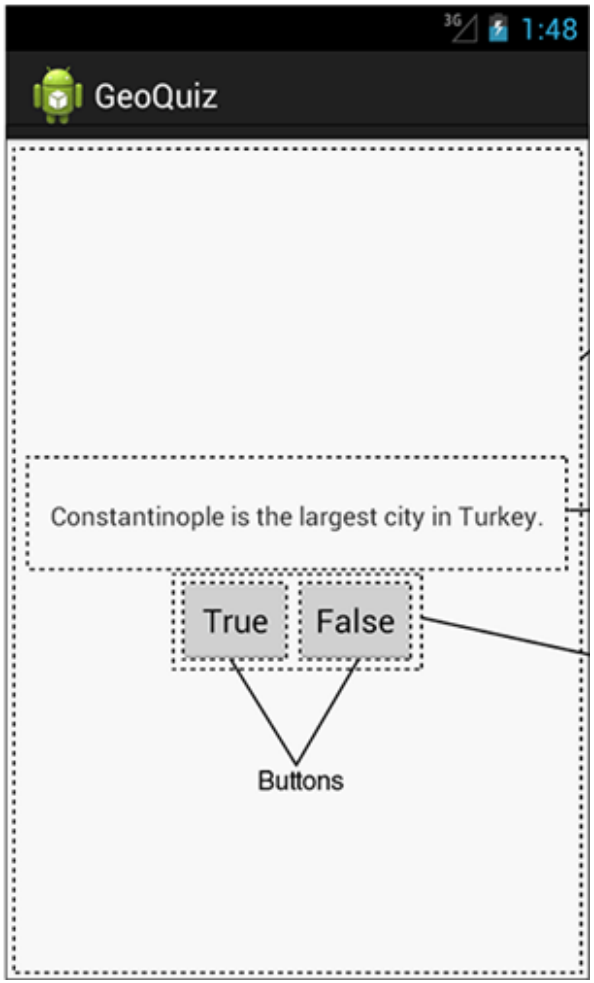
- Set listeners for **True** and **False** button

```
...
    mTrueButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Does nothing yet, but soon!
        }
    });

    mFalseButton = (Button)findViewById(R.id.false_button);
    mFalseButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Does nothing yet, but soon!
        }
    });
}
```

2.Set Listener Object For mTrueButton

3. Overide onClick method (insert your code to do whatever you want as mouse response here)

1. Create listener object as anonymous (unnamed) inner object



QuizActivity.java: Adding a Toast

- A toast is a short pop-up message
- Does not require any input or action
- After user clicks True or False button, our app will pop-up a toast to inform the user if they were right or wrong

- First, we need to add toast strings (Correct, Incorrect) to strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">GeoQuiz</string>
  <string name="question_text">Constantinople is the largest city in
Turkey.</string>
  <string name="true_button">True</string>
  <string name="false_button">False</string>
  <string name="correct_toast">Correct!</string>
  <string name="incorrect_toast">Incorrect!</string>
  <string name="menu_settings">Settings</string>
</resources>
```

QuizActivity.java: Adding a Toast

- To create a toast, call the method:

```
public static Toast makeText(Context context, int resId, int duration)
```

Instance of Activity (Activity is a subclass of context)

Resouce ID of the string that toast should display

Constant to specifiy how long toast should be visible

- After creating toast, call **toast.show( )** to display it
- For example to add a toast to our onClick( ) method:

```
public void onClick(View v) {
    Toast.makeText(QuizActivity.this,
                   R.string.incorrect_toast,
                   Toast.LENGTH_SHORT).show();
}
```

QuizActivity.java: Adding a Toast

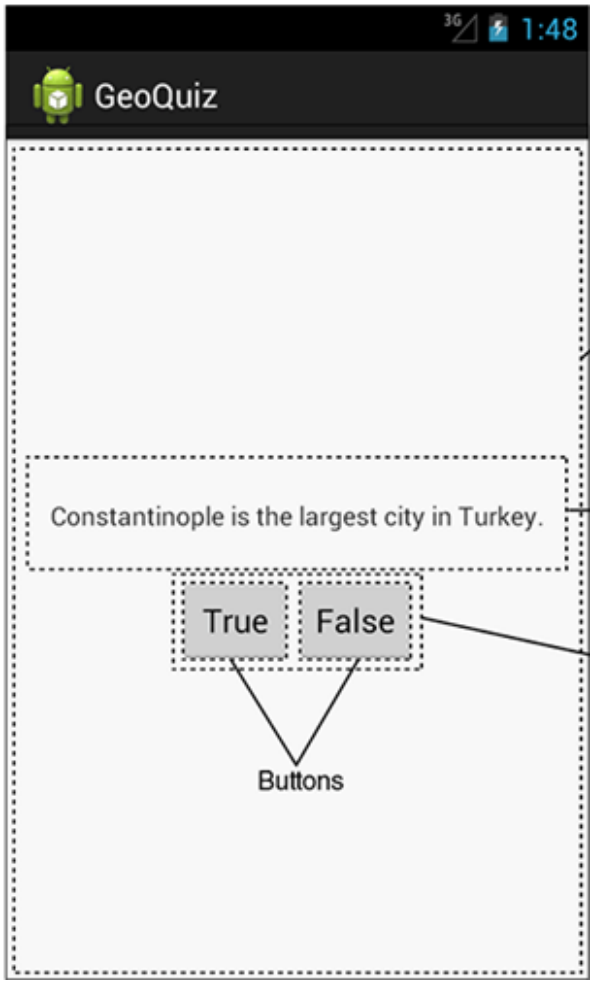
- Code for adding a toast

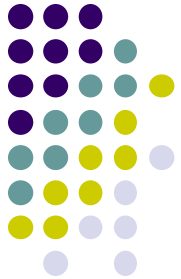
```
...
mTrueButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(QuizActivity.this,
                       R.string.incorrect_toast,
                       Toast.LENGTH_SHORT).show();
    }
});

mFalseButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(QuizActivity.this,
                       R.string.correct_toast,
                       Toast.LENGTH_SHORT).show();
    }
});
```

2.Set Listener Object For mTrueButton

3. Overide onClick method Make a toast

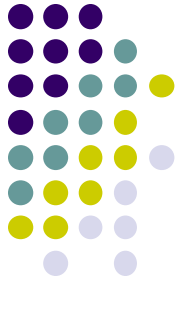
1. Create listener object as anonymous innner object





QuizActivity.java: Complete Listing

```
package com.bignerdranch.android.geoquiz;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class QuizActivity extends Activity {

    Button mTrueButton;
    Button mFalseButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz);

        mTrueButton = (Button)findViewById(R.id.true_button);
        mTrueButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(QuizActivity.this,
                        R.string.incorrect_toast, Toast.LENGTH_SHORT)
                        .show();
            }
        });
```

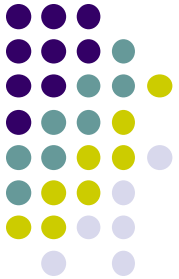


```
    mFalseButton = (Button)findViewById(R.id.false_button);
    mFalseButton.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            Toast.makeText(QuizActivity.this,
                    R.string.correct_toast, Toast.LENGTH_SHORT)
                    .show();
        }
    });
}
```

QuizActivity.java: Complete Listing (Contd)

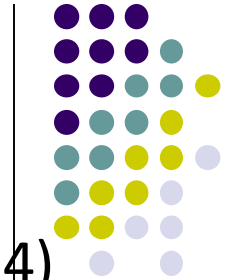
```
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

     // Inflate the menu;
     // this adds items to the action bar if it is present.

        getMenuInflater().inflate(R.menu.activity_quiz, menu);
        return true;
    }

}
```

Used if app has an Action bar menu

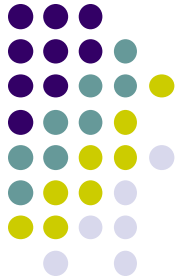


Quiz 1

Quiz 1



- Quiz in class next Thursday (first 20 mins of class Thurs, 9/14)
- Short answer questions
- Try to focus on understanding, not memorization
- Covers:
  - Lecture slides for lectures 1a, 1b, 2a and 2b
  - YouTube Tutorials (from thenewboston) 1-8, 11,12, 17
  - 3 code examples from books
    - **HFAD examples:** myFirstApp, Beer Advisor
    - **ANR example:** geoQuiz

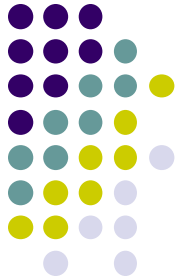


EML: Cooperative Based Groups

EML: Cooperative Based Groups

- Japanese students visiting Boston for 2 week vacation
- Speak little English, need help to find
  - Attractions to visit, where to stay (cheap, central), meet Americans, getting around, eat (Japanese, some Boston food), weather info, events, ….. anything
- Your task: Search android market for helpful apps (6 mins)
  - **Location-aware:** 5 points
  - **Ubicomp (e.g. uses sensor) or smartwatch:** 10 points
- Also **IoT** devices they can buy that would help them

References



- Busy Coder's guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014
- Android App Development for Beginners videos by Bucky Roberts (thenewboston)
- Head First Android
- Android Nerd Ranch, Third Edition