A presentation slide with a dark blue background and a lighter blue wavy header. The text is centered and includes the lecture title, course name, and instructor information.

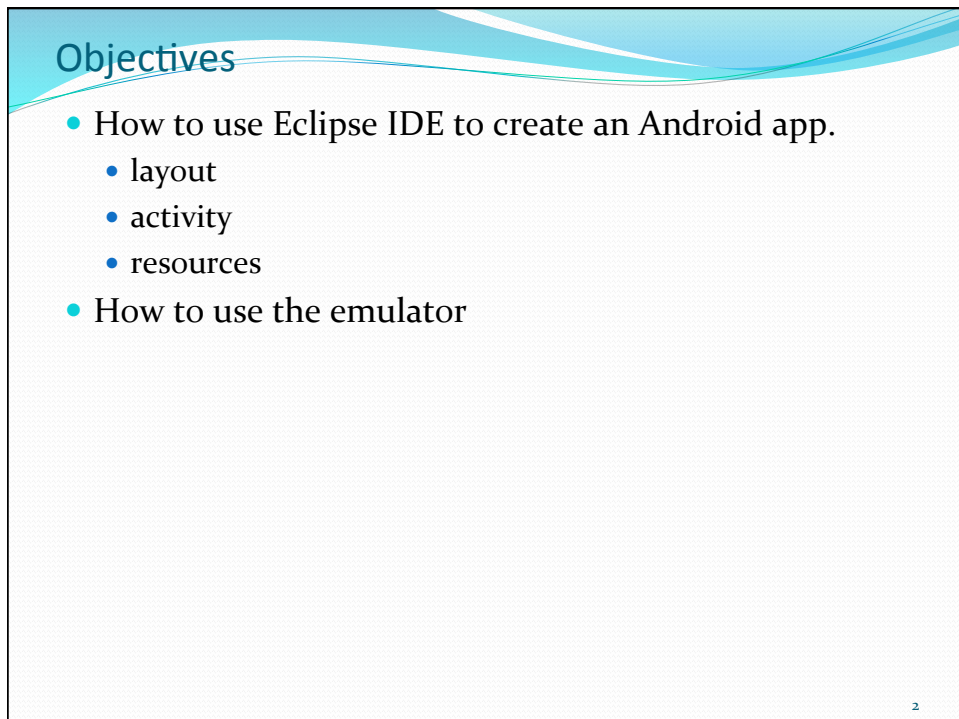
Lecture 01

First Android App

CMSC 4303/5303 Mobile Apps Programming

Hong K. Sung, Ph.D.
Department of Computer Science
University of Central Oklahoma

1

A presentation slide with a light blue wavy header and a white background with a fine grid pattern. The title 'Objectives' is in the header, and a bulleted list of objectives is on the main body.

Objectives

- How to use Eclipse IDE to create an Android app.
 - layout
 - activity
 - resources
- How to use the emulator

2

GeoQuiz app

- The user presses True or False to answer the question on screen, and GeoQuiz provides instant feedback.

Figure 1.1 (It's Istanbul, not Constantinople)



3

App Basics

- An **activity** is an instance of **Activity**, a class in the Android SDK. An activity is responsible for managing user interaction with a screen of information.

You write subclasses of **Activity** to implement the functionality that your app requires. A simple application may need only one subclass; a complex application can have many.

GeoQuiz is a simple app, so it will have a single **Activity** subclass named **QuizActivity**. **QuizActivity** will manage the user interface shown in Figure 1.1.

- A **layout** defines a set of user interface objects and their position on the screen. A layout is made up of definitions written in XML. Each definition is used to create an object that appears on screen, like a button or some text.

Figure 1.2 **QuizActivity** manages what `activity_quiz.xml` defines



4

Creating an Android project

- File → New → Android Application Project

New Android Application
Creates a new Android Application

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme:

Choose the base theme to use for the application

The package name must be a unique identifier for your application. It is typically not shown to users, but it "must" stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the "same app". This is typically the reverse domain name of your organization plus one or more application identifiers, and it must be a valid Java package name.

5

Creating an Android project

New Android Application
Configure Project

☐ Create custom launcher icon

☒ Create activity

☐ Mark this project as a library

☒ Create Project in Workspace

Location:

Working sets

☐ Add project to working sets

Working sets:

New Android Application
Create Activity

Select whether to create an activity, and if so, what kind of activity.

☒ Create Activity

☒ Blank Activity

☐ Fullscreen Activity

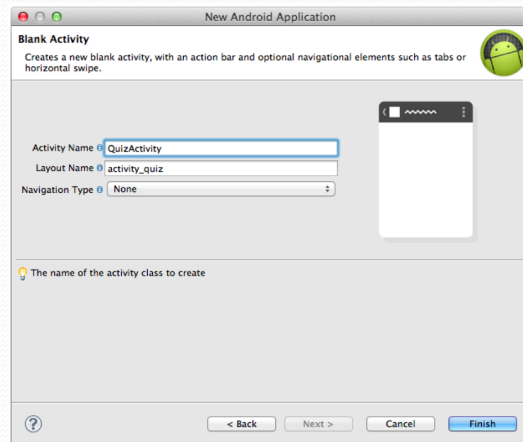
☐ Master/Detail Flow

Blank Activity
Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

6

Creating an Android project

- Naming convention (not required, though)
 - **XxxxxActivity** for the activity subclass
 - **activity_xxxx** for layout (note: lowercase, reverse)



7

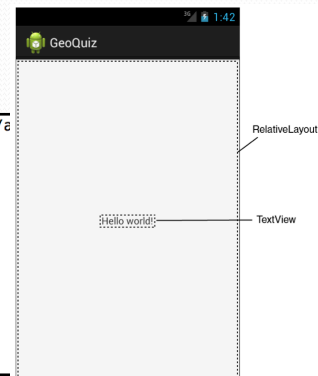
UI layout

- Widgets: the building blocks to compose a UI
 - A widget can show text or graphics, interact with the user, or arrange other widgets on the screen.
 - Buttons, text input controls, check boxes, etc, are all types of widgets.
 - Every widget is an instance of the View class or its subclasses.
 - Two widgets in the default activity layout: **RelativeLayout** and **TextView**.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/a
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".QuizActivity" >

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="@string/hello_world" />

</RelativeLayout>
```



Layout for GeoQuiz

```
<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" >

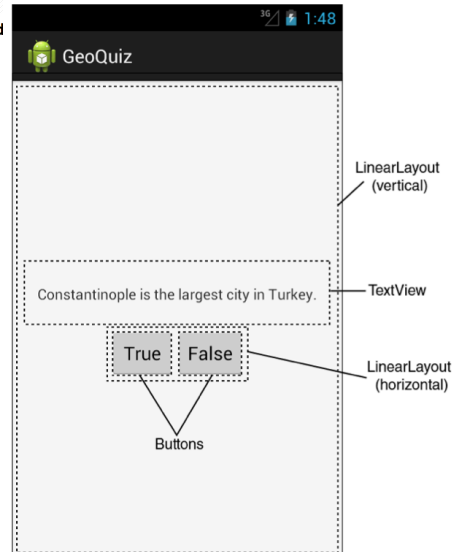
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="24dp"
        android:text="@string/question_text" />

    <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" />

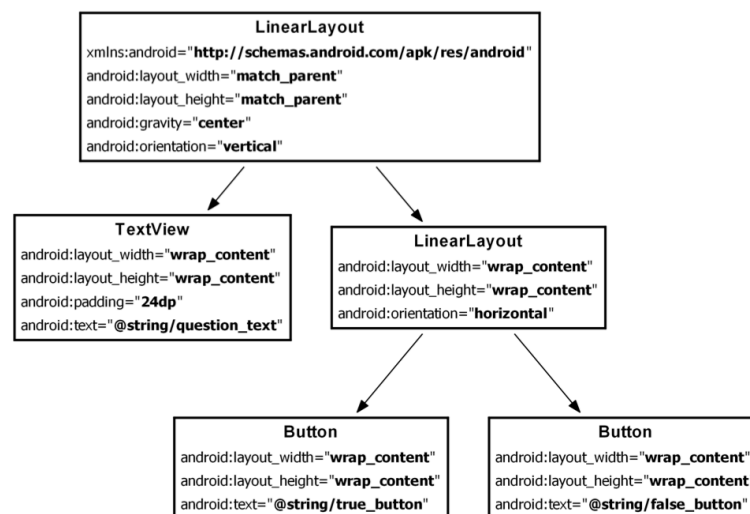
    </LinearLayout>
</LinearLayout>
```



9

the view hierarchy

Figure 1.10 Hierarchical layout of widgets and attributes



10

Widget Attributes

- **android:layout_width, android:layout_height**
 - match_parent: view will be as big as its parent
 - wrap_content: view will be as big as its contents require
 - fill_parent: deprecated (= match_parent)
- **android:orientation**
 - vertical: the first child will appear topmost
 - horizontal: the first child will appear leftmost
- **android:text**
 - this attributes tells the widget what text to display.
 - the values could be literal strings or references to *string resources*
 - a *string resource* is a string that lives in a separate XML file.

11

Creating string resources

- Every project includes a default strings file: res/values/strings.xml
- Do not delete the menu_settings string. It will cause cascading errors in other files related to the menu.

Listing 1.3 Adding string resources (strings.xml)

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

  <string name="app_name">GeoQuiz</string>
  <del><string name="hello_world">Hello, world!</string></del>
  <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>
```

12

From Layout XML to View objects

- The **onCreate(Bundle)** method is called when an instance of the activity subclass is created.
- The **setContentView(int layoutResID)** method inflates a layout and puts it on screen.

```
public class QuizActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_quiz, menu);
        return true;
    }

}
```

13

Resources

- A resource is a piece of application that is not code
 - e.g., image files, audio/video files, XML files, etc
- Resources are in the **res** subdirectory.
 - **res/layout** for layout resources
 - **res/values** for string resources
- To access a resource in code, you use its resource ID automatically created in **R.java**
 - Do not manually edit R.java
 - e.g., **R.layout.activity_quiz** : the resource ID for GeoQuiz layout.

14

Listing 1.5 Current GeoQuiz resource IDs (R.java)

```

/* AUTO-GENERATED FILE. DO NOT MODIFY.
...
*/

package com.bignerdranch.android.geoquiz;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int menu_settings=0x7f070003;
    }
    public static final class layout {
        public static final int activity_quiz=0x7f030000;
    }
    public static final class menu {
        public static final int activity_quiz=0x7f060000;
    }
    public static final class string {
        public static final int app_name=0x7f040000;
        public static final int false_button=0x7f040003;
        public static final int menu_settings=0x7f040006;
        public static final int question_text=0x7f040001;
        public static final int true_button=0x7f040002;
    }
    ...
}

```

15

Listing 1.6 Adding IDs to buttons (activity_quiz.xml)

```

<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>

```

The + sign in **android:id** is for creating the IDs.
No + sign in **android:text** is for referencing the IDs.

16

Listing 1.9 Getting references to widgets (QuizActivity.java)

```
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);
    }

    ...
}
```

17

Listing 1.10 Set listener for True button (QuizActivity.java)

```
...

@Override
public 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) {
            // Does nothing yet, but soon!
        }
    });

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

Anonymous inner class

18

Toast

- A toast is a short message that informs the user of something but does not require any input or action.

Listing 1.13 Making toasts (QuizActivity.java)

```
...
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();
    }
});
```

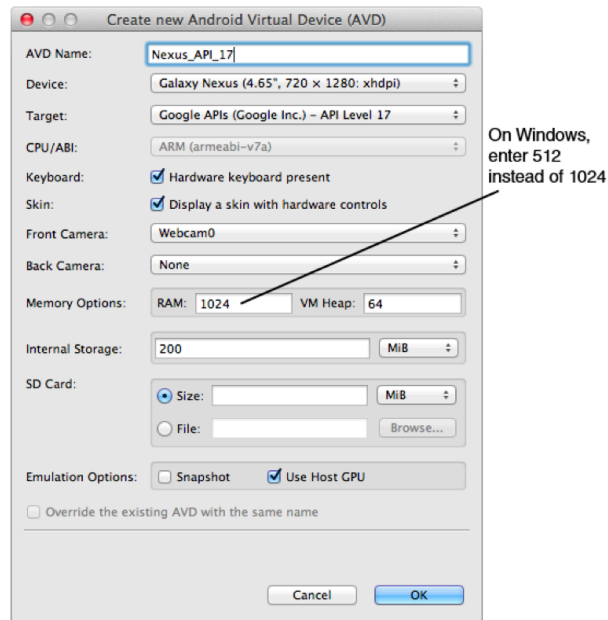
19

Running on the emulator

- To run an Android app, you need either a virtual device (emulator) or an actual device (smartphone, tablet).
 - Starting up the emulator can take a while (**several minutes** depending on your computer). Once it is started, you may quickly deploy other apps.
 - The actual device is much faster in deploying the app.
- To create an Android virtual device (AVD)
 - In Eclipse: Window → Android Virtual Device Manager
- Once you have an AVD:
 - right-click on the GeoQuiz project folder.
 - from the context menu, choose Run As → Android Application.

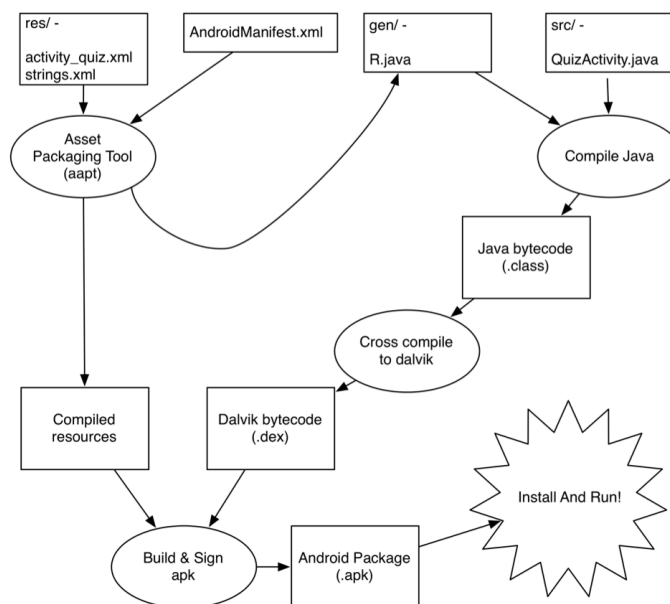
20

Figure 1.13 Creating a new AVD



21

Figure 1.15 Building GeoQuiz



22

Figure 1.16 Inflating activity_quiz.xml

