***Notes on “Android Workshop” 3***

**Step 3: Saving State Data and Alternate Layouts**



Rotating the device destroys current activity and creates a new one. Hence, the fields are initialized and onCreate is called again. That causes the questions to reset to the first one.

* First provide an **alternate layout** that will kick in automatically if the orientation changes to landscape mode. Create a new folder in **res** named (exactly) **layout-land**. Copy activity\_quiz.xml into that folder and adjust it so that the next button is in its own row, centered right. This has nothing to do with fixing our bug, but it seems a good opportunity to talk about alternate layouts in landscape mode. Note that all elements must be present in both layouts with the same name. Just their locations can be different.
* To save data across a runtime configuration change (such as changing the orientation), override void onSaveInstanceState(Bundle state) to write the current value of currentIndex to the bundle. That method is automatically called before onPause, onStop, and onDestroy.
* Read the value back from the bundle in onCreate if that bundle exists.

To write values to a bundle, use methods like *bundle.putInt(key, value)* where *key* is a string identifying the value stored and *value* is in this case the integer to be stored.

To read values back from a bundle, first check that it is not null. Then use methods like *bundle.getInt(key, defaultValue)* to retrieve a value from the bundle. If the key does not exist, return the *defaultValue*.

BUG: Since I chose to enable/disable the next button, I also need to save the state of the Next button to the saved bundle. Left as an exercise.

* Typically you override *onSaveInstanceState* to save the complete app state across configuration changes. The method *onPause* can be overridden to perform other activities such as stopping animations, background data loading, or music playing.

Note that the bundle gets completely destroyed if the user presses the “back” button! To verify that:

* Implement the methods onPause and onDestroy to show simple messages (toasts)

package org.mathcs.geoquiz;

import android.os.Bundle;

import android.app.Activity;

import android.view.Menu;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

public class QuizActivity extends Activity

{

 private final static String KEY\_INDEX = "index";

 private final static String KEY\_STATE\_NEXT = "next\_button\_state";

 private final static TrueFalse[] QUESTIONS = new TrueFalse[]

 {

 new TrueFalse(R.string.question\_africa, false),

 new TrueFalse(R.string.question\_asia, true),

 new TrueFalse(R.string.question\_oceans, true)

 };

 private int currentQuestion = -1; // so that first call to "nextQuestion" works

 private Button buttonTrue = null;

 private Button buttonFalse = null;

 private Button buttonNext = null;

 private TextView question = null;

 private class ButtonListener implements View.OnClickListener

 {

 @Override

 public void onClick(View v)

 {

 if (v.getId() == R.id.id\_button\_false)

 checkAnswer(false);

 else if (v.getId() == R.id.id\_button\_true)

 checkAnswer(true);

 }

 }

 private ButtonListener buttonHandler = null;

 @Override

 protected void onCreate(Bundle savedInstanceState)

 {

 super.onCreate(savedInstanceState);

 setContentView(R.layout.activity\_quiz);

 question = (TextView)this.findViewById(R.id.id\_question\_text);

 buttonFalse = (Button)this.findViewById(R.id.id\_button\_false);

 buttonTrue = (Button)this.findViewById(R.id.id\_button\_true);

 buttonNext = (Button)this.findViewById(R.id.id\_button\_next);

 buttonHandler = new ButtonListener();

 if (savedInstanceState != null)

 {

 currentQuestion = savedInstanceState.getInt(KEY\_INDEX, 0) - 1;

 // This won't work. Why not and how to fix it??

 buttonNext.setEnabled(savedInstanceState.getBoolean(KEY\_STATE\_NEXT,

 true));

 }

 buttonFalse.setOnClickListener(buttonHandler);

 buttonTrue.setOnClickListener(buttonHandler);

 buttonNext.setOnClickListener(

 new View.OnClickListener()

 {

 @Override

 public void onClick(View v)

 {

 nextQuestion();

 }

 });

 nextQuestion();

 }

 @Override

 public boolean onCreateOptionsMenu(Menu menu)

 {

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

 getMenuInflater().inflate(R.menu.quiz, menu);

 return true;

 }

 @Override

 protected void onSaveInstanceState(Bundle state)

 {

 super.onSaveInstanceState(state);

 state.putInt(KEY\_INDEX, currentQuestion);

 state.putBoolean(KEY\_STATE\_NEXT, buttonNext.isEnabled());

 }

 @Override

 **public** **void** onPause()

 {

 **super**.onPause();

 Toast.*makeText*(**this**, "GeoQuiz paused", Toast.*LENGTH\_SHORT*).show();

 }

 @Override

 **public** **void** onDestroy()

 {

 **super**.onDestroy();

 Toast.*makeText*(**this**, "GeoQuiz destroyed", Toast.*LENGTH\_SHORT*).show();

 }

 private void nextQuestion()

 {

 buttonNext.setEnabled(false);

 currentQuestion++;

 if (currentQuestion >= QUESTIONS.length)

 currentQuestion = 0;

 question.setText(QUESTIONS[currentQuestion].getQuestion());

 }

 private void checkAnswer(boolean answer)

 {

 buttonNext.setEnabled(true);

 if (QUESTIONS[currentQuestion].isTrue() == answer)

 Toast.makeText(this, R.string.toast\_correct, Toast.LENGTH\_SHORT).show();

 else

 Toast.makeText(this, R.string.toast\_incorrect, Toast.LENGTH\_SHORT).show();

 }

}

**Step 4: Adding a Second Activity**

Now we want to add an additional activity to our project. We want to add a cheat button to bring up a second activity. In that activity the user can choose to reveal the right answer or not and return to the previous activity with the current question. You can now answer the question correctly, but when you do that having cheated the answer you’ll get is “cheater”. If you did not look up the answer, the response should be just as before.

In other words, we need to:

* Start a second activity
* Pass data along to that second activity
* When the second activity quits, it passes information back to the calling app
* The original method should now take over again

One project can consist of many activities, each of which could be called from another one, or even from other apps. One activity, though, is marked as “special” in the *manifest file* to be the default activity.

* Copy the activity\_quiz.xml file to activity\_cheat.xml in the layout folder
* Remove the “next” button from activity\_cheat.xml (we’ll adjust it properly later)

Our second activity will use this new layout, which we know will work for now (one fewer source of potential problems).

* Create a new class CheatActivity which sets its layout to activity\_cheat.xml
* Add a “Cheat” button to the original layout of QuizActivity
* Add a click listener to that Cheat button to start the new activity. You can do that by creating an explicit new Intent, referring to the new class CheatActivity. Then call startActivity with that new intent as input. You could optionally call the Intent method putExtra to store additional data in the intent to pass it along to the new activity. In our case, get the correct answer from the current question and put it in the intent. If you want the second activity to return results to the calling activity, you use startActivityForResult instead of just startActivity.

Try it out. You will generate a runtime error and the app will quit.

* You need to modify the ***Android manifest*** to permit the second activity to run

Now try again – this time it should work. You should start the second activity from the first, and use the standard Android Return button to get back to the original activity.

* Now modify the second layout to consist of two text views and one button. The button should say “Show Answer”, the first textfield show show a warning like “Are you sure you want to do that” and the second textfield will show the correct answer if the Show Answer button is pressed.

Here is the complete source listing. You could create a new project, for example, and paste all files into the appropriate locations – you should get a working app (you might need to adjust the package name, depending on what you chose).

**dimens.xml**

<resources>

 <dimen name=*"text\_size"*>24dp</dimen>

</resources>

**strings.xml**

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

<resources>

 <string name=*"app\_name"*>Geo Quiz</string>

 <string name=*"action\_settings"*>Settings</string>

 <string name=*"true\_button"*>True</string>

 <string name=*"false\_button"*>False</string>

 <string name=*"next\_button"*>Next</string>

 <string name=*"question\_text"*>Constantinople is the largest city in Turkey</string>

 <string name=*"toast\_correct"*>&quot;Correct, well done&quot;</string>

 <string name=*"toast\_incorrect"*>Wrong (sorry)</string>

 <string name=*"cheat\_button"*>Cheat!</string>

 <string name=*"warning\_text"*>Are you sure you want to do this? </string>

 <string name=*"show\_answer\_button"*>Show Answer</string>

 <string name=*"judgment\_toast"*>Cheating is wrong.</string >

</resources>

**activity\_quiz.xml**

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

 xmlns:tools=*"http://schemas.android.com/tools"*

 android:layout\_width=*"match\_parent"*

 android:layout\_height=*"match\_parent"*

 android:gravity=*"center\_vertical|center\_horizontal"*

 android:orientation=*"vertical"*

 tools:context=*".QuizActivity"* >

 <TextView

 android:id=*"@+id/id\_warning\_text"*

 android:layout\_width=*"fill\_parent"*

 android:layout\_height=*"wrap\_content"*

 android:ems=*"10"*

 android:text=*"@string/question\_text"*

 android:textSize=*"@dimen/text\_size"* >

 </TextView>

 <LinearLayout

 android:layout\_width=*"fill\_parent"*

 android:layout\_height=*"wrap\_content"*

 android:gravity=*"center\_vertical|center\_horizontal"*

 android:orientation=*"horizontal"* >

 <Button

 android:id=*"@+id/id\_button\_false"*

 android:layout\_width=*"wrap\_content"*

 android:layout\_height=*"wrap\_content"*

 android:text=*"@string/false\_button"* />

 <Button

 android:id=*"@+id/id\_button\_true"*

 android:layout\_width=*"wrap\_content"*

 android:layout\_height=*"wrap\_content"*

 android:text=*"@string/true\_button"* />

 <Button

 android:id=*"@+id/button\_show\_answer"*

 android:layout\_width=*"wrap\_content"*

 android:layout\_height=*"wrap\_content"*

 android:text=*"@string/next\_button"* />

 <Button

 android:id=*"@+id/button\_cheat"*

 android:layout\_width=*"wrap\_content"*

 android:layout\_height=*"wrap\_content"*

 android:text=*"@string/cheat\_button"* />

 </LinearLayout>

 </LinearLayout>

**activity\_cheat.xml**

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

 xmlns:tools=*"http://schemas.android.com/tools"*

 android:layout\_width=*"match\_parent"*

 android:layout\_height=*"match\_parent"*

 android:gravity=*"center\_vertical|center\_horizontal"*

 android:orientation=*"vertical"*

 tools:context=*".QuizActivity"* >

 <TextView

 android:id=*"@+id/id\_warning\_text"*

 android:layout\_width=*"fill\_parent"*

 android:layout\_height=*"wrap\_content"*

 android:ems=*"10"*

 android:text=*"@string/warning\_text"*

 android:textSize=*"@dimen/text\_size"* >

 </TextView>

 <TextView

 android:id=*"@+id/answer\_text"*

 android:layout\_width=*"fill\_parent"*

 android:layout\_height=*"wrap\_content"*

 android:textSize=*"@dimen/text\_size"* />

 <LinearLayout

 android:layout\_width=*"fill\_parent"*

 android:layout\_height=*"wrap\_content"*

 android:gravity=*"center\_vertical|center\_horizontal"*

 android:orientation=*"horizontal"* >

 <Button

 android:id=*"@+id/button\_show\_answer"*

 android:layout\_width=*"wrap\_content"*

 android:layout\_height=*"wrap\_content"*

 android:text=*"@string/show\_answer\_button"* />

 </LinearLayout>

 </LinearLayout>

**Question.java**

**package** org.mathcs.geoquiz;

**public** **class** Question

{

 **private** String question = **null**;

 **private** **boolean** isTrue = **true**;

 **public** Question(String question, **boolean** isTrue)

 {

 **this**.question = question;

 **this**.isTrue = isTrue;

 }

 **public** String getQuestion()

 {

 **return** question;

 }

 **public** **void** setQuestion(String question)

 {

 **this**.question = question;

 }

 **public** **boolean** isTrue()

 {

 **return** isTrue;

 }

 **public** **void** setTrue(**boolean** isTrue)

 {

 **this**.isTrue = isTrue;

 }

}

**QuizActivity.java**

package org.mathcs.geoquiz;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.view.Menu;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

public class QuizActivity extends Activity

{

 public final static String KEY\_CURRENT\_QUESTION = "current\_question";

 private Button buttonTrue = null;

 private Button buttonFalse = null;

 private Button buttonNext = null;

 private Button buttonCheat = null;

 private TextView question = null;

 private int currentQuestion = 0;

 private boolean hasCheated = false;

 private final static Question[] QUESTIONS = new Question[]

 {

 new Question("Is Bonn the Capital of Germany?", false),

 new Question("New Yourk is the biggest city on the East Coast of the USA", true),

 new Question("Bert is a great programmer", true)

 };

 @Override

 protected void onCreate(Bundle savedInstanceState)

 {

 super.onCreate(savedInstanceState);

 setContentView(R.layout.activity\_quiz);

 buttonFalse = (Button)this.findViewById(R.id.id\_button\_false);

 buttonTrue = (Button)this.findViewById(R.id.id\_button\_true);

 buttonNext = (Button)this.findViewById(R.id.button\_show\_answer);

 buttonCheat = (Button)this.findViewById(R.id.button\_cheat);

 question = (TextView)this.findViewById(R.id.id\_warning\_text);

 if (savedInstanceState != null)

 currentQuestion = savedInstanceState.getInt(KEY\_CURRENT\_QUESTION, 0);

 buttonTrue.setOnClickListener(new View.OnClickListener()

 {

 @Override

 public void onClick(View v)

 {

 checkAnswer(true);

 }

 });

 buttonFalse.setOnClickListener(new View.OnClickListener()

 {

 @Override

 public void onClick(View v)

 {

 checkAnswer(false);

 }

 });

 buttonNext.setOnClickListener(new View.OnClickListener()

 {

 @Override

 public void onClick(View v)

 {

 if (currentQuestion < QUESTIONS.length-1)

 currentQuestion++;

 else

 currentQuestion = 0;

 question.setText(QUESTIONS[currentQuestion].getQuestion());

 hasCheated = false;

 }

 });

 buttonCheat.setOnClickListener(new View.OnClickListener()

 {

 @Override

 public void onClick(View v)

 {

 Intent i = new Intent(QuizActivity.this, CheatActivity.class);

 i.putExtra(CheatActivity.KEY\_ANSWER\_TRUE, QUESTIONS[currentQuestion].isTrue());

 startActivityForResult(i, 0);

 }

 });

 question.setText(QUESTIONS[currentQuestion].getQuestion());

 }

 @Override

 public boolean onCreateOptionsMenu(Menu menu)

 {

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

 getMenuInflater().inflate(R.menu.quiz, menu);

 return true;

 }

 @Override

 public void onSaveInstanceState(Bundle state)

 {

 super.onSaveInstanceState(state);

 state.putInt(KEY\_CURRENT\_QUESTION, currentQuestion);

 }

 @Override

 public void onActivityResult(int requestCode, int resultCode, Intent data)

 {

 if (data != null)

 hasCheated = data.getBooleanExtra(CheatActivity.KEY\_ANSWER\_SHOWN, false);

 }

 @Override

 public void onPause()

 {

 super.onPause();

 Toast.makeText(this, "activity paused", Toast.LENGTH\_SHORT).show();

 }

 @Override

 public void onDestroy()

 {

 super.onDestroy();

 Toast.makeText(this, "activity destroyed", Toast.LENGTH\_SHORT).show();

 }

 private void checkAnswer(boolean answer)

 {

 if (hasCheated)

 Toast.makeText(QuizActivity.this, R.string.judgment\_toast, Toast.LENGTH\_SHORT).show();

 else if (QUESTIONS[currentQuestion].isTrue() == answer)

 Toast.makeText(QuizActivity.this, R.string.toast\_correct, Toast.LENGTH\_SHORT).show();

 else

 Toast.makeText(this, R.string.toast\_incorrect, Toast.LENGTH\_SHORT).show();

 }

}

**CheatActivity.java**

package org.mathcs.geoquiz;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class CheatActivity extends Activity

{

 public final static String KEY\_ANSWER\_TRUE = "answer\_is\_true";

 public final static String KEY\_ANSWER\_SHOWN = "answer\_was\_shown";

 private TextView answerText = null;

 private Button showAnswer = null;

 private boolean answer = false;

 @Override

 protected void onCreate(Bundle savedInstanceState)

 {

 super.onCreate(savedInstanceState);

 setContentView(R.layout.activity\_cheat);

 answerText = (TextView)this.findViewById(R.id.answer\_text);

 showAnswer = (Button)this.findViewById(R.id.button\_show\_answer);

 answer = getIntent().getBooleanExtra(KEY\_ANSWER\_TRUE, false);

 setAnswerShown(false);

 showAnswer.setOnClickListener( new View.OnClickListener()

 {

 public void onClick(View v)

 {

 if (answer)

 answerText.setText(R.string.true\_button);

 else

 answerText.setText(R.string.false\_button);

 setAnswerShown(true);

 }

 });

 }

 private void setAnswerShown(boolean answerShown)

 {

 Intent data = new Intent();

 data.putExtra(KEY\_ANSWER\_SHOWN, answerShown);

 setResult(RESULT\_OK, data);

 }

}

**AndroidManifest.xml**

Open the manifest file and switch to XML mode. Find the <application> …. </application> section. Inside that is an <activity> … </activity> section. Immediately after that but still inside the application section add the tags:

 <activity

 android:name=*"CheatActivity"*

 android:label=*"@string/app\_name"* >

 </activity>

Save your project and run it. Make sure you understand the data exchange:

* from the main activity to the cheat activity via the extra data added to the intent that starts the second activity
* from the cheat activity back to the main activity by
	+ using startActivityForResult in the main activity
	+ in the Cheat activity creating a new intent, adding the data to be passed back to that intent, then posting the intent via setResult
	+ Picking up the embedded data by by overriding in the main activity:

 **public** **void** onActivityResult(**int** requestCode, **int** resultCode, Intent data)

 {

 **if** (data != **null**)

 hasCheated = data.getBooleanExtra(CheatActivity.*KEY\_ANSWER\_SHOWN*, **false**);

 }

Note that we are basically ignoring the “resultCode”, which could be “cancel” or “ok”, as well as the “requestCode”.

BUG: I can display a hint yet not be called a cheater. How – fix it.