Henry Post

ITMD 455

Lab 5: Quiz

Contents

MainActivity.java

```
package me.henryfbp.quiz;

import android.app.Activity;
import android.app.ProgressDialog;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RatingBar;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {

    private static final String DATA_URI = "http://www.papademas.net:81/sample.txt";

    private static final HashMap<Integer, Boolean> ANSWER_BANK = new HashMap<Integer,
Boolean>() {{
        put(0, true);
        put(1, false);
        put(2, true);
        put(3, false);
        put(4, true);
    }};

    private static final HashMap<Integer, Boolean> USER_CORRECT_ANSWERS = new
HashMap<Integer, Boolean>();


    static int questionNum = 0;
    Activity context;
    TextView txtView;
    ProgressDialog pd;
    ArrayList<String> stringList = new ArrayList<>();
    ImageView imageViewNext;
    ImageView imageViewPrev;
    private RadioGroup radioQuestions;
    private RadioButton radioButton;

    public static Integer getBooleans(HashMap<Integer, Boolean> h) {
```

```
        int x = 0;

        for (Map.Entry<Integer, Boolean> e : h.entrySet()) {
            if (e.getValue()) {
                x += 1;
            }
        }
        return x;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        context = this;

        BackgroundTask bt = new BackgroundTask();
        bt.execute(DATA_URI); //grab url

    }//end onCreate

    public void startQuiz() {
        buttonListener();
    }

    public void buttonListener() {

        Button btnDisplay;

        radioQuestions = findViewById(R.id.radioQuestions);
        btnDisplay = findViewById(R.id.btnDisplay);

        btnDisplay.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {

                // get selected radio button from radioGroup
                int selectedId = radioQuestions.getCheckedRadioButtonId();


                if (selectedId == -1) { // They haven't selected anything...
                    Toast.makeText(MainActivity.this, "You haven't selected anything!",
Toast.LENGTH_SHORT).show();
                    return;
                }
                try {
                    // find the radiobutton by returned id
                    radioButton = findViewById(selectedId);

                    //If answer matches our answer bank
                    boolean correctAnswer = ANSWER_BANK.get(questionNum) ==
radioButton.getText().toString().equalsIgnoreCase("true");

                    // Record user's answer's correctness
                    USER_CORRECT_ANSWERS.put(questionNum, correctAnswer);
```

```
                    RatingBar r = findViewById(R.id.ratingBar);

                    Float rating;

                    Float correct = Float.valueOf(getBooleans(USER_CORRECT_ANSWERS));
                    Float total = (float) USER_CORRECT_ANSWERS.size();
                    Float numStars = (float) r.getNumStars();


                    TextView textViewRating = findViewById(R.id.textViewScore2);
                    textViewRating.setText(String.format("%.2f/%.2f", correct, total));

                    if (USER_CORRECT_ANSWERS.size() == 0) {
                        rating = 1f;
                    } else {
                        rating = (correct / total) * numStars;
                    }

                    // Set rating to (correct / attempted).
                    r.setRating(rating);

                    Toast.makeText(MainActivity.this,
                            correctAnswer ? "Right!" : "Wrong!",
                            Toast.LENGTH_SHORT).show();

                } catch (NullPointerException e) {

                    // Our answer bank doesn't have this answer.
                    Toast.makeText(MainActivity.this, String.format("We don't have an
answer for number %d.", questionNum), Toast.LENGTH_LONG).show();
                }

            }
        });
        imageListener();
    }//end buttonListener


    public void imageListener() {

        imageViewNext = findViewById(R.id.imageViewNext);
        imageViewNext.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {
                // get new question for viewing

                if (questionNum >= stringList.size() - 1) {//reset count to -1 to start
first question again
                    questionNum = -1;
                }

                txtView.setText(stringList.get(++questionNum));

                //reset radio button (radioTrue) to default
                radioQuestions.clearCheck();
            }
```

```
    });

    imageViewPrev = findViewById(R.id.imageViewPrev);
    imageViewPrev.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {
            // get new question for viewing

            if (questionNum <= 0) {//reset count to -1 to start first question again
                questionNum = stringList.size();
            }

            txtView.setText(stringList.get(--questionNum));

            //reset radio button (radioTrue) to default
            radioQuestions.clearCheck();
        }
    });
}//end imageListener

//background process to download the file from internet
private class BackgroundTask extends AsyncTask<String, Integer, Void> {

    protected void onPreExecute() {
        super.onPreExecute();
        //display progress dialog
        pd = new ProgressDialog(context);
        pd.setTitle("Reading the text file");
        pd.setMessage("Please wait.");
        pd.setCancelable(true);
        pd.setIndeterminate(false);
        pd.show();

    }

    protected Void doInBackground(String... params) {
        URL url;
        String StringBuffer = null;
        try {
            //create url object to point to the file location on internet
            url = new URL(params[0]);
            //make a request to server
            HttpURLConnection con = (HttpURLConnection) url.openConnection();
            //get InputStream instance
            InputStream is = con.getInputStream();
            //create BufferedReader object
            BufferedReader br = new BufferedReader(new InputStreamReader(is));

            //read content of the file line by line & add it to Stringbuffer
            while ((StringBuffer = br.readLine()) != null) {
                stringList.add(StringBuffer);//add to Arraylist
            }

            br.close();

        } catch (Exception e) {
```

```
                e.printStackTrace();
                //close dialog if error occurs
                if (pd != null) {
                    pd.dismiss();
                }
            }

            return null;

        }

        protected void onPostExecute(Void result) {
            //close dialog
            if (pd != null) {
                pd.dismiss();
            }
            txtView = findViewById(R.id.textViewQuestion);
            //display read text in TextView
            txtView.setText(stringList.get(0));
            startQuiz();

        }
    }//end BackgroundTask class

}//end activity
```