Danielle Hooven

IST269-225

Cyril Keiffer

04 May 2020

**Final Project: Harry Potter Trivia App**

**Description**

Harry Potter enthusiasts and friends will love this trivia game. The game is quick and easy to play as there are only a few questions. See what house you would be sorted into depending on your results. Play a round and test your knowledge or learn some new trivia about “the boy who lived” a.k.a. Harry Potter.

Disclaimer: This app is a beginner project that was created using Android Studio to demonstrate mastery of techniques learned in an introduction to programming mobile application course. This app is beta and is only intended for private use.

Price: Free

**How I created the app.**

1. Reviewed the textbook and coursework, searched for beginner coding project ideas and decided to do a Quiz app relating to Harry Potter.
2. Listed out what techniques/methods from each chapter I could use in the app and how.
3. Created a mockup design of how the app will look.
4. Wrote out pseudo-code to help plan out the process flow.
5. Created activities for main (beginning), questions 1- 4, splash, and results.
6. Layout
   1. Uploaded the drawable resource images.
   2. Wrote the xml code to display each layout with widgets and attributes to match the mock-up design.
   3. Added the strings and string-array in the process.
   4. Created an animator resource.
7. Java
   1. For each activity with a button, created button object and added onClick event listener with a startActivity function to open the next screen in the process.
   2. RadioButtons (Question 2 & 3)
      1. Created variable to hold score, radioGroup object and radioButton object.
      2. Added code to check if a radioButton is selected.
      3. Assigned value of correct to appropriate button.
      4. Added if else statement to check for selection first and then compare selection with the correct answer variable.
      5. Inserted code to increase score counter if answer is correct.
   3. Spinner Control
      1. Created variable to hold score and spinner object.
      2. Assigned value to correct answer and not selected.
      3. Added if else statement to check for selection first and then compare selection with the correct answer variable.
      4. Inserted code to increase score counter if answer is correct.
   4. DatePicker
      1. Created variable to hold score and datePicker object.
      2. Initiated date.
      3. Assigned value to correct answer variable.
      4. Added if else statement to compare selection with the correct answer.
      5. Inserted code to increase score counter if answer is correct.
   5. Splash
      1. Added code to set timer, code to display frame animation, and startActivity to open the final results activity.
   6. Results
      1. Created objects for the display items that would change depending on the results.
      2. Wrote out the switch case statement for the various result possibilities and what text and images should be displayed depending the final score.
   7. SharedPreferences
      1. Went back through and added the code to create a sharedPreference object, edit the score in the key, store the key, and get data.
   8. Debugging and Cleaning Up
      1. Ran the program multiple times, troubleshooting any errors along the way.
      2. Noticed a pattern in each activity. Went back through and cleaned up the code so they all matched and added pseudo-code comments to clarify the process.

**What I found easy.**

The easiest parts were designing the layouts, adding the onClick listeners, and the splash screen with frame animation. Although designing the layout was easy, adjusting the attributes to get the exact look I wanted to a lot of trial and error. I thought sharedPreference was going to be a lot harder than it was but because I had already added a variable to hold the total score, it was so easy to assign that to a key and call it up in the next activity and update it. I just repeated that pattern after each question.

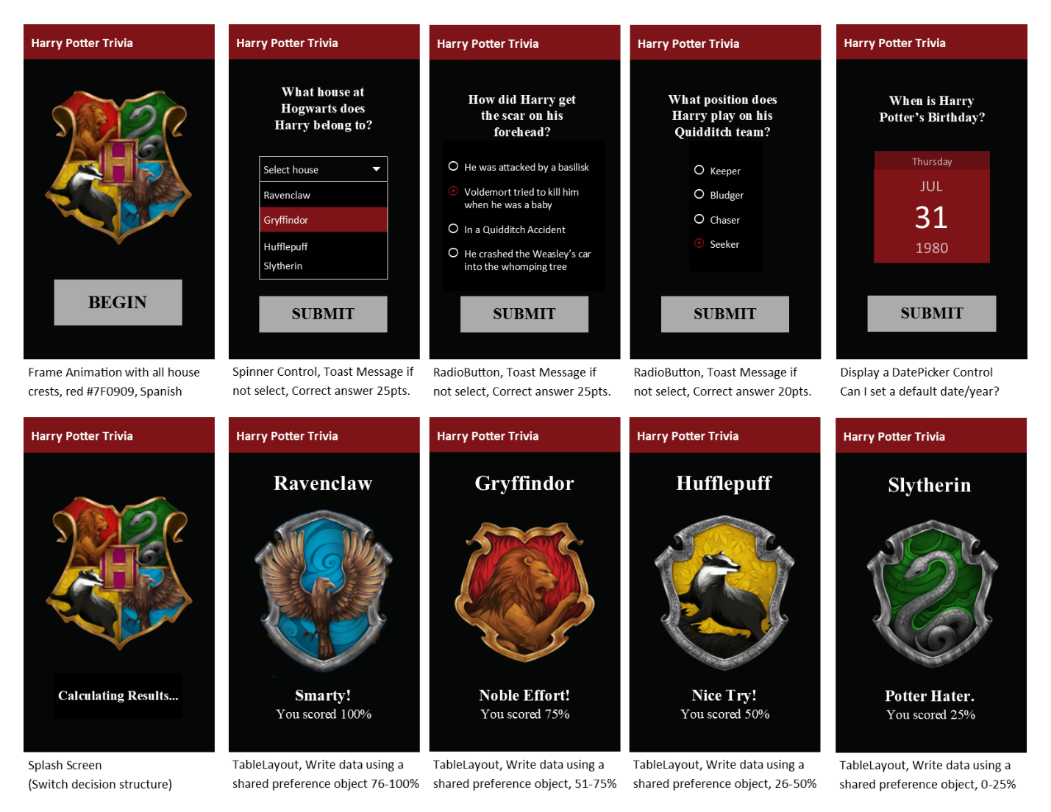
**What I found challenging.**

It took a lot of forethought and planning to work out the logic, but I think the datePicker was probably the most challenging. I researched a lot to figure out how to change the look, set a date range (never worked ☹), and initiate a date. I also did additional research for examples on how to check that an answer is correct using if statements. My first attempt was pretty close, so I didn’t need to do much adjusting. I wish there were more customization available for the visual side of things.

**Techniques from Each Chapter…**

* Specify the use of the layout in the user interface (Ch. 1)
  + Relative Layout
* Create multiple Android Activities (Ch. 2)
  + Each question is a new activity
* Develop a user interface using a Spinner control (Ch. 3)
  + Question with dropdown list to select answer
  + Uses a string-array to store list items.
* Use an Android Theme (Ch. 3)
  + Customized black background with red title bar
* Place a RadioGroup and RadioButtons in Android applications (Ch. 4)
  + Question with RadioButtons to select answer
* Write code using the Switch decision structure (Ch. 5)
  + Case 4: 100% Ravenclaw
  + Case 3: 75% Gryffindor
  + Case 2: 50% Hufflepuff
  + Case 1: 25% Slytherin
  + Case 0: 0% Slytherin
* Create an Android project using a splash screen (Ch. 6)
  + “Calculating Results”
* Display a custom toast message (Ch. 7)
  + Message pops up if an answer is not selected.
* Display a DatePicker Control (Ch. 8)
  + When is Harry Potter’s birthday?
* Display a custom layout…a table layout (Ch. 9)
  + Final screen with picture and results
* Create an Android application with Frame animation (Ch. 10)
  + Animated image on splash screen
* Write data using a SharedPrefence object. (Ch. 11)
  + Correct answer gets 1 point, wrong answer gets 0.
  + Store the point, use it for running total.
  + Stored data is used to decide result page display.
* Target various languages (Ch. 12)
  + Spanish alternative.

**Mock-Up Design with Notes**



**Psuedo-Code**

1. Opening
   1. Display title, image and button.
      1. When user taps begin, the first question is opened in a new activity.
2. Question One
   1. Display a text field with the question and spinner control. The options in the spinner are each of the Hogwarts houses.
      1. When the user taps submit
         1. If no selection is made, display custom toast message.
         2. The point value for the selected answer is added to the total.
         3. The second question is opened in a new activity.
3. Question Two
   1. Display a text field with the question, RadioGroup and RadioButtons. The user selects a RadioButton to answer.
      1. When the user taps submit
         1. If no selection is made, display custom toast message.
         2. The point value for the selected answer is added to the total.
         3. The third question is opened in a new activity.
4. Question Three
   1. Display a text field with the question, RadioGroup and RadioButtons. The user selects a RadioButton to answer.
      1. When the user taps submit
         1. If no selection is made, display custom toast message.
         2. The point value for the selected answer is added to the total.
         3. The four question is opened in a new activity.
5. Question Four
   1. Display a text field with the question and calendar to select date.
   2. Initialize default date
      1. When the user taps submit
         1. The point value for the selected answer is added to the total.
         2. The splash screen is opened in a new activity.
6. Calculate Results
   1. A splash screen opens for 12 seconds with the text “Calculating Results” and displays animation with the Hogwarts crest images for 2 seconds each.
   2. After timer, open results activity
7. Results
   1. Display title, image, comment, and total percentage in table layout.
   2. Use switch decision case structure pulling in the final score saved in persistent data.
      1. Case 4: Total = 4 pts (Ravenclaw)
         1. Display 4 table rows within a table layout with a house title, image, phrase, and results.
      2. Case 3: Total = 3 pts (Gryffindor)
         1. Display 4 table rows within a table layout with a house title, image, phrase, and results.
      3. Case 2: Total = 2pts (Hufflepuff)
         1. Display 4 table rows within a table layout with a house title, image, phrase, and results.
      4. Case 1: Total = 1 (Slytherin)
         1. Display 4 table rows within a table layout with a house title, image, phrase, and results.
      5. Case 4: Total = 0 (Muggle)
         1. Display 4 table rows within a table layout with a house title, image, phrase, and results.