Anthony Powell

Anjopowe

A290-Android

February 24, 2015

*Final Project Phase 2b – Programming/Logical Design*

For this application, there will be a total of 11 activities: MainActivity, LessonsHome, LessonPage, ChordsHome, ChordPage, ScalesHome, ScalePage, SongsHome, SongPage, Tuner, and About. Each activity will be represented in it’s own Java file, as well as have it’s own unique layout (XML) page. Each activity will run separately on it’s own page, and will be designed as follows:

**MainActivity:**

This will be the main activity and first page that opens when the application is started. The page will use a LinearLayout (vertical), will implement OnClickListener, and will consist of seven buttons and a title. The title will appear at the top in the title bar next to the icon, and the buttons are as such: Lessons, Chords, Scales, Songs, Tuner, About, and Exit. OnClickListeners are set for each button to allow an action to take place when the buttons are clicked, and clicks are received by the OnClick function by case. The Lessons, Chords, Scales, Songs, and Tuner buttons start new activities in a new page (e.g. Lessons opens up LessonsHome page). The About buttons creates an Intent that opens up the AboutActivity, which is displayed by a dialog. The Exit button closes the application.

**LessonsHome:**

This activity will display all of the lessons and will use a ListView layout. The title bar at the top of the page will read “Beginner Lessons”. Each lesson will be displayed as a list item, and each lesson will be connected to an OnClickListener. When a lesson is selected, an Intent is created with the LessonPage activity, and an extra variable is passed to the activity based on which lesson is selected (all in the onClick function). This variable is used for determining what data to create the following page with.

**LessonPage:**

This page is for the individual lessons and uses a vertical LinearLayout. Only one Java file will be used to represent the Lessons as a whole, and depending on which Lesson was selected, the code will determine which page to display. When the Intent is created, a variable selecting the specific lesson is also passed into the activity, and this will help determine which lesson to create. Each lesson can be made up of multiple resource types, and not necessarily always in the same order, so each lesson will have it’s own resource array (e.g. A lesson could consist of a paragraph, then a diagram of a chord, and then more text). The variable passed through the Intent allows us to select which array to go through. The correct array is then looped through, from start to finish, placing the objects from the array in the LinearLayout.\*\*

**ChordsHome:**

This page will display the list of chords the user can lookup and uses a ListView layout. The title bar will be a string reading “Chords”. This page will be run the same as the LessonsHome page, with onClickListeners being set for the list items, which are all chords, and when one is selected the onClick function creates an Intent to open the ChordPage activity with an extra variable passed to determine what data to use for the chord page.

**ChordPage:**

This page is for the individual chords and uses a vertical LinearLayout. The generation of this page will be based on the Intent variable passed in, and will consist of the chord name in the title bar and a diagram (given by an image resource) showing how to play the chord within an array for each specific chord. The first place in a chords’ array will be the name, and the second will be the image resource.

**ScalesHome:**

This activity displays the different scales for the user to look up and uses a ListView layout. The title bar will be a string reading “Scales”. This page will have the same functionality as LessonsHome and ChordsHome. Each list item will be a different scale, and onClick will create an Intent with an extra variable passed for the next page to use to create the Scale page.

**ScalePage:**

This page is for the individual scales and uses a vertical LinearLayout. The generation of this page will be similar to that of the individual chords’ page, with an image/diagram showing how to play the scale, but with a third place in the array containing a string with text about the scale and its use.

**SongsHome:**

This activity has a list of songs to view and uses a ListView layout. The title bar will be a string reading “Beginner Songs”. It will use the same functionality as LessonsHome, ChordsHome, and ScalesHome. Each list item will have an onClickListener, and onClick will create an Intent with the SongPage activity and pass an extra variable depending on which is clicked.

**SongPage:**

This page is for the individual songs that will be included on the application. The title bar will have the song’s name, and then a linearLayout will be used to display the how-to for the song. This can consist of many different resources, so the generation of each song’s page will be the same as that of the individual lessons’ pages. An array will be made for each song containing any images needed and any text needed, and the array will be looped through to display each resource in order on the page. \*\*

**Tuner:**

The tuner page will use a RelativeLayout. The title bar will contain a string (“Tuner”). The top of the layout will be a textView of large centered text saying “Standard Tuning (EADGBE)”, and below that will be a quick instructional on how to use the tuner. Then, there will be six buttons in 2x2x2 positioning, with each button representing a string on the guitar. All six of these buttons will contain an OnClickListener, and the onClick function will take care of the actions resulting from pressing the buttons. When a button is clicked, a sound with the pitch of that string being played, so onClick will have 6 cases depending on which button is clicked and determine which sound to play. This will be done by creating a MediaPlayer within onClick and setting the mediaPlayer’s sound based on the button that’s clicked and then starting it. Each time a button is clicked, the mediaPlayer is reset.

**About:**

This page will be a dialog that appears when the About button is selected from the Main Activity. The dialog title will simply be “About *AppName*”, and the text will be a short description of the application itself and its purpose. The layout used will be a ScrollView in the case that the dialog box doesn’t contain enough space on the device being used.

**Consistency:**

A constant theme will be used throughout the application, and the color scheme will be consistent throughout each page. The Home page will have its own look, as will the tuner page, but text and buttons styles will be the same. For the other pages, the “home” pages will all have the same menu style. Then, the individual pages will all have the same general layout. Although they will contain different resources in different positions, the resources will be styled the same to keep the entire application consistent. Also, where images such as diagrams are used, consistency will be kept amongst these as well. For example, chord diagrams will all have the same look, as will images with tablature or scales.

**Navigation:**

Navigation in this application will be very straightforward. There is a general theme of Main->’TypeHome’->’TypeIndividual’ for the majority of pages with the exceptions of the about page (a dialog) and the tuner. All navigation is completely linear in that you would never need to jump to one page outside of the above theme, you simply follow the order (e.g. Main to ChordsHome to IndivChord to ChordsHome back to Main). Therefore, the only navigation other than clicking a button/list item to go to another page is going back, which can be done with the device's back button.

**External Resources:**

*Images/Diagrams* – added to an images folder and implemented with Drawable resources into the layouts

* will need images containing tablature, chord diagrams, and scale diagrams

*Sounds* – added to a sounds folder and played using the Android MediaPlayer

* the sounds needed will just be that of the pitches for each string for the tuner

***\*\*NOTE:***

As for the process of looping through an array of resources when the page can differ (used in the Lessons and Songs pages), I’m not entirely sure at this point how that will work. A different method that what is described may need to be used as I’ve never done something like this before. Regardless, something similar (or to accomplish the same task) will be done, but the exact method isn’t known at this time assuming the method above doesn’t work.