ECE 1778 – Creative Applications for Mobile Devices
September 2016
Programming Assignment P2

Containers, Fragments, Select, Lists and Files

The goal of this assignment is to learn more of the basics of Android/iOS capabilities, beginning with basic layouts of what a user sees, and then selects, how to present lists and how to store files. **Warning:** There is quite a bit of work in this assignment, at least 6-8 hours, perhaps more if you have less experience.
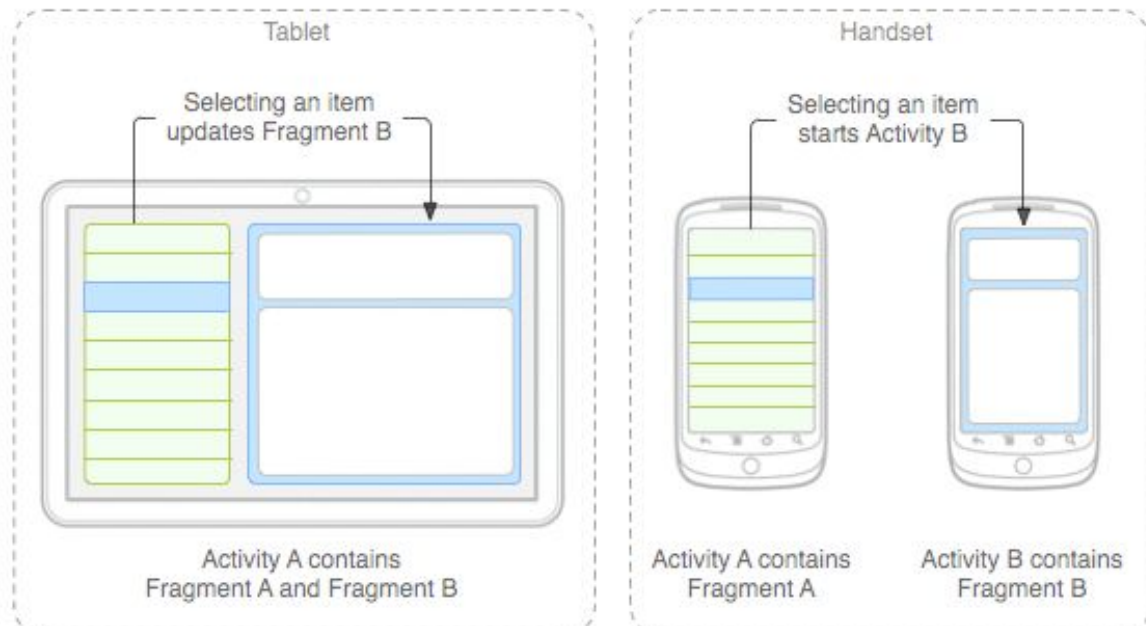
1   Review

Read and do the chapters titled "Linear Layout and the Box Model," "The Tactics of Fragments" and "Assets, Files and Data Parsing" **The Busy Coder's Guide to Android Development**, Version 7.6, doing the small coding exercises given there.

If you are developing on the iPhone, and Objective-C, **Beginning iPhone 7 Development Exploring the iOS SDK** by Nutting, Olsson, Mark and LaMarche and, covers similar territory in Chapter 4: "More User Interface Fun", Chapter 8: "Introduction to Table Views" and Chapter 13: "Basic Data Persistence."

If you're using Swift on the iPhone, and using the **Beginning iPhone Development with Swift 2,** by Mark, Topley, Nutting Olsson and LaMarche, the same chapters apply – 4, 8 and 13.

2   A Note About Fragments

In Android programming *activities* are containers, which hold logically independent units of your UI, which in turn are called *fragments*. In the simplest case an activity will hold and display a single full screen fragment, as shown on the handset in the image below.

Tablet

Selecting an item updates Fragment B

Handset

Selecting an item starts Activity B

Activity A contains Fragment A and Fragment B

Activity A contains Fragment A

Activity B contains Fragment B

One of the (many) benefits to having coded these UI elements as fragments is to ease the transition from phone to tablet app. Notice the tablet in the image above. Many other frameworks also require the use of fragments, so we require you to do that in this assignment if you're using Android. The UI system will check how much screen real estate there is and when there is enough it decides to layout fragment A besides fragment B, rather than swapping between fragments A and B as it would on the phone.

Android design guidelines *heavily* encourage the use of **dynamic fragments**. All of the interesting UI paradigms, designs, and frameworks all require knowledge of fragments. As a result we will introduce and enforce the use of fragments in various parts of the assignments of this course including in this assignment.

When completing these assignments **iPhone developers can ignore** the concept of using fragments, as there is no direct analogy to this design methodology in iOS.

3   Assignment

NOTE: before starting this assignment, please go back and read 'Braiden Brousseau's Guide To Quality Apps' that was part of Assignment P1. Your assignment should obey those guidelines, as part of the grade will be assigned for fulfilling those requirements.

You are to write an app that allows the user to create a list of movies along with their the year it was made and one of the actors in it. This information should be stored in a file or files on the device, and so permit the storage and loading of those.

The app should work as follows:
- The starting screen should have five buttons: EnterTitle, View, Store, Load and Exit
- Pressing EnterTitle should take the user to a new screen where they can enter a series of records describing the movie: the title, the year the movie was made, and one actor's name. The selection of the year should be from a drop down list of different years, not free-form text. The user should be able to enter as many movies as desired from this screen, and return to the starting screen through a 'Done' button.
- The Store button should cause the list to be written out to a file on the device using a file name provided by the user.
- The Load button should allow the user to read in a previously stored file by showing a list of all the previously stored files. (This takes some extra work).
- The View button displays, in a scrollable list, the set of movies entered, along with year made and actor's name.
- The Exit button terminates the application, but makes sure any that any list that has been created or modified, will be stored first.

While implementing the features described above you are required to use only two Activities. A main Activity that is launched when your app starts and an Activity that gets launched when the View button is pressed which will display the current list of people. Screen transitions that occur when Add people, Load, or Store are pressed should be accomplished by swapping in and out fragments **dynamically**, not launching new activities.

In general when to swap to a new fragment in the current Activity and when to launch new Activities can be subtle, non-obvious distinction. The requirement to have both in this assignment is strictly for learning purposes; a more obviously beneficial (but also more complicated example) use of fragments will be required for assignment 4.

Important Note on Marking:
       8/10 marks are assigned to meeting the above specifications.
       2/10 will be based on the quality of the User Interface and User Experience.

       Please refer to Braiden Brousseau's guide to quality apps to understand the guidelines for good quality user interface and user experience.

Due date: Thursday October 6th, at 6pm, marked out of 10, 0.5 marks off every hour late.

Submit your work the Blackboard Course Portal, under Programmer Assignments, under the 'P2' Assignment.
1. Android developers: a zip file containing your final Android application file (.apk); use your student number as the filename. Also submit the complete Android Studio project directory in a separate zip file.
2. iPhone developers: you must submit the complete project directory, including source, in a zip file. Use your student number as the filename. Please do your development on the 8.0 version of the SDK, and make sure that you haven't included any files by reference. Please test your submitted zip file before sending it in.