

Lab 4: Intents and Activities: CS71 Demo's App

Due: Feb 28th at 11:59pm

Overview

In this lab, you will program a multi-activity applications that demos previous class labs and exercises as well as at least two new activities that will communicate information to calling Activity via Intents. The layout of the demo Activity will rely on `ListView` and an `ListAdapter` employing `CheckedTextView`'s to organize the demo application for the user. After demoing each application, the check mark in the `CheckedTextView` will become checked. **There will be no provided skeleton code for this lab: You will start from basic Android Application Project in Eclipse.**

Deliverables

Your submissions will include the entire Android project. Please submit all requisite files such that your application will run. You must still provide a `README` file. Your `README` file must include, (1) your name, (2) the name of all *relevant* files in your submission, and (3) a short description of the program and your development process. Failure to include a `README` or not including sufficient information in the `README` will adversely affect your grade on this lab.

Project Retrieval and Submission Instructions

To submit this lab, once you've created the Android project in Eclipse, you will share the project over SVN to the following URI

```
https://cs71.cs.swarthmore.edu/svn/cs71/submission/<username>/04
```

Where `<username>` is replaced with your swarthmore username. Click finish, and then follow the on-screen instructions to commit. You may update and commit your project until the deadline. All commits after the deadline will be rolled back to the submission just prior to the deadline.

Application Description and Interaction

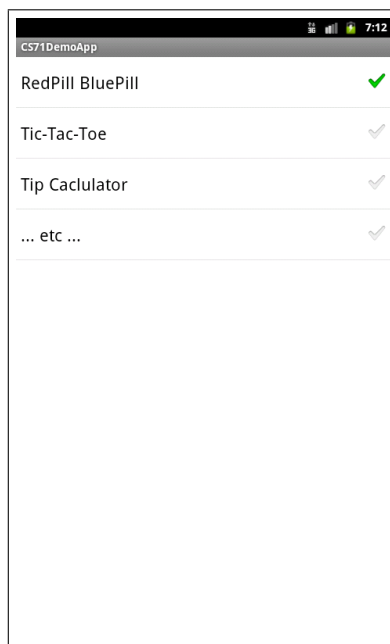
In this lab, you will make a demo application that will show off all the Android Activity's you've written, both during lecture and in the lab assignments. Additionally, your demo application must include two new Activities that can participate in the Intent response procedure, challenging a user to solve a small puzzle, the response being the successful or unsuccessful completion of the puzzle. You may choose your own puzzles, but I would suggest word games, e.g., anagrams, or other simple to program games.

Listview and Checked Listed Items

Your demo applications layout will be constructed using a `Listview` and an adapter for `CheckedTextView`. You can do so using the android resource ID

```
android.R.layout.simple_list_item_checked
```

set as the `TextViewResourceID` in the adapter. Each item in the `Listview` will display the name of the demo app, propagated using an `ArrayListAdapter`. Using the adapter, the `Listview` will generate a `CheckedTextView` for each item in the `ArrayListAdapter`. The check mark in the `CheckedTextView` will indicate weather the demo has been completed, and demos are started by clicking on items in the `Listview`. A sample image of the application with the first item checked is below:



The check mark should only appear **after** the demo has run: **You should not set the check mark in `OnItemClickListener`**. By *after*, I mean that you must use a `Intent`'s to indicate that the demo Activity has completed (e.g., when the user hits the *back* button) and the `Listview` comes back to the foreground before highlighting the check mark. This is the primary challenges in this lab; passing information between Activity'ies using `Intent`'s. Below I've provided hints by listing some of functions you might want to use to accomplish this task:

- `startActivityForResult()`

- `onActivityResult()`
- `onBackPressed()`
- `finish()`
- `Intent.putExtra()`
- `Intent.getExtras()`

Additionally, you will find it necessary to edit the demo `Activity`'s minimally, but not the core functionality.

Requisite Demo Activities

Your demo app is required to support, with their text names and descriptions.

- *Enter the Matrix* : The `RedPillBluePill` exercise from lecture
- *Tic Tac Toe* : Your `TicTacToe` Application from lab
- *Tip Calculator* : The `TipCalculator` exercise from lecture
- *RPN Calculator* : The `AndroidRPNCalculator` from lab
- *Life Cycle Demo* : The `HelloLifeCycle` exercise from lecture
- *Extra Demo 1* : A new small task-based Activity
- *Extra Demo 2* : A new small task-based Activity

You will likely find it easiest to copy past previous code into the appropriate places in the Application Project: **Don't forget the layouts and other resource values.**

For your two new demo apps, these must be some sort of task based activity where the user is required to successfully complete in order for the check mark to appear. For example, the user could be tasked to solve a simple anagram.

Competition and Extra Credit

There is no specific extra credit on this lab, instead all extra credit will be competition based, focused primarily on the extra apps. Be creative. The winner of the competition will receive 1 point of extra credit, the second and third runner up will receive .25 points, if there are sufficient entrants, e.g., more than 5 entrants.