CST2335 Graphical Interface Programming - draft

# Project Assignment

Due: See Blackboard for due dates

## Overview

This is a group project for groups of 4 members (all in the same lab section). Students will form pairs (within their lab section), and the lab instructors will put pairs together to form teams of 4. If you would like to be assigned a partner, send an email to your lab instructor. If you have chosen a partner yourself, email the names of the pair to your lab instructor. You must work in the group to which you are assigned. Use your assigned group number WITH your names in labeling submissions. Be sure to work through the Group Activity Worksheet together to exchange contact information amongst your group (Algonquin student email addresses at a minimum). You should also determine who would be working on what part of the project. Your group may choose a name for your group if you like. All work must be the work of the group members and ONLY the group members:

* **If a member submits plagiarized work, the whole group will be charged.**
  + **Double-check each other’s work to ensure sources are cited within program comments.**
* Each person will be responsible for their own part of the project and graded on this separately
* A portion of the grade will come from an evaluation of your participation as a group member

## Purpose:

The Project is assigned to give you experience in:

* Developing software in a group environment.
* Dividing workload to meet deadlines.
* Designing modular software that allows for that division.
* Learning from the work of others

## The Project

Your team's task is to develop a multi-function Android application with a tabbed Action Bar navigation system.

* There should be one Action Bar with buttons that navigate between functions
  + Include in your documentation the names of the group members and what application each person worked on
* Each person should choose one application (a different one each) and create a window for it
  + Also include a help screen that displays simple instructions and the name of the component's author
* The final team deliverable is a single Android Studio project that incorporates each person's application
* Note: Please feel free to bounce ideas off each other, ask for help debugging etc.
* User interface design is up to you, but Fragments should be used to make best use of the screen real-estate of a variety of devices (phones and larger-screened tablets, with support for both portrait and landscape modes)

## Hints

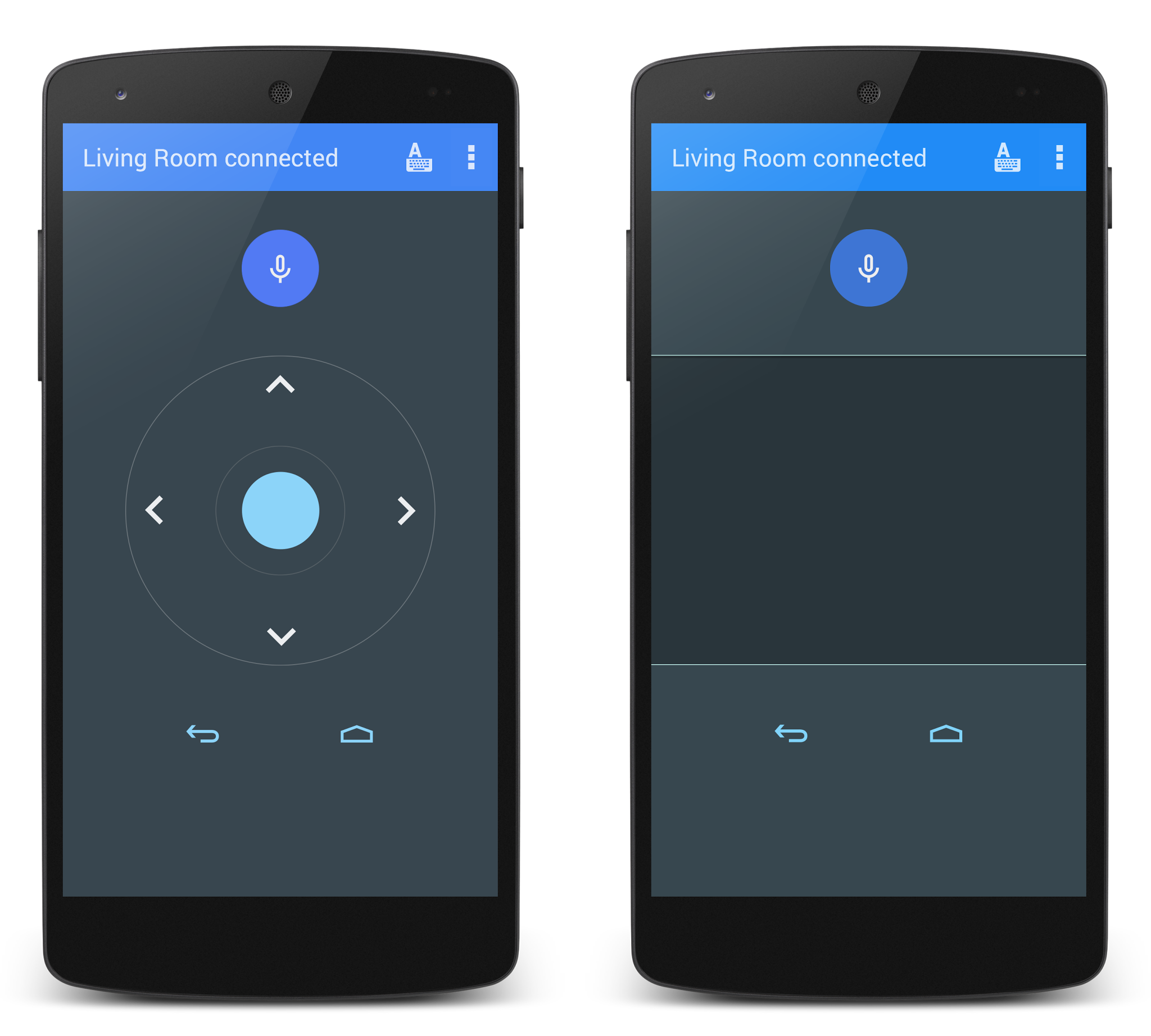
* Create a new GitHub repository to integrate the team's sub-applications early, even if some of the pieces are just a placeholder. INTEGRATE EARLY.
* See http://developer.android.com/guide/topics/ui/controls.html for a list of input controls that are available on the Android Platform (you've already used TextViews, but you can choose appropriate input mechanisms for your sub-application).
* A suggested structure for each sub-application is that of the FragmentBasics project. If each sub-application is structured the same way as FragmentBasics, each can be developed relatively independently, however, READ MY LIPS: INTEGRATE EARLY!
* An application that creates and uses an sqlite database has been posted to Blackboard -- you can copy/adapt pieces of that application to introduce database functionality in your own sub-application.
* A resource for learning how to use the Action Bar is http://developer.android.com/guide/topics/ui/actionbar.html#ActionEvents
* INTEGRATE EARLY

## The Applications

Each of the applications (as they are intended) requires similar programming techniques. Each application takes information from the user, and stores it in a database. Each application also provides functionality to summarize or analyze the whole body of data entered into the application. Beyond that you are free to get creative.

There is potential in some of these apps to make use of the Android environment, in the way of sensors, or native applications. That is not required, but you may be able to earn some bonus points if you make use of sensor data and/or integrate with native applications. For example, it would be intriguing to use motion and GPS sensors in your application. Check with your instructor before you take on this type of work, to ensure it is within scope.

**Living room smart environment remote interface**

* Include an instruction window that the user can access from a menu on the navigation bar
* You should use fragments to show a ListView displays several items that you would find in a living room. Selecting one of these items should display detailed controls for the item. You must design this interface yourself:
  + Lamp 1 – a simple on/off light, lamp 2 - a dimmable light, lamp 3 – a smart light strip that can be dimmable, as well as the colour of light.
  + A television – this should be a simple interface which has: on/off, channel entry, a direction pad (DPad) with “Enter” in the middle:
  + 
  + Smart window blinds. Use a slider to simulate opening and closing blinds on a window:



The items in the device list should be in order of most frequently used to least used. You must store the number of times each item was selected in a database.

**Kitchen smart environment remote interface**

* Include an instruction window that the user can access from a menu on the navigation bar
* You should use fragments to show a ListView displays several items that you would find in a kitchen. Selecting one of these items should display detailed controls for the item. You must design this interface yourself:
  + Microwave: The use can enter the cooking time, reset the clock, stop the microwave, or start. You must enable/disable the buttons as your interface transitions from entering the cooking time to starting to cook: disable reset and start buttons. The clock should actually count down to 0, and then use the vibration motor to vibrate the phone to show that the microwave is done.
  + Fridge: The user should be able to see the current temperature of the fridge and freezer sections. The user should also be able to change the temperatures within certain limits (maybe a slider or spinner?).
  + Main light: should be on/off and dimmable.

**House settings**

* Include an instruction window that the user can access from a menu on the navigation bar.
* You should use fragments to show a ListView displays the settings of a house. Selecting one of these items should display detailed controls for the item. You must design this interface yourself:
  + Garage – allow the user to open / close the door, and turn on a light. Opening the garage door should also turn on the light automatically. The user should be able to turn the light on and off afterwards.
  + House temperature – the user should be able to view and set the house temperature. The user should also be able to create a schedule by adding time/temperature settings that should appear in a list.
  + Outside weather – use your work from the lab to display the current temperature outside.

**Automobile**

* Include an instruction window that the user can access from a menu on the navigation bar.
* You should use fragments to show a ListView displays the settings of a car. Selecting one of these items should display detailed controls for the item. You must design this interface yourself:
  + Temperature settings – allow the user to set the temperature in the front, and back of the car (2 zones)
  + Fuel level – show the amount of fuel in the tank, and an estimate of how many kilometers the car can drive (look up a real car’s fuel economy settings (100L/km) to calculate this. There should be a “fill tank” button which simulates filling the car’s gas tank (look up the car’s fuel capacity on the internet).
  + Radio controls – include preset radio stations (6) that can be configured by the user.
  + GPS directions – This should launch the google navigation Intent
  + Lights – There should be a setting for turning on the headlights (normal, high), as well as a dimmable light inside the car.
  + Odometer showing how far the car has driven, and a trip distance counter that can be reset to 0.
  + Drive – An “On/Off” button which queries the user how far to drive. The user enters the number of kilometers for this trip, which should update the Odometer, and fuel level based on the fuel economy calculations. There should also be a “check oil” light that come on every 6000 km.

## Suggestions:

* As early as possible:
  + Decide who will work on which application.
  + Determine the additional tasks and decide who will take on each, for example Technical Lead, Action Bar, Project Management and Communication Lead, Code Custodian, Documentation, Test Plan Integrator, Integration Tester, and any others you can determine
  + **Discuss and document a code-freeze date for the group project**, I recommend **72 hours before the actual due date** so final code files can be merged into the project in preparation for upload to Blackboard.
* Help each other with debugging
* Attempt to merge code modules into the larger project to ensure that things are working at least once a week.

## Grading Guide

* Grading in 3 parts
* Arrange a single submission of the group deliverable by one of the group members on behalf of the entire group.
* Each student is graded on his or her application separately (80%)
  + Each student should upload a short word document indicating what sub-application and other tasks they worked on. (also upload the Grade Activity Worksheet, see below)
* Each student is graded on his or her team participation (15%)
  + 7.5% is based on your self-evaluation
  + 7.5% is based on the average of your team members peer review
  + Note: If you do not submit an individual Self and Peer form your self-evaluation becomes zero
  + Note: If a team member does not submit their self and peer form, the other team members will not be penalized.
* Group activity worksheet (5%)
  + ***All you need to do get the full 5% is to upload the filled in worksheet by the end of Week 13***

**Individual Application Grading (80%)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Criteria | Naming | Comments | Concepts | Compiles | Functionality/  Execution | Problems? | Total |
| Points | 1 | 2 | 4 | 1 | 4 | -4 | 12 |

Naming

All source code files have appropriate names, and Java syntax follows recommended naming conventions.

Comments\*\*\*

All class declarations and class members (constructors, properties, methods etc.) have a Javadoc document comment with at least a summary entry.

\*\*\* Make sure your name is within the correct source code files

Concepts

The program solution demonstrates requested concepts, notably (but not limited to) Android programming with Layouts, Fragments, Databases, AsyncTasks, ListViews and BaseAdaptors.

Compiles

The project provided compiles properly in Android Studio

Functionality / Execution

The program runs with the expected functionality with no crashes from typical user input

Problems?

An additional 4 points may be deducted from the score if the program is (for example) terribly designed, or crashes when receiving expected or unexpected user input.

Total:

The expected maximum score is out of 15:

12 points for the individual portion, plus 3 points for group portion

**Group Participation Grading (15%)**

At the end of the project, submit the following self and peer assessment rubric for grading, one for yourself and one for each team member.

Note: If you are grading ‘Exceptional’ or ‘Poor’ or lower provide specific details to back your claim for each person it applies to.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | | Interactions with group | | Reliability | | Attitude | |
| (Self): | | 3 2 1 0 | | 3 2 1 0 | | 3 2 1 0 | |
|  | | 3 2 1 0 | | 3 2 1 0 | | 3 2 1 0 | |
|  | | 3 2 1 0 | | 3 2 1 0 | | 3 2 1 0 | |
|  | | 3 2 1 0 | | 3 2 1 0 | | 3 2 1 0 | |
| Notes / Details: | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| (use more space if needed) | | | | | | | |
| Criteria | Exceptional(3) | | Meets Requirements(2) | | Poor(1) | | Non Participant(0) | |
| Interactions with group\* | Frequently uses constructive statements when interacting with others, attempts to help others often. | | Usually uses constructive statements when interacting with others, usually attempts to help others. | | Rarely uses constructive statements, rarely attempts to help others | | Uses negative statements, put-downs etc. when interacting with team members.  ~or~  Never attended a team meeting, didn’t interact with others | |
| Reliability\*\* | Attends almost all team meetings, if unable to attend always communicates to keep up to date with team | | Attends most team meetings, if unable to attend usually communicates to keep up to date with team. | | Attends some team meetings, if unable to attend occasionally communicates to keep up to date with team. | | Rarely attends team meetings, rarely communicates\* to keep up to date with team.  ~or~  Never attended a team meeting, never communicated with team. | |
| Attitude | Always has a positive attitude about tasks and is never critical of the work or other group members | | Usually has a positive attitude about tasks and is rarely critical of the work or other group members | | Sometimes has a positive attitude about tasks and is often critical of the work or other group members | | Always has a negative attitude and is frequently critical of the work or other team members  ~or~  Never participated | |

\* Help means help with debugging, searching for resources (textbook or online readings etc.), and general help with problem solving. Help does not mean extensive tutoring or doing another team members work.

\*\* When emailing the team, send the email to all team members.

Note: Include the lab professor (Eric Torunski or Abdul-Rahman Mawlood-Yunis ) as a cc (Carbon Copy) on any emails like this.

Project Team Activity Worksheet:

Lab Session \_\_\_\_\_ Group Number \_\_\_\_\_

**Introductions:** Introduce each other and share contact information, Algonquin College student email address at a minimum.

|  |  |  |
| --- | --- | --- |
| Team Members | Names (Print Clearly) | Email (Print Clearly) |
| My Name |  |  |
| Team Member: |  |  |
| Team Member: |  |  |
| Team Member: |  |  |
| Professor: | Eric Torunski | torunse@algonquincollege.com |

Basic Group Charter / Ground Rules:

* Expected time for email to be responded to?
* Who to email if I must miss team time (one person, everyone)?

(Tip: Also always email lab professor)

* How will I send my work to the group if I cannot attend a meeting?
* Any there any other items to discuss? Document them here:

Does anyone have external time constraints (work, family etc.) around out of class-lab meetings?

When working together do you take breaks? If so - how many and for how long?

(Etc.)

Project Assigned Productivity Application:

|  |  |  |
| --- | --- | --- |
| Team Members | Names (Print Clearly) | What Productivity App for Project 1? |
| My Name |  |  |
| Team Member: |  |  |
| Team Member: |  |  |
| Team Member: |  |  |

Code Freeze date and time for assembling the project: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Recommend 72 hours before due date at a minimum)

**Grading:** Uploading the finished worksheet automatically grants 5% towards the project; upload should be completed before the end of the week of Nov 23. Everyone needs to upload their own version of the worksheet to get credit.