

Project

EC327, Fall 2018

Assigned: November 7, 2018 Wednesday

(Tentative) Demos: *Week of December 10, to be announced*

Final Report and Code Submission Due: December 12, Wednesday, 10:00pm

(no extensions or late submission)

(Prepared in collaboration with Prof. Trachtenberg and Prof. Densmore)

Total points: 100 + *extra credit*

Goal

The aim of this team project is to design and build a marketable Android app. Your project will be graded equally on the following criteria:

Front-end

The user-interface for the project must be:

- crisp and intuitive (i.e., no instruction manual required), providing appropriate and understandable feedback,
- robust to all sorts of behavior, normal or otherwise, meaning your app should neither crash nor leave strange elements on the screen,
- visually pleasant, properly utilizing icons, backdrops, and graphical-user artifacts.

Back-end

The back-end code should be:

- efficient; it does not have to be optimal, but the user should not have to wait too long for any response,
- robust to all manner of input from the user interface. The back-end should never crash or lock in an infinite loop,
- secure (at a basic level), in that there should not be reasonably obvious ways to get the back-end to do something wrong,
- well-commented and easy to understand.

Marketability

Would people use the product?

- You should specify who might want to use the project, and for what purposes.
- It should be clear how the design decisions in the project conform with the potential use of the project.
- Your project should include a concise and clear description targeted toward the class of potential users.
- Students should make sure to stay away from licensed or copyrighted content (e.g. pictures, games).
- For *extra credit*, students can submit their project to the Google Play marketplace, with the number of downloads and app like votes (by the end of the semester) determining the amount of extra credit.

Sample project topics

Most teams typically suggest a project to the instructor for consideration. The following are given as samples, but feel free to pick a topic similar to one of the topics below.

Project 1: Rose-colored glasses

The goal of this project is to design and build a program that will show manipulated video taken through the Android phone's camera. Your project should have at least two *useful* video manipulation routines, for example:

rose-coloring

A routine to tint the video towards the color **rose**.

brightening

A routine that brightens camera video.

motion detection

A routine that highlights motion in the camera video.

Project 2: Gamefare

The goal of this project is to design and build an interesting Android-based game. Your game needs to contain:

- some significant back-end processing
- input from at least two phone sensors (including the touchscreen)
- a graphical display

Reasonable game options include, in increasing difficulty:

Tic tac toe

An interesting variation of the classic [tic-tac-toe](#) game.

Othello

A reasonably intelligent implementation of the game of Othello (otherwise known as [reversi](#)).

Go

A simple but visually interesting player for [Go](#).

Asteroids

A variant of the [classic](#) game (this is harder!)

Project Tasks

The project has five components:

1. Lead - This person coordinates the efforts of the project.
2. GUI - This person implements the graphical front-end used by the project, probably in Java/XML.
3. Interface - This person is responsible for programmatically connecting the various components into one project.
4. Processing - This person writes the back-end image processing, in C or C++ (or Java).
5. Documentation - This person manages all documentation and marketing for the project.

In five people teams, each task can be handled by one person. In four people teams, the recommended organization is to share the documentation task across all non-lead members in an organized way and have the Lead handle marketing. You can also arrange it such that two people jointly do two tasks, e.g., processing and documentation by two people or GUI and interface by two people. **Make sure to seek the instructor's approval on your task distribution to group members by November 14, Wednesday.**

Task Specifications and Grading

1. Project lead

The project lead is responsible for organizing and maintaining the pace of all work being done for the project. This includes:

- Setting up the time-line and objectives.
- Interfacing with all team members to maintain progress in development, including calling meetings.
- Checking all design specification for system-level problems and performing system-level integrated testing.
- Completing all component work that other members fail to do.

In other words, the project lead is ultimately responsible for making sure that the project completes on time. The project lead's component will be graded based on:

- Submitted documentation for timelines, objectives, and meetings
- Overall software architecture description (and images)
- The status of the project at the end of the semester.

2. Graphical User Interface

The GUI is responsible for interacting with the user. It should ideally be designed in Java using XML resources, as much as possible, to code interface elements.

The GUI will be graded based on being:

- crisp and intuitive - how easy is it for a user to understand what to do; is the feedback (e.g. error messages) clear and complete?
- robust - is it possible to crash the GUI or have it do inappropriate things (e.g., leave windows or icons on the screen, perform incorrect actions)?
- visually pleasant - does it properly utilize icons, backdrops, and other artifacts to improve the user's experience? Keep in mind that sometimes less is more (and sometimes it isn't).

Specific Hints

- The GUI should be implemented in Java. You can check out the [Hello World](#) tutorial to get a glimpse of how Android programs are organized.
- You can see some more examples at <http://developer.android.com/samples/index.html>.

3. Android Project - Interface Component

The interface is responsible for programmatically connecting the various components into one project. This includes:

- Setting up a version control system for maintaining all code used by the project (GitHub, Google Code, or other – you will be sharing the link with the course staff later on).
- Packaging all code into one application, with appropriate resources, and packaging it (and signing it if it is being put onto Google Play).
- Figuring out what data should be passed between the Android GUI (Java) and the native code (C/C++/Java) (and implementing JNI routines that will do it).
- Testing routines to demonstrate the proper passage of data.

The interface will be graded primarily on correctness (measured in part through your testing routines) and structure of the program.

Specific Hints

- The hardest part of this component will be simply figuring out how to use the NDK properly. The coding should not be significant.
- The [Android NDK](#) comes with a number of very helpful examples and an explanation of how to use them. We would recommend, for example, looking at hello-jni for a very simple example.

4. Project processing

This person writes the back-end image or game processing, in C or C++ or Java. The actual implementation should be decided by the group, but all computationally intensive processing should be done in this component. For example, this component might determine what move to make in a game of Go, or it may implement the image processing on an image provided from the GUI through the interface.

The processor will be graded based on:

- efficiency - the implementation should be efficient enough to provide a good user experience,
- robustness - the back-end should never crash or lock in an infinite loop; it should also always provide reasonable data to the other components,
- basic security - there should not be reasonably obvious ways to get the processor to do something wrong,
- commenting and readability of the code,
- testing regime - thoroughly test your code and provide comments/code for easy testing.

Specific Hints:

- We suggest that your code be thoroughly tested as each component is written.

5. Documentation

The documenter manages all documentation and marketing for the project, specifically:

- Putting together (and writing, if necessary) all documentation for the project, both front-end and back-end, in an organized, concise, and appealing fashion,
- Getting all team members to agree to a statement of work at the end of the project, describing what each member contributed,
- Putting together a short presentation of the project, including motivation for why the project is useful, who will use it, how they will use, and why they will use it.

The component will be graded based on the organization, clarity, conciseness, and appeal of all written material for the project. **Clear & comprehensive** documentation is essential for receiving a high grade.

Specific Hints

- We strongly suggest that you run documentation by not only group members, but also by a few people with no relationship to the project in order to check for the required elements.

Deliverables

Project teams are to submit:

- Documentation for all aspects (front-end, back-end, and marketing) of the project.
- All code for all aspects of the project, including instructions for how to compile and run the code from scratch. Include a sufficient number of test cases as part of the instructions.
- A **maximum 3 minute** video presentation of the project in action, including its architecture/development, target audience, and use.
- *We may arrange a project demo session (tentative, subject to resolving scheduling constraints).*

Common Hints

- You will need the following free software:
 - [Android Studio](#)
- Useful Links:
 - **Android Basics**
 - [User Interface](#) (Learn about Android front end)
 - [User Input](#) (Make your app interactive)
 - [Multi-screen Apps](#) (Intents, activities and fragments)
 - [Networking](#) (API interfacing using JSON)
 - [Data Storage](#) (Local storage with SQLite)
 - **Intermediate Courses**
 - [Developing Android Applications](#) (Overview of a relatively complex app)
 - [Material Design](#) (Make your apps look good)
 - [Location Services](#) (Use location services in your app)
- Requisite software is available on the PHO307 Linux computers, if you do not wish to use your machine. To access it, log onto any of the machines and follow the instructions at <http://collaborate.bu.edu/ece/android> to set up your paths appropriately.

Submission

Select a team name, identify the app you would like to develop, and decide on the task distribution. E-mail ec327fall18@gmail.com with your team name and task distribution by **November 14, Wednesday, 10:00pm**, with a subject line: ProjectInfo <YourTeamName>.

Email should contain:

TeamName: <YourTeamName>

App: <The app you will develop in 1-2 sentences at most>.

Team Members: <Full names, BU IDs, and BU usernames for each member>.

Task Distribution: <Match the 5 tasks given above with team member names>.

Do not include any other information in the email.

Submission instructions for the final report and code will be given later in class.