

**Project 3: Final Project**  
Android App Development  
100 points

In teams of twos, you are to create a non-trivial Android app of your choice that implements the given requirements. The app can be a game, utility, or educational in nature. Some examples include:

- Maze game that requires tilting the device to move a marble through a maze
- Painting app that allows you to load an image on your device, edit and save it
- Computing seminar attendance app that scans Harding IDs
- Harding app that tracks your movement on campus and tells you information about which buildings you are near

### **Project Proposal**

Each team must submit a typed project proposal by **Monday, Nov 13 at 9:00 am**. Drop the proposal off at my office. You may propose more than one idea if you'd like. I must accept your proposal before you begin developing. Your proposal should include the following:

1. Names of teammates
2. Description of what the app will do (one paragraph or more, bulleted list is fine) that includes how you will use a sensor
3. A drawing or screenshot of the app's primary UI screens

The proposal counts as a homework score. Proposals that are well-written with a clear description and well-thought out UI will receive full credit.

### **Class Presentations**

Each team will have 5-7 minutes to present their app to the class on the final week of class. I will clone your projects from GitHub on the podium computer so they can be quickly shown to the class.

Although the presentation time is very fast, do your best to impress! The impression you make in your short presentation will affect my evaluation of your app. Show how your app can restore its state during the presentation.

### **Requirements**

Your app must implement the following:

1. Have a custom icon
2. Uses a sensor (camera, accelerometer, etc)
3. Runs in a phone emulator

4. The app should be near the complexity of previous project. However, fragments are not necessary, and the app may run in just landscape or portrait mode; no need to implement state-saving code.

## Teams

You may form your own two-person teams. If you are unsure about who to work with, you could post your idea for a project on Canvas and see if anyone would like to work with you.

Teams should create a private GitHub repo and regularly save their work to the repo. Add me as a contributor to your repo (username: fmccown). Your source code on GitHub should not be modified after the deadline to avoid late penalties. It should include all the images and sound files used to create the project. I will clone your repository on my machine to grade it. One teammate should submit the team's README file to Easel before the deadline; there is no need to submit anything else to Easel.

The repo's README file should contain the following information:

1. Link to the repo on GitHub
2. Brief summary of what the program does
3. List of any known bugs that still remain
4. List of each teammate's contributions
5. The percentage of work performed by each teammate

Ideally the work performed by each teammate will be 50/50, but if one teammate does more than the other, the percentages should be adjusted accordingly. The percentages will be used to determine each teammate's final grade on the project. The repo history on GitHub should be a fairly accurate indicator of how much work each teammate did.

## Grading

Each team will meet with me for 20 minutes the final week of class or during finals week. I will ask you questions about your project, like why certain design decisions were made, what do you wish you had done differently, and how much effort was put into the project. The effort put into this project should match the effort put into previous projects.

Other criteria:

1. Does your app implement the functionality set forth in the proposal?
2. Is your program well written? Did you use proper Java naming conventions for variables, methods, and classes? Did you use constants and functions when appropriate?
3. Did you clearly cite any code that you found online?
4. Did you use git adequately when developing your solution? There should be at least 20 commits in your repo, and each person must have committed at least twice.