Recall that the three key goals for this project are:

1. **Learn something new (related to Java, CS, technology) on your own.** Ideally documenting that learning for posterity. (At a minimum, this should be *new to you and your collaborator(s),*but it could even be new to the knowledge of humanity. That would be cool.)
2. **Build something real.** Which may push you to a language other than Java. (Real, of course, within the constraints of computer science -- downloadable, but not necessarily touchable.)
3. **Collaborate.** Either within the class, within the school or externally. Nobody *really* works alone. (Past experience suggests that Mr. Battis will be really hard to convince on collaborations beyond the classroom.)

Building on our preliminary project ideas and our conversations, construct a document with your partner(s) that proposes the (current version of) your project. Key items that you will want to include to both be descriptive and plan ahead include:

1. **A pitch.** Describe the big idea of what you're doing. You don't have to go into pages of detail, but a good abstract and a detailed explanation are necessary. This should be fairly specific -- it shouldn't be a generic description of projects like your project, but an actual description of *your* project. *We'll end up referring back to this as the spring progresses as a "centering" document: Are we still on track towards the original goals? How far have we progressed towards these goals? Do we need to adjust the goals?*
2. **Envisioned deliverables.** What do you anticipate producing as a result of this project? When? Be as specific as you can be at this moment in time. Obviously, as we learn more and develop the project, these deliverables and their descriptions will need to shift and change, but put some initial benchmarks out there. Identify which of these benchmarks might be considered as a "finished" project (and why), knowing that many of you are considering very ambitious projects that may, or may not, require more time than we have this spring. *We will refer back to these as the spring progresses, to make sure that your project is on track to completion.*
3. **Action Plan.** What do you need to do to produce your deliverables? What will you have to learn? What will you have to make? What will you need to buy -- and how much will it cost (knowing that we do not have an unlimited budget)? When will these things happen? How long will they take? What has to happen in which order? Why? What absolutely needs to happen first? *We will transfer this into a project-management tool so that we can track the progress on your project and adjust accordingly.*

Pitch

We want our project to be making a phone game designedfor iOS. We want this game to contain many different levels that are made up of random obstacles the user has to pass. Our goal is to have a unique combination of different game themes and platforms to make our own semi original game idea. The game is going to be a sidescroller that has some sort of randomness implemented into each level’s obstacles that will make the game challenging to play. The game will hopefully be relatively easy to learn to play, but difficult to master the game.

Benchmarks & Action Plans (Total time for project: 13 weeks)

- Game Planning (1~2 weeks)

In this step, we will establish and narrow down goals set for this game by researching and exploring different options and themes of phone games we could pursue. We could map out each level we would want, establish a scoring system, and develop game elements and features, such as the random obstacles. To aid in this process, some researching game design and planning steps may be needed. We should keep in mind that the ultimate goal is publishing a working game on the App Store, so the game should be designed to not violate copyright and be appealing for users to download.

- Learn About Platform (Unity) (2~3 weeks learning)

We will be using Unity, which is a professional multiplatform game development engine that supports both 2D and 3D experiences. We will take some time to explore and learn about Unity and what kind of opportunities are available and unique to it. We should think about how we can take advantage of these unique opportunities, and maybe even go back and revise game plan. This would be our “learning something new” part of the project, as we will have to learn a new language to operate Unity. After assessing Unity’s functions and comparing them with our desired game plan, we should decide whether we need the professional version of the program. We should also update our action plans with more platform specific goals.

- Art (<1 week)

After learning about the platform and establishing what our goals are, we will decide what kind of art would be interesting and appropriate for our project. Additional research about what kinds of art options are available in Unity may be needed. Discuss and decide what graphical style we want to use, and whether we want to create our own art or if we want to ask another person to help us with the art. Possible options at this time include 3d models, 2d sprites, etc.

- Flowchart & Storyboard (~3 weeks)

We will use flowcharting software introduced in class to map out the game. In our flowchart, we will need to discuss specifics of each step, such as details for levels and title screen. Flowcharting will help us plan each step of the game and realize if we missed any steps during our game design period. After flowcharting, we will storyboard each screen and level that the game will display (taking into account that many will be randomly generated). After we know exactly what elements and features we will need in the game, we will take some time to find the art, or discuss placeholder models, that we discussed in the previous step.

-- CODING -- (~7 weeks; very rough estimate)

Obviously the game needs to be coded, but not much can be said at this stage because we don’t really know about Unity. The coding part will become more clear and broken down as we plan the game and as we learn about the platform.

\*\*One concern/question we had was how we should collaborate with each other during the coding process. We should give some thought to how we would group our work together when we work on the code from different computers, and set up/learn how to proficiently use GitHub to achieve that goal.

- Animation (1~2 weeks?)

We will integrate graphics and art into the game. We will learn and start to work with different animations that would work with the art style we would choose. (Animation, if we are in a time crunch, could be a step that we can cut out.)

- !!!!!Working Game!!!!!

This is the first major benchmark that can be considered a “finished project”. This is our major deliverable for our project. An acceptable end product would be a basic, but functional game that could run on the Unity iOS Simulator App. Goals for this game, as mentioned in the pitch, would be a sidescroller that has randomness implemented into each levels obstacles, making the game harder to play. If all goes as expected this will be a fun and entertaining game that is easy to play, but hard to be good at.

- Possible Additions:

add network aspect / connect to Gamecenter / multiple players / better graphics / more levels

These are some options for possible additions for after we make the game that would make the project more advanced, if time allows. This would be additional deliverable if we achieve the working game

- App Store

After any additions are made, our final goal is to release the functional game on the app store, so everyone can play our awesome game. During this step, we should give some thought to how we could (potentially or for reals) make profit off of our game. Some ways to profit includes charging money per download or charging for microtransactions. A game that motivates users to download and can potentially make profit for us as a real game on the App Store would be our envisioned optimal end goal.

\*\*Additional thoughts about team dynamic: We should discuss how our team would work best: splitting tasks (maybe even Benchmark steps) up to independently complete, working through each task together, or a balance of the two.