

# Project 2 Draft

## Team 2

Names: Ian Boyer, David Carr, Samuel Costa, Maximus Latkovski, and Jy'el Mason

Criteria	Maximum	Score
Introduction	5	3.0
System Architecture	10	8.0
Data Design	5	3.0
Component Design	15	13.0
Interface Design	20	19.0
Prototype	20	12.0
Milestones	15	10.0
Spelling, grammar, style, and formatting	10	8.0
Total	100	76.0

## Comments:

Fair job.

Your introduction is purely material from your requirements document. You should discuss that this document is for design. Also, you don't mention the intended audience of this document. (Which you didn't do on your requirements document either, but it looks like I didn't catch that.) I think your definitions and other descriptions from before would make sense here. You kept the part that was supposed to change and left out the part that could have stayed the same.

For your system architecture, I like the balance between simplicity and information you give in your block diagram. I wish you had specifically said that you were using MVC in your text. I think your Views should list some or all of the views you will manage. I don't understand why there are a separate Game Control and View Controller components. The Game Control seems like it should do the job of the View Controller as well. Why have this middleman? I think the Network/Player components are probably part of the Controllers.

In the data design, there is some vagueness about how the data will actually be stored. More importantly,

I want to know exactly what will be sent across the network. Whole objects? Shorthand values? I recommend that you address all comments in a document and remove them before submitting the work.

The component design is pretty good. A lot of information is shared in the format you used. However, a simple class diagram would be really helpful to visualize the connections between classes. Again, some comments made by the team could have been addressed before submission. Consider all data structures carefully. Lists and Sets are sometimes more efficient than arrays.

Your interface design is thorough. I recommend removing all back and quit buttons, relying on the X button in the upper right of every window (in Windows).

The first problem with your prototype is that you don't explain it at all. You don't tell me what I have to do to run it, and you don't tell me what I should expect to see when it runs. The second is that there's not enough there. I applaud the approach to put off GUI since students often get swallowed up in GUI without getting any real internals done. However, your code seems only to partially test if a given word is allowed. It doesn't even load a dictionary. As I mentioned to someone in your team, all these methods that throw `InvalidPositionException` should really return `true` or `false`. Throwing an exception is an emergency procedure that's quite expensive in terms of run-time and isn't needed when we're simply asking a yes-or-no question.

There are a number of problems with your milestones. The first is that the implementation is supposed to be done by November 8, and all of your deadlines are after that. You're getting started quite late on your GUI. You don't have any testing tasks on your milestones. You break down your tasks in the descriptions, but you don't show those broken-down tasks on your Gantt chart.

There are some ungrammatical and confusing sentences in your document. Your code style is pretty good, though everything should be in a top-level package for your project, something like `scrabble`.