

This assignment is due by the start of class Tuesday, Oct. 2.

Complete your beta! Following are the requirements.

Functional Requirements:

The beta requirements incorporate the alpha requirements (as described in the 9-18 lecture) and the following options. You should choose options that add up to at least 100 points. You may earn a maximum of 130 points for these options.

- Grid generation: 100 points
- Options worth 10 points each
 - Win/lose condition/message
 - Erase cell (immutable only)
 - Play again
 - Darken grid cell momentarily when selected
 - Flash X momentarily when player tries to place inconsistent number
 - Show/hide consistency check for current number
 - Show all invalid cells
 - Erase all invalid values
 - Show how many times a number needs to be placed
 - Juice
- Super juice (class votes on points)
- Other ideas (see me)

Use cases and algorithms were discussed in class. See lecture notes for details.

Supporting documentation:

In addition to your code you should submit:

- Uses cases: You need not reiterate the details of the uses cases that I specified, identifying them by name is sufficient. You should, however, describe your use case for entering a number in the grid, since there are two models for that behavior (choose number then a cell and vice versa). You should also describe your use cases for any functionality that I didn't specify (e.g. erasing a cell) or that you handle differently than how I specified it.
- UML class diagrams: The diagrams you used in your walk through should be updated to reflect the added functionality of the beta.
- UML sequence diagram: A sequence diagram explaining the enter number in grid use case.
- Coding standards
- README: describing the version of OS and Xcode needed to run your project.
- Critique: A brief report describing what you learned in the walk through and what you did (or did not do) to your code/design in response to the critique.

You should also report any remaining known bugs, design problems, etc.
Note: you will get more credit if you identify problems with your beta than if we find them on our own.

Grading (350 pts total):

Your project will be evaluated for functionality as well as design and implementation quality (e.g. following good design principles, having clean code, etc.). Your documentation be evaluated for professionalism, clarity, conciseness, and consistency with the actual code. (Hand-drawn diagrams are *not* appropriate at this stage.)

- Basic back-end design/implementation (100 pts): alpha functionality of model and model-controller interaction.
- Basic front-end design/implementation (80 pts): alpha view and view-controller interaction
- Beta options (100 pts)
- Supporting documentation (50 pts)