6.170 P2.4 Self Review & Critique
Carolyn Zhang (carolynz)

## Summary assessment from user's perspective

Overall, the site is easy to understand, and the core elements of the site all work as expected. It's easy to create a new note and edit its properties.The automatic note arrangement and rearrangement may have been too confusing for some users; perhaps I should have included an explanation somewhere on the site.

The persistence of the note-editing toolbar when the note is deselected is also another design decision I should have better explained within the app. A good workaround would have been to include a "close" icon for the toolbar upon hover.

## Summary assessment from developer's perspective

Controllers are skinny and models contain the bulk of the (very little) backend logic. My Javascript code was a bit of a jumbled mess, and could definitely be refactored. The code was also insufficiently-commented, and should have had more detailed specs.

There was also not enough separation of concerns. I stored a note's formatting by storing its CSS class selectors as strings in the database. For example, the "height" column in the Note table would store values like "noteheight1," the CSS class for a note with a height of 1 unit. I should have stored these values as integers and used methods in the model to construct the appropriate string value before passing them to the view.

One major mistake that my peer reviewer pointed out to me was that I forgot to include authorization in the controllers. Between P3.2 and P3.3, I accidentally deleted the `load_and_authorize_resources` line in the controller that would have controlled user permissions!

## Most and least successful decisions

I think my most successful decision was making the notes automatically-rearrangeable but still have customizable dimensions. It creates a very smooth and easy-to-use interface that I would personally use.

My least successful decision (other than the dreadful `load_and_authorize_resources` mistake) was probably not taking more time to structure my Javascript code properly. Debugging was quite a headache and took me several days, all due to my poorly-written code!

## Priorities for improvement

1. Adopt a better testing methodology. I need to write tests as soon as I add new functionality to make sure things don't break; simply testing the app manually and then writing tests at the end is not enough!

2. Writing more consistent and thorough specs so the code is more readable. This makes it easier to understand my own code design, during both coding and reviewing.

3. Learn how to write better, cleaner, more maintainable Javascript!

## Other reflections

While working on P3.3, completely scrapped my code for a Hashtags extension and added in a formatting toolbar functionality instead. When I switched, I had wanted to let users be able to upload note background images, change text and background colors, and add drop shadows to the text, so that users could create a collection of visually-appealing notes. I only got to implement very basic text-editing function before I had to stop due to time constraints. I'm not very satisfied with the end product; I regret changing course so late to create only a half-baked result.

Another contributor to my time constraint was high number of bugs I ran into due to my sloppy Javascript code and general lack of familiarity with Javascript/jQuery. I sort of let my code spin out of control into a disorganized mess, which did not help at all.