CS BS Requirement Visualizer

Errica Cheng, Dyllan Cole, Alan Curiel, Stokley Voltmer

Overview

The main goal of our project was to produce a visualization of the requirements for a BS in Computer Science at WPI that was intuitive and easy to use. We aimed to visualize the requirements in a treemap. Our original project proposal was as follows:

> "Viewing degree requirements in Bannerweb is confusing and unintuitive. I think that it would be of benefit to many CS students to be able to visualize the degree requirements and paths to completion. Data will come from the CS tracking sheet and course catalog, as well as allowing students to upload their transcript and see their requirement progress in a Treemap based visualization. There will also be ancillary views to allow slotting in courses. I think it will be challenging to calculate which courses count towards what and where to display them. The interactive elements could also be complicated."

Our primary motivation was the dismal state of Bannerweb's degree evaluation page, although WPI is soon switching to Workday Student to replace Bannerweb. To that end, we see this project as more of a fun little tool than something that we intend for people to actually use.

Related Work

We were inspired by the CS tracking sheet, Bannerweb's degree evaluation feature, and the schedule planner among others. For the actual treemap itself, we drew inspiration from d3's example treemaps.

Questions

The main question that we were really trying to answer with this project was "How can we visualize CS degree requirements in an intuitive manner?" We knew right off the bat that we wanted some sort of treemap-based visualization, but we had to determine the specifics of our implementation to maximize the utility of our project. In the course of implementing that treemap, many other questions came up:

- How do we display degree requirement categories?
    - Answer: Large labeled grey box

- How do we display courses?
    - Answer: Small faded purple box
- What is the best way to indicate a completed category?
    - Answer: Highlight around edges
- How do we size the treemap nodes?
    - Answer: Proportional to credit hours

Data

Our visualization draws on two primary pieces of data: a handcrafted JSON file describing the contents of the CS tracking sheet to determine requirement categories and total credit hours, and a listing of all courses offered at WPI. The listing of courses was obtained by parsing through the Bannerweb course catalog for the last eight semesters, then running the data through a tool that deduplicated courses.

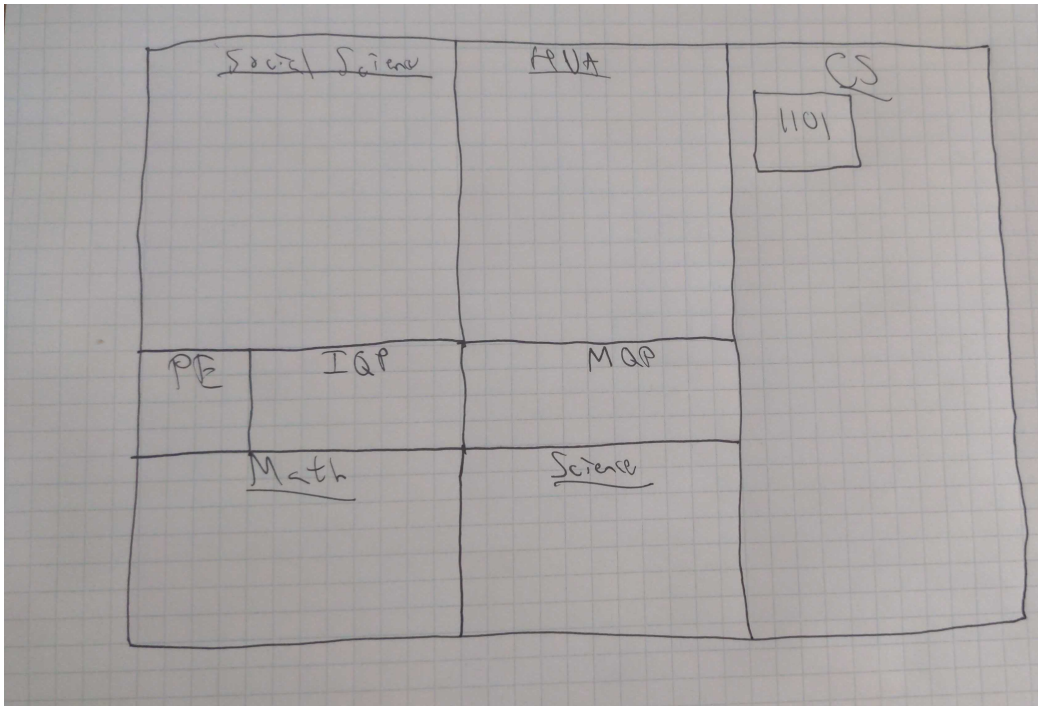
Additionally, our tool has the capability to import data on what courses the user has taken from Bannerweb. This is accomplished by having the user enter their credentials, then sending their credentials to a GCP (Google Cloud Platform) function. The GCP function then posts their credentials to the login page to get a session ID along with some spoofed referrer headers to convince Bannerweb that we are logging in from Bannerweb itself. We then use the session ID to request the user's unofficial transcript page. Finally, the GCP function parses through the contents of the user's unofficial transcript and returns a list of courses that the user has taken to our main program.

Exploratory Data Analysis

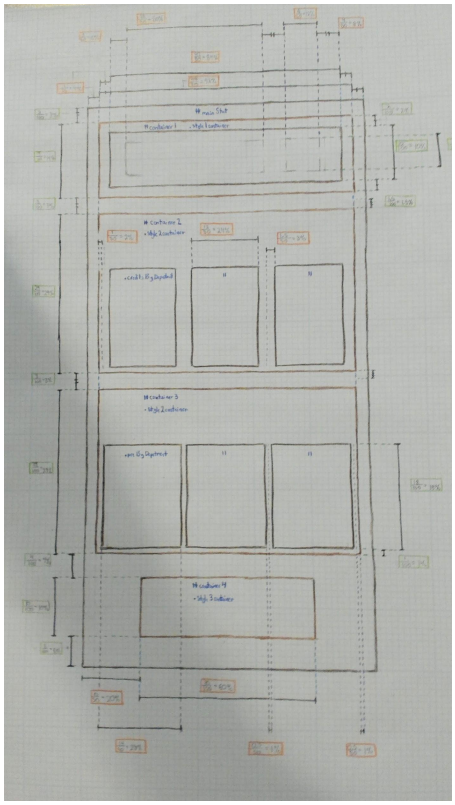
Our data consists of the course requirements for CS majors at WPI, so we started by looking at the tracking sheet given to CS undergrads and the visualization of computer sciences courses in the undergraduate catalog. The catalog was grouped by prerequisite and degree requirements. We did not end up using this information, but it allowed us to explore the possibility of including prerequisite information.

Design Evolution

A major concern of our design was ensuring that it did not overwhelm the users with a flood of information, while at the same time being able to give the user a broad overview of their courses. To this end, we settled on a tree map to contain all the courses, seen here in its earliest iteration.



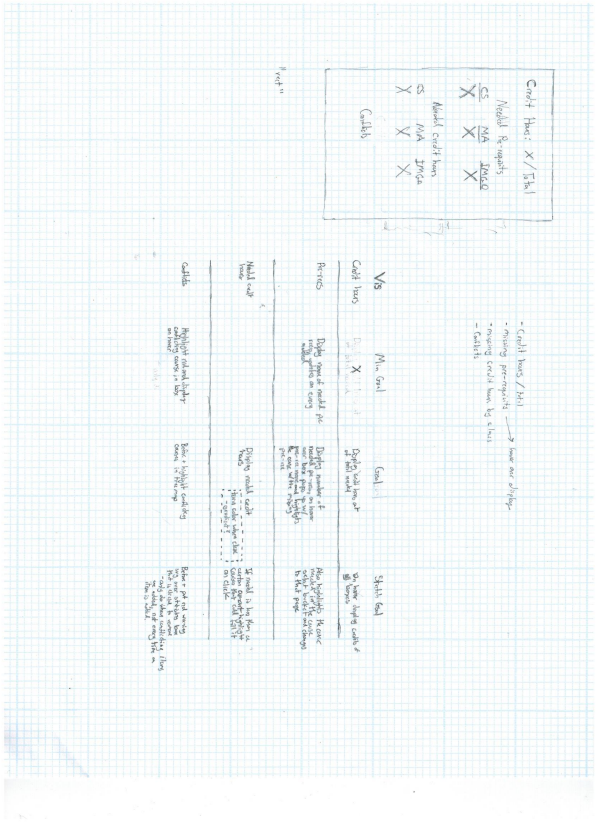
Another concern of ours was that a tree map would lose the ability to see data one might typically expect from a course overview. As such, in order to maximize the utility of our visualization, we needed a box that could display statistics about the courses. The box, which we called the statistics box, can be seen in its planning stage below and to the left.

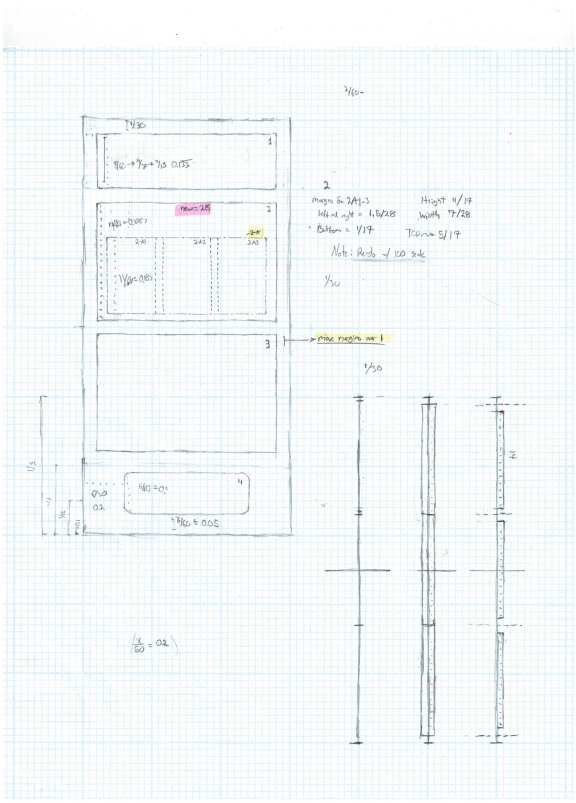


The statistics box is populated with information regarding credit hours. Specific attention was paid to credit hours for MA and CS courses, which the group decided would be the most useful to see up front.

The statistics box was originally going to hold a button (seen on the bottom of the image on the left) that would turn red when the user input duplicated classes to the treemap. However, this idea changed in our final iteration. Instead, a pop-up warning displays now when the user attempts to add a duplicate class, stopping the problem from happening in the first place rather than attempting to fix it after the fact. The decision was made to both populate the treemap with imported classes and to allow the tree map to be populated by dragging and dropping in classes. This change allowed for greater flexibility and user experience.

The below images are of notes on the relative sizes for the elements in the statistics box, as well as an early plan of action for the statistics box. In the end, much of the brainstorm was not implemented in order to avoid inundating users with information.





The courses were initially going to be held in expandable folders on the bottom of the page; however, it was decided that it would be better for the user to be able to see a simpler view and not have to worry about opening and closing folders. As such, the class container turned into a box that displays one department's worth of classes at a time. The departments are then chosen via a scrolling list on the left of the box, which is far more intuitive.

Implementation

- TreeMap: The tree map contains the current courses a user has inputted from either Bannerweb or the catalog. It separates into required categories from the tracking sheets. The sizes of those categories are based on the amount of credits required.
- Catalog: This catalog shows all the courses at WPI separated by department. You can drag a course into the tree map area, and the tree map will sort out where the course belongs and how big to make it.
- Import: The import section allows the user to sign in to their bannerweb and extract course information Which will propagate the tree map.

- Statistics: The statistics bar shows the current count of credits earned and what is still needed based on the courses in the tree map.

Evaluation

In our final version of this project, we are able to see what courses we have taken and how close we are to meeting requirements. It works by reading a list of course objects and updates everytime we give it a course or import from Bannerweb.

If we had more time, one major change we would make would be allowing the user to zoom into departments more naturally. We would also allow them to see prerequisites.