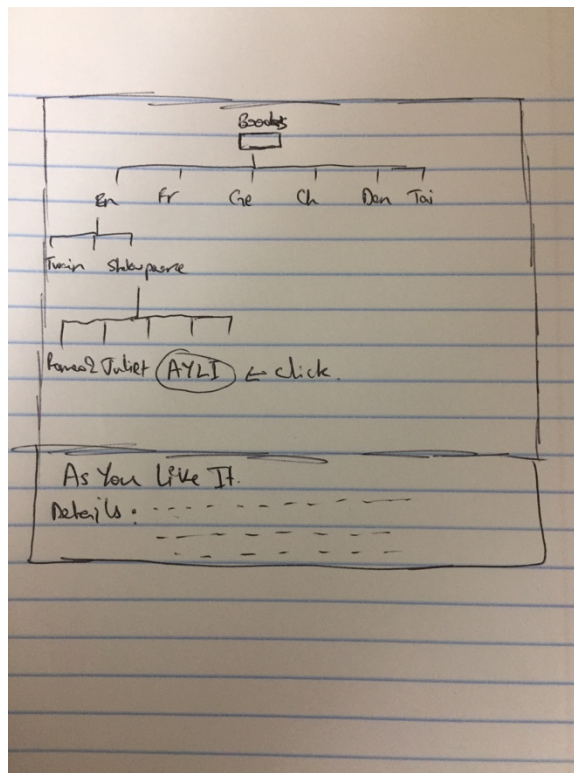


Assignment 3: Creating Interactive Visualization Software

Pre-Visualization

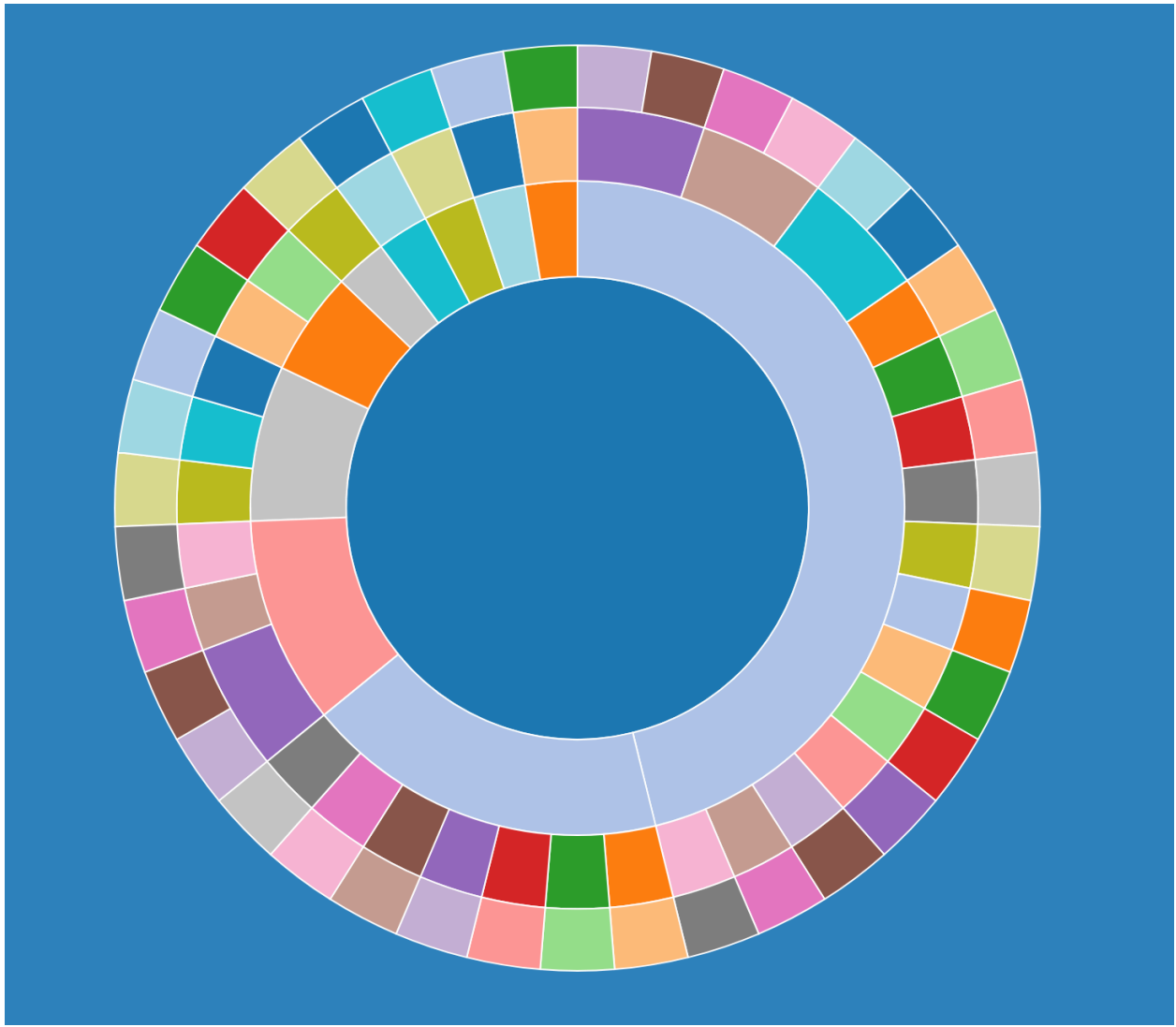
Our dataset¹ is a collection of the top 1000 most popular books on Project Gutenberg (as determined by downloads). Each book has information about its authorship, publication date, title and a few other fields. Project Gutenberg is a volunteer effort to digitize and archive cultural works. Most of the texts on Project Gutenberg are classical old texts that have lost their trademark, and hence can be digitally uploaded for free. From the first look at the dataset, we plan to create a tree like visualization where from each author we can see the books, and what language it was written in. When clicked on a book, a small section below will show information about the book, such as publication date, the subject of the book etc.



¹ <https://think.cs.vt.edu/corgis/json/classics/classics.html>

We decided on a tree like visualization because it is easier to understand hierarchy in this way, and humans are used to this idea of tree structures, hence not requiring any special understanding of the visualization. The interactive techniques we plan to use is dynamic querying where clicking on a value will expand the tree.

Post Visualization:

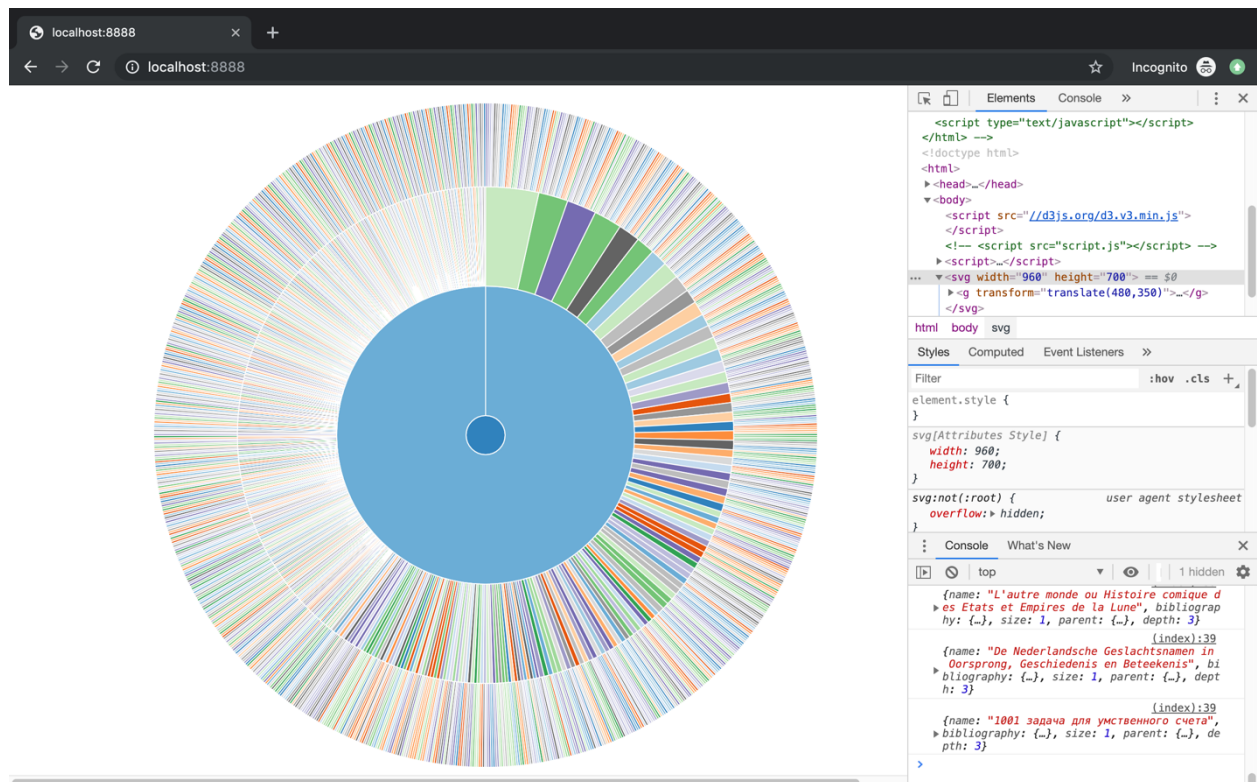


The final interaction we resorted to make differs in many ways from what our initial idea was. We initially planned to make a tree, but instead we resorted to making a sunburst plot. There

were many issues we faced from the beginning of the development phase, majority of the problems coming from the data itself. The data was in JSON format, but the data itself had many fields. Most of them which we didn't require. To get rid of them using Javascript took us quite some time, and to reorganize the data in such a way that d3 could read it also took majority of our time. The dataset contained books of 11 languages, English being the leading. The English language had around 300 books, while German was the second leading language, and it had only 30 books. Due to this we decided against making the tree structure. A tree structure of that proportion would be very unclear.

On looking through the internet, we stumbled across a sunburst chart. A sunburst chart does exactly what a tree structure does, but it's much more interactive, and can contain many more details in a compressed manner. The zooming in, and zooming out help in seeing even the small values of a data that might occur.

While coding the sunburst viz, we faced the problem of unproportionate circle partitions. English had the most number of books, so it took more than 90% of the proportion of the ring, while the other 10 languages divided the rest of the 10%.



This resulted in a very clumped up, and very unclear visualization, one which confuse the user more than help them. Hence the decision we came to was to remove all books of the English language, and just focus on the foreign languages. This resulted in a much more even visualization throughout. We also had to scrap the idea of indicating the details, mainly due to the difficulty of using the dataset, and we also felt like it would exceed the scope of the assignment. The visualization techniques we ended up using were dynamic querying, and tooltips. The biggest drawback to this was the fact that we could not indicate names on the sunburst chart itself (other than the tooltip). This was mainly due to the problem of some books having very long names, such as, Die Leiden des jungen Werther – Band 1.

Work Distribution:

Initially for choosing the visualization we took the approach of parallel prototyping, where both of us came up with storyboards for our designs, and then came together to discuss the best

one, and that is how we came up with the tree viz. Later on as development started, we worked separately to figure out what visualization to make (since we scrapped the tree). Later once data manipulation/cleaning began, we work together to increase productivity, and decrease the amount of time that would be spent in communication if we were working separately. Overall it took us around 6-8 hours to make this visualization.