CSE 512 Final Project
How to Pick an Ideal Major

Warren Shen, Thomas Lin, John Cheng

Abstract—
visualization result is combined with interactive features, animation and visual augmentation approaches. There are two main diagrams in the project: Integrated Donut Chart and Animation Bar Diagram. A total of 173 majors are analyzed in this project. The visualization is built on Observable [7], mainly by d3.js [6], in JavaScript and HTML. The dataset is College Majors [8] - Graduate Students (25+ ages) from American Community Survey.

Index Terms—Interactive Visualization, Animation, College Major, D3.js

1 Introduction

Major selection is one of the most critical decisions that nearly everyone will encounter in their life. Since major selection is heavily related to one's interest, family and future career, picking an ideal major turns out to be a rather difficult problem. In common situations, high school and undergraduate students may pursue their ideal major with scientific exploration and by researching. As a result, students might end up making an inappropriate selection, thus leading to some negative influence in one's life.

To solve the problem, the goal of this project is to provide a simple, intuitive, informative and fun system to assist users to get familiar with their potential major. Based on income and major popularity, the visualization system will represent ranking and sorted result for user to explore and compare the information in multiple major categories. We expect the system to provide sufficient information for users to understand what the real world looks like, and how the industry works. By knowing the potential income and job opportunity in the future, the system can be helpful for students to make an appropriate decision while they not yet entering the certain department.

In traditional static visualization such as bar chart or list, it usually cannot represent all the major in once since there are simply too many. In this case, users may not receive enough inspiration before they feel bored. On the other hand, the information of each major is usually separated on different websites or documents. It takes time to gather the information and generate a persuasive analysis. As a result, one of our major targets is to simplify the complicated raw data among all departments, while still representing comprehensive information and result at the same time. To achieve such features, there are two direction we plan to implement. The first one is to simplified the searching process. By categorized each major into groups, the data can become hierarchical and easy to access. With this feature, more detailed information can be shown on the inner layer of the visualization. Secondly, we plan to reinforce the first impression of the majors that are more competitive and promising. The main approaching is to generate a straightforward ranking by either animation or other visual augmentations. In such manner, the user can reach the target data in which they are more interested.

Through the research and evaluation results of the project, we found that dynamic data representation approaches, such as interactive features and animation, being more effective and attractive than traditional static visualization. Hence, we implemented several interaction features including tooltip, click events and animation control user interface. Through the various of shape, color and data transformation, we expect the user may more willing to explore the system, and therefore get sufficient information after using the system.

- *Warren Shen is with the Department of Mechanical Engineering at University of Washington. E-mail: yrsh20@uw.edu*
- *Thomas Lin is with the Department of Mechanical Engineering at University of Washington. E-mail: tzlintw@uw.edu*
- *John Cheng is with the Department of Mechanical Engineering at University of Washington. E-mail: john226@uw.edu*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

2 Related Work

In this section, we listed some works which inspired us to construct the blue print of the visualization. The visualization of the project is a variant of the Sunburst [2] and Icicle [4]. The interactive feature and animation are inspired by Pokedex [11] and Brawltime Ninja [12].

2.1 Pokedex, a Pokemon Dictionary

One of our major related works is the Pokedex, Pokemon sorted by properties. It is a fairly straight forward concept of an application of visualization. There are two completely unique parts for this visualization. Pokemon, abbreviated from Pocket Monster, is a Japanese media managed by The Pokemon Company, which is a Company founded by Nintendo, Game Freak and Creatures. The main part of the franchise is centered on the fictional creatures created by Satoshi Tajiri called Pokemon, everything else in the Pokemon universe revolves around the Pokemons, such as catching the Pokemons, training them, and also raising them for combats or keeping them as pets.

The whole purpose for this visualization graph is to showcase and display the various 'Types' or 'Egg Groups' among different Pokemons. In the first part, since for a Pokemon, there are a lot of times not only one group that it belongs to; consequently, there would be a variety of combinations of egg group to which one Pokemon belong. There are mainly some basic interactions to let the users choose between different combinations that we want to aggregate. There is also a dropdown selection on the bar to simplify the process of 'egg group' selection. Meanwhile, the individual bar description is also moved to a tooltip to increase the number of Pokemons that could be shown in a single graph. The design of the graph is focused on simplicity, users can make comparisons between selected Pokemons and remove unnecessary distractions along the way, such as the option that the check box provides to hide or show the individual properties. The whole graph is also zoomable to increase the flexibility and clearness across users who has different comparison purposes.

While the first graph has a more detailed approach, the second graph is much simpler, but still provides a sophisticated and high-standard approach of the visualization for the Pokemons. The second graph takes an approach of a donut chart, which looks intriguing before we even started exploring the interface, The donut chart, also called zoomable sunburst graph, links the close relationship between Pokemon 'types' and Pokemon "egg groups". The interactive chart zooms in when you click on any properties on the chart, and it goes into the next layer of that property, the chart goes in with a seamless, well-designed animation, the layers go deeper and deeper until it reaches the end which is the actual names of the Pokemon, after navigating through two completely different type/ egg group selection.

We took into account and reference of quite a number of concepts from the Pokemon index, especially the layering of the zoomable sunburst chart. It is designed to wrap the different aggregations that we as a visualization designer choose to present. The radio button from the second graph that lets the users to select which category to compare is also one of the key designs that we adopted; the framework was decided soon after we went through this visualization graph.

2.2 Brawltime Ninja

The next reference that we considered is the brawl time ninja, which is a third-party information website from a popular mobile game, brawl stars. The game Brawl Stars is a multiplayer online battle area, with a third person, bird's eye view, hero shooter video game that is developed by the Finnish video game company supercell. In this game, there are different characters and different abilities for each character. There are also different maps and different modes that favors different characters. The key decisive part of this website is the large amount of data and the fast-refreshing rate of the data being shown for each character. The visualization for this project is written in CSS/ HTML. Which is also mentioned in one of our latter TA's lectures on Fridays. It could be broken into three main parts. The character tier list, the map tier lists, and the profile search. Each part plays an important role for the visualization project, and so are they equally important.

The first part, the character tier list, shows the win rate for every character. And due to the large numbers of the characters, around 60 characters, for the character selection part, there is a smaller sliding bar which shows a less pixelated graph of the character, and also a larger sliding bar that shows a more high-res image for the same character and the name for it on the side. The smaller sliding bar is clickable and shows you the neighboring six characters of the character that you click in the larger sliding bar area, the user can then select the larger sliding bar for the selection of that character. The concept of the double sliding bars, the big one and the small one, is a carefully designed feature, it relates to the webpage scrolling bars on the right-hand side of a normal search engine. We could utilize the small scrolling bar vertically on the right of a browser to locate the part of the browser we wanted. The difference is that the character selection graph has a small graph on the parts that the user is currently selecting different from the browser toggles that only has a plain toggle of the side of the tab.

The second part is the aggregation of the different characters in terms of their win rate, so not only could the visualization show the win rate of an individual characters, it also has the option to show all the characters side by side, and rank them with the win rate from high to low. A neat design here is that in order to simplify the visualization, the win rate chart only displays 10 characters at a time, starting from the top 10, since for most users, it is a lot of time only important for them to know the top characters that gives the best win rate, and for the little portion of users that actually want to know the rest of the list, they simply have to click on the 'next' button on the chart. The design could greatly improve the tidiness of the whole page, and emphasize of the data that is of the most interest of the users, the top win rate ones in this case.

For the second part of the project, the map tier lists part, there are also some noticeable features for the design. For some background knowledge, in the game, there are 10 different modes and for each mode, there are 7 maps that rotates daily, in order to form a one-week cycle for the game. In total, there are roughly 70 different maps in the game. For each map, the website has access to the player database that gives the win rate for every character for every map. Even though there are 70 maps in total, the project manages to keep it as simple as possible, and there are always only 6 maps in display at a time. The author did a crucial thing to achieve this goal, which is prioritizing on certain maps to show in the first place. The priorities being the current maps in rotation for each mode in the first row, and the upcoming maps after the current maps being on the second row of the visualization. It focuses on a lot on the relative maps instead of the other ones. Since for a user, you would only focus on the maps which is currently on rotation, and the next one in line of the rotation as well, the other maps are relatively less important in this case. For each map, it also shows the characters that has the highest win rate all the way to the lowest win rate. The same visualization design of the top 10 characters is adopted just as it was in the first part, showing only the top 10 character in the first page to save space.

The last part of the project is the profile search part, which could show all the data for a specific user as long as the player ID is provided. This is the part where the database has to be easily aggregated and fairly up-to-date, having to show all the information of a placer, the data management for this visualization project also plays an important part at the same time.

Conclusively, this visualization project has a good amount of bullet points in terms of its designing scheme, despite the fact that our team is not so familiar to CSS/HTML. The toggling menu for the character selection is a neat design takeaway, as well as the importance of the data refresh-rate. For a visualization to be so well-synchronized, a lot of work is being put in for the backend part which is the database management system, as well as the periodic maintenance of the map updates once a new map or a new character is introduced to the game.

3 Design Goals

In order to pursue the effectiveness and performance of the visualization, there are four core targets we would like to achieve: Simple, Informative, Intuitive and Fun. The first goal, Simple, is to organize and simplify the complex information we may get from various resources. We aim to provide a clear structure which is easy to access and deliver our idea in the first glance. The constraints of the goal are that sorting and presenting different kind of data at the same time is not easy. For example, median income and major population are two independent attributes. How to combine and plot them onto a diagram is an issue. The requirements of this target are to reduce and sort the result to minimize the amount of information being displayed at the same time. In addition, the interaction or animation features should not be too distractive or combine multiple elements. With such manner, the user can concentrate on what the main idea is presented.

The second goal, Informative, is to provide sufficient information that reader may be interested in. That is, to provide enough layers and content for user to further explore for what they want to know. At the same time, the result should be clear and keeps the Simple rule as well. The constraints of the goal are that picking and transforming the right data are not always effective. If a query is incorrectly transformed or not attractive enough, the result of the visualization may become redundant or deceptive. To prevent such misuse, we need to be meticulous about what extra data we take and explain them in a careful way. The requirement of the target is to present data in multiple aspects. We plan to prevent at least three different point of view to generate our idea in the final diagrams.

The third goal, Intuitive, is to make the visualization to be user-friendly. In other words, user may know how to use the user interface and interactive function in a short time, even without watching the tutorial or description. The main constrain is that an intuitive interface is never easily constructed while having multiple functionalities. Even the most well-known applications like Twitter or Snapchat needs a bit of indication to navigate the users through the various implementations. As a result, the requirement of our target is to let users to become familiar with the visualization within a couple minutes, and substituting text indication with graphic guiding.

The fourth goal, Fun, is to attract audience to actively explore the visualization. Furthermore, the user may appreciate the work and help to promote the system to social media. The main constrain is that the visualization must be vivid, unique and attractive enough for the audience, otherwise they may get bored and leave the website within seconds. Hence, the requirement is to present an aesthetic diagram on the main page to create a positive first impression. On the other hand, the data representation needs to be dynamic or interactive in order to attract the attention of the audience.

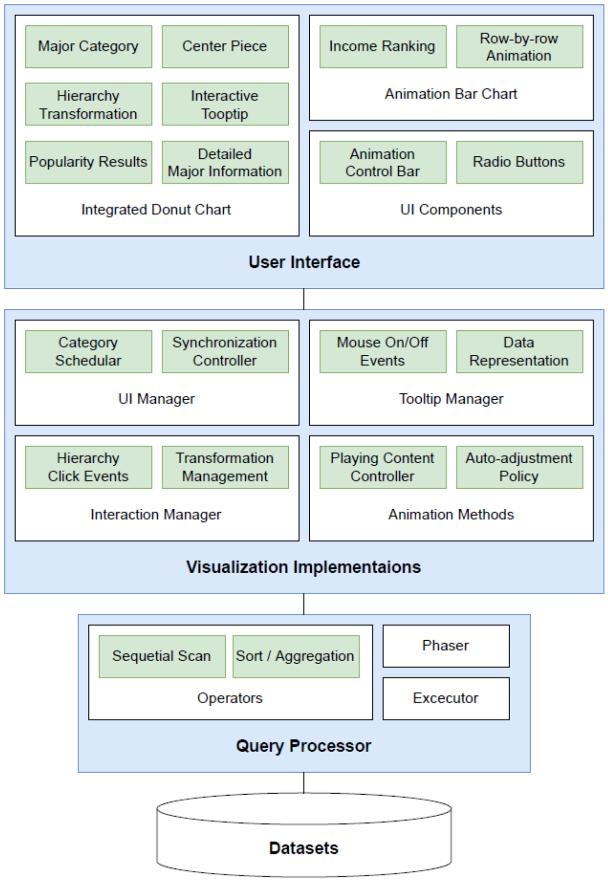

Fig. 1: Architecture

4 Architecture

There are four layers in the visualization system: data deliver layer, query processor, visualization implementations and user interface, from bottom to top. In data deliver layer, we simply import the datasets into the observable system. After raw data is imported, the query process will scan the data, and make it available in other JavaScript function. To transform the data into the target form, we applied operators such as join, aggregate, and sort. To accelerate the later categorized result, we prepared additional index for later implementation as well.

For Visualization Implementations, we basically built up four main function groups to control our data transformation and delivery. Firstly, the UI Manager is a combination of HTML and JavaScript attributes which generate simple UI components and categorized the current representation. Since there are two figures in the result, the synchronization control is necessary to align the two visualizations plot. Secondly, the Tooltip Manager controls the tooltip events to support the interactive features and data division. The tooltip including mouse-on event, mouse-off event, and miscellaneous data transformation and delivery policies. As a result, tool tip manager is one of the most important parts to support overall interaction user experience. The third part is Interaction Manager, which maintain the interactive features of the system. The most explicit feature of our interaction implementation is the hierarchy structure of the donut chart. While the diagram can represent either parent layer of children layer, the event controller needs to strictly handle the color, shape and text transformation among each event. The last one is the Animation Manager, which control the playing content and communicate with UI manager to make the scheduler work. On the other hand, the shape, color and data transformation will also be applied during the animate being played. Overall, four groups of managing functions transforms our data into a visualized for into the user interface by mostly D3.js in JavaScript and HTML.

In the top layer, user interface, there are mainly three parts that represent our final result of the visualization. The first part is the Integrated Donut Chart, which represent the complete list of majors within categories. The inner ring demonstrates the main categories, and the outer circle shows the specific majors. With the tooltip features, the figure is interactive while using, and the center piece will be displaying detailed information which correlated to what your mouse is point at. The click event will make the hierarchical data transform in the certain way, which further strengthen the interaction experience. The second part is the Animation Bar Chart, which mainly consist of the income ranking of the certain major category. The diagram is auto-playable with row-to-row animation which can attract user attention. The color gradient and text information further strengthen the distinction of each major rank. The last part is the User Interface Components, which includes the animation control bar for the previous two diagrams, and radio buttons which allow user to select the data content they are interesting in. All features are represented on the Observable platform, and are supported by JavaScript, HTML and markdown.

5 Methods

This section contains the result of the visualization diagrams, implementation methods, designing decision and some alternatives which are not introduce in our final results. The two main visualizations are the Integrated Donut Chart and Animation Bar Diagram.

The work is inspired by the Animated Transitions in Statistical Data Graphics [9] and Graph Visualization and Navigation in Information Visualization [10]. The content of the visualizations are variants of Zoomable Sunburst [5], Bars Animation [3] and Horizontal Bar Chart [1].

5.1 Design decision

After the early survey was conducted, there are several visualizing approaches we can implement. To optimize the effectiveness and reach the design goals, we concluded some design decisions as the following.

5.1.1 Data Categorization

As we first observe the data, we found that the whole dataset would be better to be expanded in a vertical order. That was to say, there was a main category that would later be divided into several sub-categories. Specifically, the main category in this project referred as the faculty; the sub-category referred as the departments, or the majors, belonged to the faculty.

5.1.2 User Interface

We could not expect the user to read the whole instructions before interacting with our plot. There was no denying that a graph could captured more attention in most of the cases. Therefore, it was necessary to include some instructions in the graph, such as the button that told the user to click to return, or a centerpiece used to display a detail. Finally, the label accompanied with the unit could make give the user more senses about what our graph were going to deliver.

5.1.3 Auto-played Animation

For those who don't know which faculty to look the first, we designed a function to display the data and switch between different faculties automatically. Besides, by adjusting the color and the transparency of the rectangle or sector, we could make a specific one more obvious, which could force the user to focus on that data.

5.2 Integrated Donut Chart

The Integrated Donut Chart is the main visualization which present the hierarchical information of major with the corresponded income and major popularity. The diagram is aligned with the UI components and Animation Bar Chart.

Fig. 2: Integrated Donut Chart

5.2.1 Data Transformation

According to our design goals, the plot should be displayed in at least two formats. Before being chosen by the user, the donut chart comprised of two donut charts, with the inner one representing the main category, or we could describe as the faculty of the major, and the outer one referring to the major.

As a result, before making the data into visualization, we had to transform the data into a tree structure, in which the parent node was defined as the faculty, while the children node defined as the major. Module d3-hierarchy was used in this project to help transform the data as well as create the additional information of each node in the structure, such as the index of each major when being arranged in a sequence. Then this index could be used to create the coordinate of the sector comprising the donut chart.

In details, the raw data was grouped by their faculty to create the name of the parent node. In this first layer of the structure, there was no numerical value being stored. The place to store those values were at the second layer of the structure, which was composed of the name of the major in the format of a string accompanied with several numerical data. For this project, we included the population, the median and quartile of the salary, which would later be used to calculate the portion of its sector within the donut chart.

5.2.2 Visualization Control

With the coordinate produced by the tree structure, the boundary of each sector could be defined but still need to be modified. The area of the sector corresponded to the population or the estimated gross product of department, which was defined as the population multiplied by the median of the salary. Compared with the total value of either population or estimated gross product of department, the portion could be calculated, and which would next be used as a factor to multiply the index. In other words, adjusting the coordinates of the corners of the sector according to the area.

Additionally, the labels being displayed on the sector was another thing to consider since the length of each major varied. If the whole name was displayed even if it will exceed the area of the sector, we believe, would be inappropriate. To solve this issue, a function was defined to cut the name of the major into certain length that will fit in the sector, while still long enough to present the name of the major. Take the name “mechanical engineering” for example, the length was too long to display. Alternatively, we could only display “mechanical eng. . . ” to indicate the major being depicted without misleading.

To make the visualization better, another cause to be put into consideration was the size of the portion. That is, we could define a threshold related to the central angle. With the sector which central angle is less than the threshold, the label would be hidden to avoid to put all the labels in a mess.

5.2.3 Attributes Highlighting

Each sector referred to data relating to different majors. For this reason, we try to give each sector with diverse attributes to help users clarify the difference between each sector. The first attribute was the color filled in the sector, in which each major was assigned to a distinct color that distributed evenly from a rainbow color map. The second attribute was the opacity. We believe that the more opacity, the more obvious of a picture. There were three different values of opacity defined, for the largest one referring to the sector which major was chosen to display in the animation of the horizontal bar chart, the middle one referring to the parent node, and the least one referring to the children node.

5.2.4 Animation

Before the user chose any sector, the donut chart would display all the faculties and majors to present a whole picture of dataset. When a certain one faculty was chosen and clicked by the cursor, the donut chart would turn into the one with only the data with majors belonging to the chosen faculty. When the user would like to return to the home page, the animation could be triggered by clicking on the center of the circle in the donut chart.

To complete this animation, the first step was to calculate the new coordinates of the sector when any sector was clicked. We define a function that would be triggered by a click event, in which the transition including the transformation of coordinates was defined. Moreover, with an attempt to make this transition more like an animation, the transition could not take place immediately. In other words, a function would be designed to interpolate the coordinates between the initial point and the final point. The number of interpolation points relied on the time that we expected the animation to last.

Another thing to consider was the transferring of the attributes. For example, if a sector was filled with color steel blue, we would expect the color to keep the same. This fulfillment relied on the function to get the current attributes, so it can pass to the next sector while corresponding to the same major.

5.2.5 Center Piece for Detailed Information

A centerpiece was placed in the center of the donut chart to display the whole set of data, including the population, the median and quantile of salary. The text displayed in this area would change according to the sector with a cursor moving over, for which a function was designed to be triggered by a mouse-move event. This function served as the bridge between the sector pointed by the cursor and the centerpiece. To put it another way, this function should get the attribute and transfer it to the centerpiece. For this reason, each position to display the information was assigned a unique id so it could be easily appointed to change the text.

To emphasize the sector being chosen, another attribute to change while being trigger by the cursor was the width of the boundary. Originally, the boundary of all sectors in the donut chart remained the same. When a sector being chosen, the attribute of stroke-width, the width of the boundary, would be selected and enlarged.

Afterwards, especially when the cursor moved out from the sector, everything would expect to recover to the status in the former step, in which case the centerpiece should filled with the gray color, so another function was designed and would be triggered by a mouse-out event. Again, this function changed the text displayed on the centerpiece by selecting the id of the place for displaying the information when no sector was pointed by the cursor.

In addition to the text, another feature that helped visualizing the major being chosen was the synchronized color between the centerpiece and the sector. Consequently, the color of centerpiece would change once the cursor was moving over the sector.

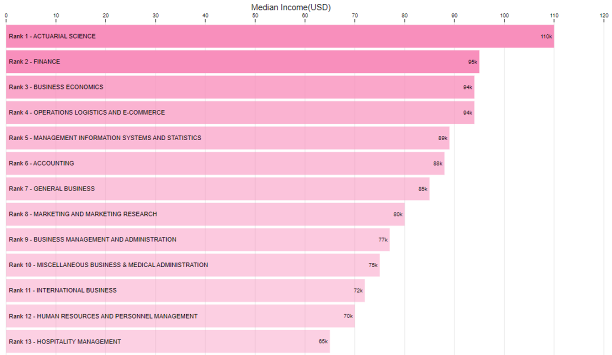

Fig. 3: Animation Bar Diagram

5.3 Animation Bar Diagram

Animation Bar Chart shows the income ranking of the selected major category. The diagram provides additional information to support the Integrated Donut Chart. This part is aligned with the UI components as well.

5.3.1 Data Transformation

As the same with the dataset used to draw the donut chart, the hierarchy structure was necessary to being created. The only difference was that we would select, group, and sort the data before the transformation into tree structure. The reason was that we intended to emphasize more details about a top-ranking major within a specific faculty that had the more salary, and the faculty that we would like to focus was chosen by the user, as there was a scrubber bar on the top of the donut graph.

5.3.2 Visualization Control

With the coordinate produced by the tree structure, the order of each rectangle could be defined because the data had been sorted. Different to the case in donut chart, the bar chart didn't design to present the portion of each major comparing to the whole group. Instead, the position of each rectangle told us about the ranking, and the length of the bar could be referred as the value of the salary. The shape of rectangle would guarantee there was enough space to display the label even in the full length of the name.

5.3.3 Attributes Highlighting

To guarantee the accordance with the donut chart, the bar chart shared several attributes with the donut chart. As the bar chart was designed to present the ranking within each faculty, all the color of the rectangle would be the same and corresponding to that in the donut chart. While there was still difference between each rectangle, which was the opacity. We define a function that could calculate the value of the opacity as a function of the index, which could be served as the ranking in this case since the data had been well sorted. What we expect to deliver was that the major ranked in the top place could be presented in a most obvious way, so the opacity would be one. Vice versa, the major with lower ranking would be assigned the opacity that was close to zero.

5.3.4 Animation

In the horizontal bar chart, we used the animation to force the user focus on the rectangle with higher ranking. In other words, all the rectangles would not appear at the same time. Instead, they appeared in a certain sequence in the order of their size. There were many ways to define the duration. The general way to make the rectangle appear in a sequence is by defining the time to do transition as a function of the index. For instance, if a rectangle were to be designed to appear in a constant interval of time, the function could be defined as time equals to 200 multiplied by index, in which 200 referred as two hundred milliseconds. By changing the function of time, for example, in a square with respect to the index, the rectangle with lower ranking would appear slower than the previous one.

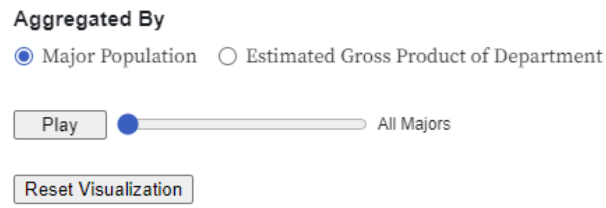

Fig. 4: UI Components

5.4 UI Components

UI Components are the interactive items which listed on the top of the visualization. The UI Components control the representing data type, current category and animation content.

5.4.1 Animation Control Bar

This function was designed for those who would like to be gain some recommendations from both donut chart and bar chart. In this point of view, the plot would be created with respect to each faculty automatically and refreshed within a constant time interval. To fulfill this goal, we put a scrubber bar on the top of the donut chart, which contains all the faculty names and would be used to serve as the output of the scrubber bar. As we clicked on the button that said "play", the scrubber bar would switch to different faculty automatically. Simultaneously, both the attributes in donut chart and the horizontal bar chart that was defined as function of this value would be switched and refreshed. To make this animation more realistic, the duration of each transition mentioned in the previous paragraph would help. Nonetheless, the user could still have the power to stop the function of automatic displaying by clicking the pause button. This would keep the value come out from the scrubber bar the same.

5.4.2 Auto-Refreshment of Integrated Donut Chart

The visualization of the donut chart was more likely to be determined by the user. The only thing connected to the automatic displaying was that a sector would be much more explicit to be seen. The way to do this was by setting the opacity of this sector to be nearly one. As the scrubber bar switched the value automatically, the most obvious sector would change to indicate the faculty that was displayed in the bar chart then. One thing to be noticed was that if the user were to stay at the same page to look more about details, they should push the button to stop the automatic displaying.

5.4.3 Auto-Refreshment of Animation Bar Chart

The more and not least, all the rectangle comprising the horizontal bar chart belonged to the faculty chosen through the scrubber bar. As the value of that was changing, the whole bar chart would be calculated and refreshed.

5.5 Alternatives

During the developing process, we found that there are several approaches that are available but not effective. To utilize the value of the experiments, we listed some alternatives that is not introduced in the final vision, and why we did not apply these approaches.

5.5.1 Zoomable Bar Chart

Compare with the zoomable donut chart, we found that the bar chart was not good to present the portion while compared to the whole group even if it could deliver the relationship between parent and children node. Besides, as the number of majors in our dataset is large, it would make the plot too wide with an attempt to appropriately display the label.

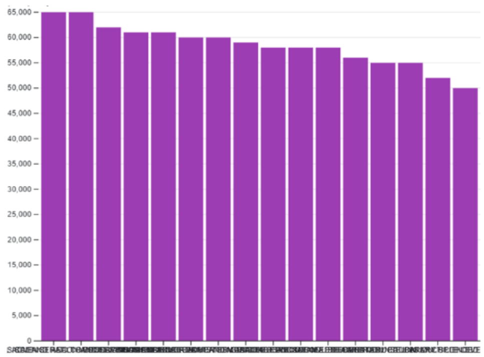

Fig. 5: Vertical Bar Chart

5.5.2 Vertical Bar Chart

What we finally applied in our project was the horizontal bar chart. The most prominent factor was the visualization of the label. As out data suggested, the label usually referred to the name of the major. They could be too long to be displayed in the donut chart. As for the vertical bar chart, the label was placed under the x-axis, accompanied by the large number of different majors, all the label would be in a mess and make the neighbor one invisible.

5.5.3 Radio Button and Drop-down Menu

Initially, we used several radio buttons to present the faculty for the user to choose. While as we put all the radio buttons in a sequence, again like to place all the label in a sequence, it was not a good way to explicitly display all the name of the faculty. Alternatively, we turned to look for an alternative solution by using a drop-down menu. This did solve the problem since the list of faculties was displayed in a vertical direction. While another problem came to appear was that we could not observe the switching process. Finally, we found the scrubber bar could best fit our needs.

6 Evaluation

In this section, we have interviewed potential users to assist us on evaluating the effectiveness and performance of our visualization result. We gathered the comments from students from three different departments of different ages. For the sake of enhancing the validity of the feedback, we analyzed the comments in different aspects and see if they reach our design goal. If not, it is still a good opportunity for us to further improve or develop our system and refine our visualization. There are three aspects we would like to discuss: problem solving, visual encoding and other recommendation.

6.1 Problem Solving

For the problem-solving part, we would like to know "Is our topic, pick up an idea major, really an issue that our target users care about?". On the other hand, do we really solve the problem to support users to explore what they would like to know.

Conclusively, our subjects do appreciate the comparison we made about the income ranking. However, the major popularity and gross product of the department doesn't seem to be directly related to the major selection. Instead, they may want to know about some more detailed description of each major. For instance, what is the major about? What is the potential job we can get from this major? Is the major correlated of my interest? How much chance and difficulty would we meet upon entering a certain major? Ideally, the subjects would like to know the relationship between the major and audiences themselves. As a result, we may generate more layers to describe the introduction of the major, as well as their correlated industry, and also the overall potential. In addition, the trend of the major growing speed may also attract users. If we can highlight some of the majors that thrived in recent years, it will be attractive for user to pay more attention on such departments.

6.2 Visual Encoding

In this part, we would like to know whether our audience likes the features of the visualization, and whether it is a high quality of work or not. In general, our subjects do appreciate our presenting type and extra implementation such as interactive features and animation. The first impression of the diagrams is good, and the hierarchical click events and tooltip are smooth and fairly comprehensible.

There are some points could be modified to make the visualization better. For example, some transformation functions of the two diagrams are not fully aligned. Some text contents are trimmed or hidden. The number of items show on the donut chart is too much and will lead to distraction.

Through these comments, we can adjust the relevant feature in our work to make the presentation better. For instance, setting a limitation of maximum number of items on one layer, and a minimum area of each item to make sure the content is completed. In addition, we can create a floating UI interface to make the operation of animate control smoother.

6.3 Other Insights

Besides the two aspects mentioned above, our subjects provided their personal insights to help us further improve the visualization system. Firstly, we can create a personality search tree, to make user more engaged with our product. Like an AI decision tree, we can setup a tree on the backstage and generate a QA interface for audience to interact with. If the user selects a certain personality or area of interest, we can use a group of gradient color to provide a recommendation list of major for the user.

Secondly, we can build a major network which links to other related majors, corresponded industry or job opportunity, and even the job searching system. By doing so, users can make further plans for their career path, and may think deeper about how to set their academic goal. Overall, besides income comparisons, people care about what the major is doing, does the major match their interests, and the correlated job opportunities. If we can introduce the attributes above, the visualization can be supportive in more aspects.

7 Discussion

There are three main topics we will like to discuss in this chapter, which are takeaways, project limitations and future work. The takeaways section contains some brief descriptions on how we utilized our knowledge and what we gained from the lecture as well as the final project. In the limitation part, we will be discussing the major obstacles that we met and how we might proceed into developing the project if the bottleneck is removed. Lastly, we listed the future work of what we are interested in if we have more time to work on the project.

7.1 Takeaway

This is the last project for this course, which is also the assignment on which we spent the most time. One major takeaway from this project is the utilization of D3. For all of our group members, this was the first time that we were taught and used D3, for a front-end visualization tool, it is not as simple as excel or tableau but also not as sophisticated as OpenGL. There is a plethora of template such as hierarchical edge bundling, density contours, and many other well-designed templates to choose from, which is another plus for D3. We learned some basic commands that fiddles with the input data from the TA sessions as well as some sources on the internet.

From the feedback from the sample interactions, we asked from our friends, we also learned to make modifications based on that to make further refinements to our final product. One example being that we got feedback that the center display of our donut chart still shows the value that the mouse was previously on, even when the cursor has already been moved away from the subject. We then did some research of the functions 'mouseover', 'mouse move', and 'mouse-leave'. And then added a 'mouse-leave' function on our SVG. The resulting graph then had a new functionality that clears up the center display when the cursor was moved away from the subject, which added more reactiveness and improved on the user experience eventually.

One last takeaway we got from this final project is the concept of communication and teamwork. While all of our group members are from the same major, it doesn't necessarily mean that we were always having the same ideas, there were quite a lot of instances that we weren't sharing the same concepts and visualizations about the final projects. The importance of communication ended up being the key factor throughout the making of the project. We were holding online meetings almost every day or every other day during the time where we were formulating the visualization graph, especially the decisions of which features and dimensions we should emphasize on, due to the limitations of the size of the donut graph. One more crucial factor is teamwork, this is a rather all-rounded project that is consist of a visualization interactive interface, a reasonable backstory and motivation, as well as users feedback and the nitty-grit-ties within the d3 encoding, we had to come up with a logical schedule in the early stage that efficiently dispatches all the workload amongst the three members, I am personally grateful for all my teammates that kept up with the schedules without any procrastination that altogether made this final product possible.

Conclusively, we learned the utilization of d3.js, and the modifications that could be done with some early-stage product feedbacks from general users to contribute to the mid stage product refinement. We also had a solid training on teamwork and the importance of communication among our teammates, which could be valuable even after we graduated from academia and stepped into the industry. Having the knowledge of the importance of those skills could turn out being crucial when we are working with a team in the future.

7.2 Limitation

The first limitation is that the maximum dimension of a diagram is limited to some extent, while aligning multiple diagrams is also challenging. That is, if we want to design a visualization which is both simple and intuitive, we need to be meticulous of the appropriate distribution the hierarchical data while presenting the whole picture at the same time.

The second limitation is that we found that some of the library of the d3.js is not flexible enough. In other word, if we try to integral multiple features or create our own function, chances are that the result are not behaving like what we had expected to be. In this course, we have learned some technique and examples on JavaScript, but little about HTML and CSS. If we can learn more about the two, we can probably generate our own animation and interactive features on our own, and to create more unique ideas.

The third limitation is that the datasets which are published on the Internet is actually very limited. Take our datasets, graduate students' income, as an example, all we found was a dataset published around ten years ago. As a result, we cannot extend the number of aspects of our project since there is no additional data for it. In this situation, we concluded that generating a good dataset might actually be harder than creating a great visualization. This could be one of the inspirations to us on how to gather useful data, and how to visualize them in the future.

7.3 Future Work

There are three directions of how to improve and extend our project in the future. Firstly, we would like to extend the dimension of our data. Currently, we focus on the income and popularity of majors. Through the feedback from our target users, we think that the description of the major, the correlated interests, the matched personalities, and the related work field are also important. Therefore, if we could get the related data of the majors, we can make a lot of extension function to make the visualization run is various aspects. We can also generate more layers or searching system to enhance the interaction user experience. Secondly, we would like to implement an AI decision tree which provide a series of question to let the user to interact with. For example, our system may ask the user: what is your interest? Which do you prefer, paper work or practical projects? After finished such interview, we can generate a list of recommendation as feedback for user. It may make the result more attractive and persuasive for our target audience.

The last working direction is that we want to build a network system which is linked to popular job searching websites such as Linkedin and Handshake. It is really important for student to learn to find a desire internship of formal job as early as possible. If we can combine the idea into our visualization, it will incite the user to realize what their future will be like. After some job searching, the major difference and comparisons will be more meaningful for the audience.

8 Conclusion

In this project, we learned how to visualize data into an effective way. Through the support of interactive features and animation, we successfully attracted audiences' attention and make them willing to explore our results. Even though we have encountered several obstacles during our design and developing process, the final result is being able to present our core target that helps our user to select a major in a scientific way. To further improve the effectiveness, we expect that our work can be a contribution to the society that helps developing more visualization and network for young student to plan for their future. That is, the visualization system can be systemized and informative enough to support content exploration in a plethora of aspects and eventually making positive impact.

9 Team Member Contributions

Warren is the leader who constructed the architecture of the visualization system. He designed, optimized and maintain the code of the two main diagrams. He also helped for additional experiments such as alternative approaches, interactive features and animation methods.

Thomas is the manager of the user interface, user experience, and problem solving. He surveys the external content such as papers, datasets and reference work, in order to generate new idea or improve the effectiveness of the results. He also contributed on interview and evaluation of the user experience feedback, and introduce the UI modifications.

John is the director of the visualization design. He gathering ideas and generate blueprint of the system, and help to review and improve the performance of the diagrams. He also designed the poster and construct the architecture of the final paper work.

References

[1] M. Bostock. Bar chart, horizontal. https://observablehq.com/@d3/horizontal-bar-chart, 2018.

[2] M. Bostock. Partition, sunburst. https://observablehq.com/@d3/sunburst, 2018.

[3] M. Bostock. Stacked-to-grouped bars. https://observablehq.com/@d3/stacked-to-grouped-bars, 2018.

[4] M. Bostock. Zoomable icicle. https://observablehq.com/@d3/ zoomable-icicle, 2018.

[5] M. Bostock. Zoomable sunburst. https://observablehq.com/@d3/zoomable-sunburst, 2018.

[6] M. Bostock. D3.js – data-driven documents. https://d3js.org/, 2021.

[7] M. Bostock and M. Meckfessel. Observable. https://observablehq.com/, 2016.

[8] M. Freeman. College majors. https://github.com/fivethirtyeight/data/tree/master/college-majors, 2014.

[9] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 2007.

[10] I. Herman, G. MelancËon, and M. S. Marshal. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 2000.

[11] Hmcll. Pokédex. https://observablehq.com/@hmcll/homework-2-pokedex, 2021.

[12] B. Ninja. Canal grande tier list for brawl stars. https://brawltime.ninja/tier-list/mode/bounty/map/Canal-Grande, 2018.