Final Report – Group 16

Daniel Pereira
IST (89425)
Lisbon, Portugal
daniel.r.pereira@tecnico.ulisboa.pt

Isabel Soares
IST (89466)
Lisbon, Portugal
isabel.r.soares@tecnico.ulisboa.pt

Rodrigo Sousa
IST (89535)
Lisbon, Portugal
rodrigo.b.sousa@tecnico.ulisboa.pt

ABSTRACT

UPDATED—21 December 2020. The following report summarizes the process of development of a visualization from scratch in guided and iterative process for the course of Information Visualization. The visualization must focus on a specific domain. Ours focuses on "The Evolution of Mobile Phones: Brands and Specs". This was the first step of the development: the domain was established and hypothetical questions that the visualization must answer were created. Then the process was distinguished in the following steps: Data Processing, development of a Visualization Sketch, development of a First Prototype and finally development of the final product, the visualization itself. Each of these phases were revisited further along in the project with the intent of improving by combining what we had defined with the new knowledge that we were acquiring, either of the domain itself or of the implementation details. The final product must be capable of allowing a user to explore the information about the domain, in an easy, intuitive and profound way. Our final product accomplished these goals.

Author Keywords

Visualization; Information Visualization; Phone Models; Phone Brands; Specifications; Hardware; Technology, D3

ACM Classification Keywords

•Human-centered computing~Visualization~Visualization application domains~Information visualization

INTRODUCTION

In this project we are looking into **"The Evolution of Mobile Phones: Brands and Specs"**. With this visualization, we hope to show how the brands and models developed over time both economically and in terms of the technology and its hardware.

We think this is an interesting subject because we are a technological generation and through this project, we will be able to expose the evolution of a device so crucial to our lives. Furthermore, this is a subject that is highly unexplored as of now. There is no shortage of specifications and hardware components in current technology, what is in fact missing is a tool with the potential to analyze its growth.

At the beginning, the questions that we proposed were:

- What are the brands that manufacture models that prioritize battery life over other specs?
- What cell phone brands had a peak in sales? When?
- How many models did each brand develop in a given time period?
- Is there a correlation between the number of models of a brand and that brand's revenue?
- Is there a cyclic period of releases of phone models? Do the peaks occur every year? Every six months?
- When did a certain specification / hardware component start to be implemented on phones? What was its prevalence in phone models across the years?
- Is there a relationship between the sudden usage of a new component (like Bluetooth, DUAL SIM, etc. …) by a brand and the change in revenue of that brand?

We still felt that there was an obvious opportunity that would allow us to further this domain, so we ended up adding another question.

- "How did the battery life of a certain brand evolve over time?"

RELATED TO WORK

After some research on web about this theme, we discovered that there are not many visualizations related to it. Most of the visualizations were very simple like a bar chart that includes a few brands where we can see the most popular brands along the years. [1]

At the beginning, before deciding on the evolution of cell phone brands, we had the idea to focus on a more concrete subject: "Mobile phone activity in a city" namely hourly phone calls, SMS, and Internet communication of an entire city. [2] However, we decided that a broader domain about cell phones would be better for our visualization.

Due to the small number of visualizations available and the simplicity of the ones available, we opted to innovate, in other words, we decided to start from the bottom… making a visualization that contains a little bit of what we saw during our research and our own knowledge of the domain to guide us on the important factors that needed to be available to be explored through our visualization.

THE DATA

The data that most of the visualization focuses on was acquired from *Back4App* [3], a company which offers the possibility to create with it a backend but that also has a database with a good offer of free datasets to be used, as it was the case with ours. This dataset contained over 8 thousand cell phone models from over 100 brands, each model having more than 30 variables which defined it. Although having a lot of information would be good for our visualization, this was too much, this also meant that cleaning it up would take some work.

In the other hand, from the economical side of the visualization, we started out by using some data that was published on *Wikipedia* [4] which showed the profit of various brands on cell phones throughout the years, but this one was small specially when compared to the one mentioned before. We hoped and tried to find more or better data that we could correlate with the rest further along in the project, but we were, in general, unsuccessful. Most of what we found were already developed visualizations or videos which had no data publicly available or if it was available had the revenue in total (not only made by selling phones), had the number of phones sold and not the monetary value associated, or had the market share of the various brands in cell phone sales and not the concrete values [5].

Most of the work and cleanup of the data was done through Pentaho. The **first dataset** was processed mainly by parsing string attributes and in a way standardizing the information available, for this we needed to implement several strategies and combinations of them, like the use of *Filters* and *Regex Commands* to capture the sensors of each phone and separate them into their own attributes. Some derived measures were also added, like for example the *Aspect Ratio* parsing two attributes from *Display Resolution* and calculating their ratio, *ram_MB* and *im_MB* which were not using the same units across models, and *Year*, *Quarter* and *Month* which had to be parsed from *Announced Date* (and in some cases computed, for example when we had the month, we also wanted to define the quarter). **Missing values** and **outliers** were also treated accordingly employing the strategy of a *sentinel value* and *removal*, respectively.

The **second dataset** was easier to deal with, due both to its simplicity and size, we simply converted the table into a dataset with *brand* and *year* as keys and *sales* (in Millions) and *number of models* as attributes.

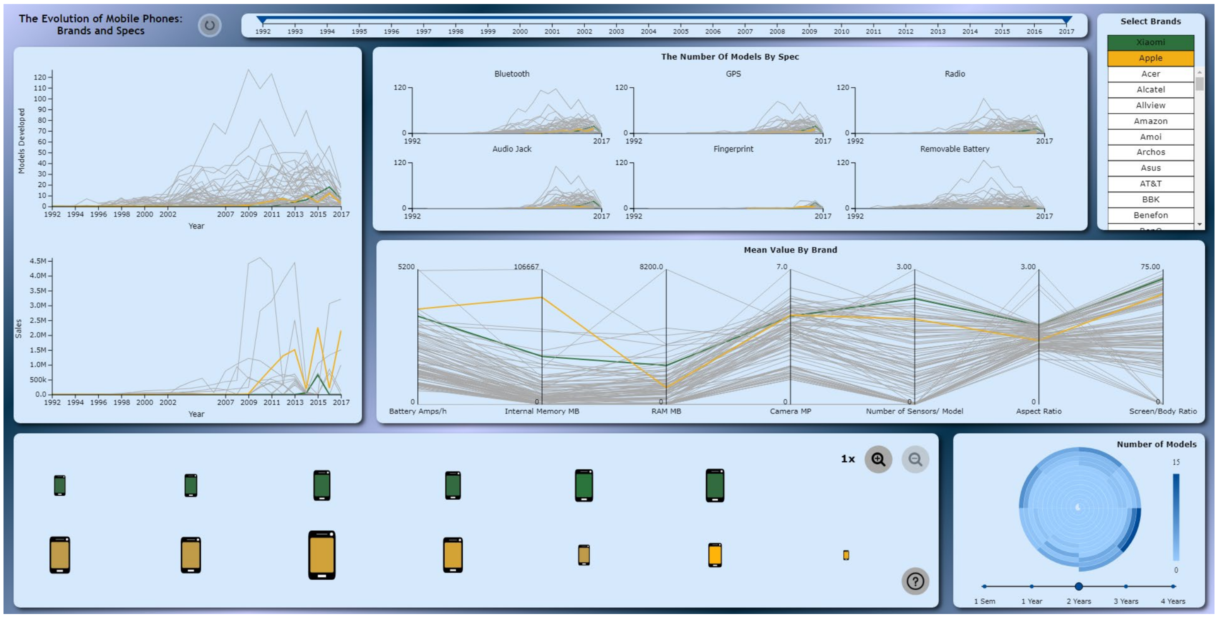


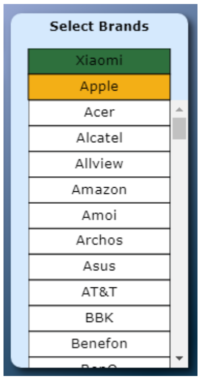
Figure 1. Visualization Overview

VISUALIZATION

Overall Description

Our solution for the visualization was to contain various idioms, all interacting with each other, to expose information about brands that would not be obvious otherwise. Each idiom is contained in a rectangle space and all the components fit on the computer screen.

Initially, there is a predefined **selection of brands** (see Figure 1) when the visualization starts, and the user may select or unselect brands through the brand selection box on the top right. This allows a selection of up to 0 to 4 brands, and all these selections are supported by other idioms. The brand selection box allows scrolling since the list of brands does not fit on the small space. The selected brands will stay at the top of the box and each brand box is given a color when that brand is selected.



Select Brands

- Xiaomi
- Apple
- Acer
- Alcatel
- Allview
- Amazon
- Amoi
- Archos
- Asus
- AT&T
- BBK
- Benefon

Figure 2 - Select Brands.

The user may also **select a range of years** (see Figure 2) on the time selection box, located on the top of the visualization. The smallest time range that can be selected is of 2 years, while the biggest only must not go over the minimum and maximum years (1992 and 2017, respectively). The time selection can be done by adjusting the upper and lower limit separately or by dragging the selected range.

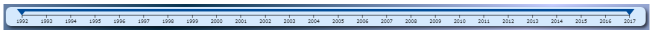


Figure 3 - Time Selection.

On the top left, below the time selection box, is the **line chart of models developed** (see Figure 3) over time. It shows lines over time representing the change of models developed over the years (where each line represents one brand). The user can select or unselect a brand by clicking the corresponding line, as an alternative of using the brand selection box. The lines have a grey color when they correspond to an unselected brand and have one of 4 available colors otherwise. Hovering over a line will show a tooltip containing the name of the brand that corresponds to that line, along with number of models developed, sales and year being hovered. The time axis changes when the time range selected changes, allowing users to only see the years they are interested in.

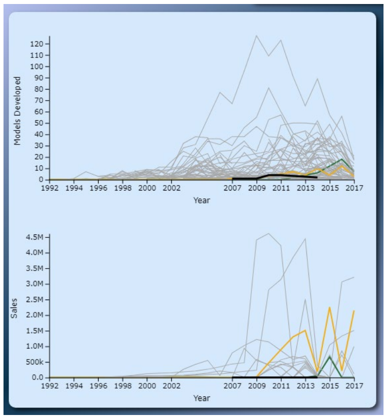


Figure 4- Line Charts.

Below the models developed line chart there is a **sales line chart** (see Figure 3) as well. This line chart is like the models developed line chart in all the aspects described above, with the exception that it instead represents the change of the sales over time. Hovering over a line on either chart will highlight that line, along with the line on the other chart that corresponds to the same brand being hovered.

Directly below the time range and between the line charts and brand selection box, we have the **small multiples line charts** (see Figure 4). The 6 small multiples represent the change of models developed over time that contain a specific component (*Bluetooth*, *GPS*, *Radio*, *Audio Jack*, *Fingerprint* and *Removable Battery*). These small multiples also behave like the line charts, except the tooltips only show the number of developed models (so they do not cover the small multiples). When the title of the small multiple is hovered, it gets highlighted and when clicked, it will show dashed lines on the models developed line chart for each of the selected brands (along with a legend on the top right of the line chart), allowing the user to visualize how many of the developed models contain that component. Up to one component may be selected.

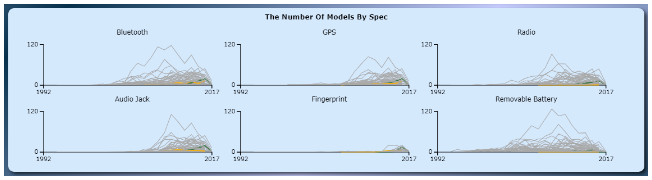


Figure 5 - Small Multiples.

Below the small multiples and the brand selection box is the **parallel coordinates chart** (see Figure 5). This allows to visualize the mean values of each component on the selected time range. Selected lines are also colored, and the user can also select brands here and they also have a tooltip on hover, displaying the value of each component next to their axis. Each axis represents, in their default order: *Battery*, *Internal Memory*, *RAM*, *Camera*, *Number of Sensors per Model*, *Aspect Ratio* and *Screen/Body Ratio*. Each axis may be dragged horizontally, allowing to switch the axes ordering. It also allows the user to select multiple brushes (one for each axis). When a brush is added, all lines not contained in all brushes will appear more transparent (except lines for selected brands), allowing for the user to better visualize the brands that correspond to the values selected by the brushes. The upper and lower limits of each brush are displayed next to it.

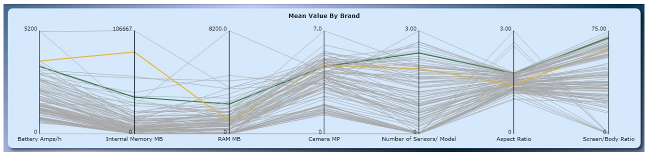


Figure 6 - Parallel Coordinates.

To the bottom right is the **spiral chart** (see Figure 6). This allows the user to see if there is any cyclic pattern in the release of developed models. The user may select the period of one revolution on the bottom of the chart. Each arc of the spiral chart represents either one month, quarter or semester, depending on the revolution period selected below. Hovering over one arc will show a tooltip describing what period of time that arc corresponds to, along with the number of developed models on that period. It also shows a rectangle region on the line chart corresponding to the range of time of that arc. There also is a visual scale on the right to illustrate what value each color represents.

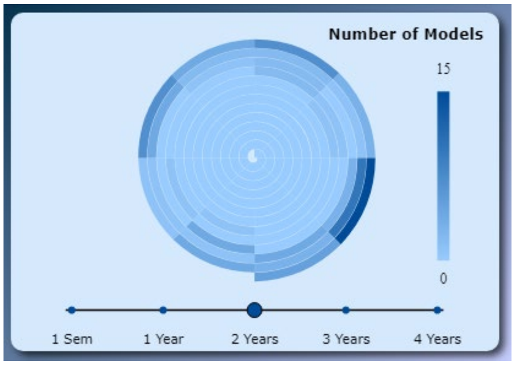


Figure 7 - Spiral Chart.

On the bottom left is a **glyph chart** (see Figure 7). This shows glyphs that encode information about some attributes of the models released for the selected brands and time period. Each glyph encodes mean values for all models released in a year by a brand. These glyphs are displayed in rows, one for each brand (up to 4 rows like the selected brands), and along each row they are displayed in chronological order. Each glyph has a *color* corresponding to the *selected brand color*, and that *color's saturation* represents the *mean internal memory* of that brand's models in that year. The *glyph's size* represents the *mean battery life*. There is also a zoom feature to better see the information of the glyphs. When hovering a glyph, the glyph increases to a bigger size (fixed for all brands selected) and displays text on its "screen", allowing the user to see the exact encoded values. A description of the encoding is also displayed by hovering the "?" circle on the bottom right of the glyph chart.

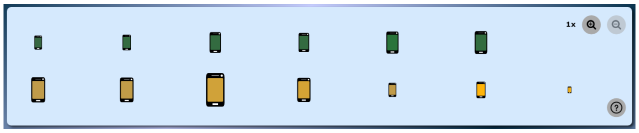


Figure 8- Glyph Chart.

On the top left, there is also a **reset button** (see Figure 8) that brings the visualization back to its initial state.

Figure 9 - Reset Button.

Rationale

First, we start to consider the best charts to answer each question according what we learned in the theorical classes. For instance, we thought the *spiral chart* was the best solution to visualize the cyclic period of releases of phone models every year. We also thought *small multiples* was the best solution to display the existence of specific components and its interaction with the line chart (translating a line from the small multiples into the line chart) is good to study the prevalence of a component. The *glyph chart* illustrates how the value of an attribute like battery life evolved over time for the selected brands. The *parallel line chart* helps to know what brands prioritize certain components over others.

As an example of the problem of scalability, one of the problems we faced was that initially the intent was to display glyphs for each phone model. The result was that with brands like Samsung, hundreds of phones were being displayed in the same line, so we decided to make the glyph represent the mean values of all phone models from a single year.

Regarding the **evolution of the prototype**, our final version has many differences from the initial sketch. The *glyphs* used initially were of cogwheels, but these were unrelated with our theme and it was harder to encode the attributes in them. Our *sales* and *models developed lines* were supposed to be displayed on the same line chart but using an axis to represent two different units was confusing. The *spiral chart* was also intended to provide a cyclic view for sales as well, but our data was not appropriate for this display (only had numbers for years and not for specific months). For *brand selection*, it was intended for users to be able to select the brand on the brand selection box, and then unselect them by clicking on the bubble with the brand logo generated from the previous selection, but this was considered confusing and did not serve an additional purpose, so it was scrapped in favor of the simpler selection box. Finally, the *time selection* was initially intended to allow selecting months, but we only have sales data for whole years, so we went with only allowing a selection of a range of years.

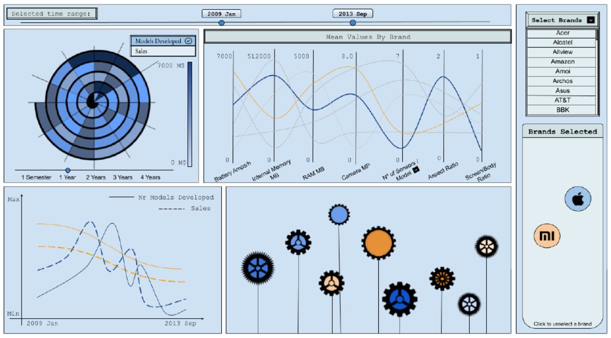


Figure 10- First sketch.

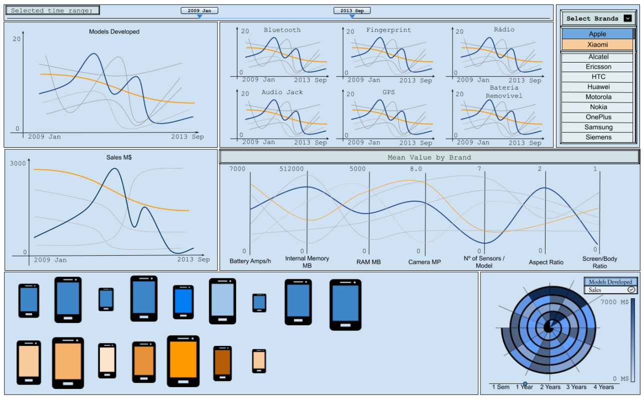


Figure 11- First sketch with improvement.

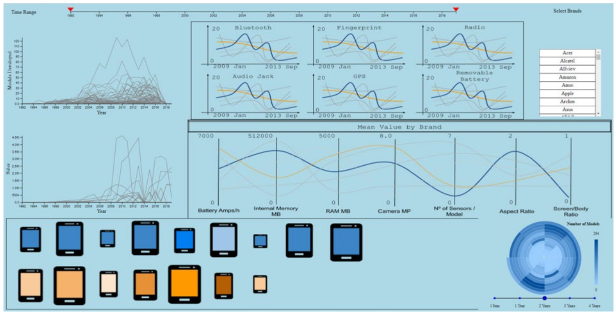


Figure 12 - Intermediate sketch.

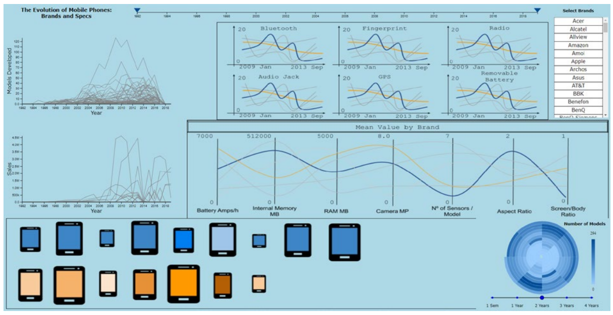


Figure 13 - Intermediate sketch with improvement.

Demonstrate the Potential

Suppose that the user starts the visualization and wants to see how the **number of smartphone models developed** by Samsung in a year correlate with **the sales in that year**. First, he would change the selection to only have Samsung selected and years starting at 2008, for example. The user would then notice that in 2012-2013, the number of sales was at a peak while the number of models developed were low. This is one unexpected result, as usually having more models available should result in bigger sales for the brand. (Answered both *"What cell phone brands had a peak in sales? When?" and "Is there a correlation between the number of models of a brand and that brand's revenue?"*)

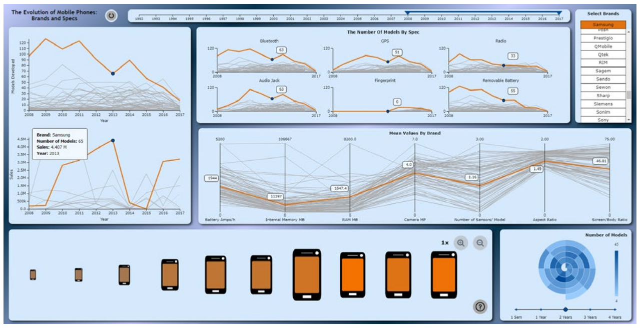


Figure 14 - After selecting Samsung and years starting at 2008.

Afterwards, the user decides to look at the **spiral chart** to check if there is a one-year cyclic pattern of release of phone models, so he switches the period length to 1 year. At this point the user decides that he cannot discern a pattern from the spiral chart, so he concludes that Samsung does not have a specific time of the year when they release more models. At this point the user decides to check if other brands are the same, so he unselects Samsung and selects Alcatel. Immediately, he can see that Alcatel has a very discernible one-year pattern around January and February. This is an interesting finding, as it implies that some brands release most of their models at specific times of the year, while others do not (Answers *"Is there a cyclic period of releases of phone models?"*)

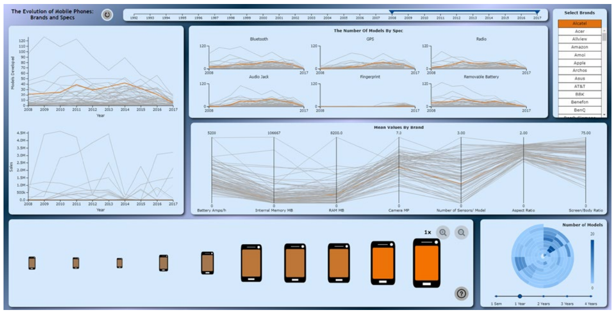


Figure 15 - After selecting Alcatel and a period of 1 year on the spiral chart.

Finally, the user gets interested in studying which brands release models that prioritize a certain spec over others, so he selects a smaller time period of 2011 to 2013, then looks at the **parallel coordinates chart** and observes that there's one brand line at the top of the Battery Amps/h axis, so he clicks it and sees that it's the brand Apple. It can then be seen that Apple is not doing as well on the Camera MP attribute, so the user concludes that Apple prioritized battery life over camera quality in this time period (Answers *"What are the brands that prioritize battery life over other specs?"*)

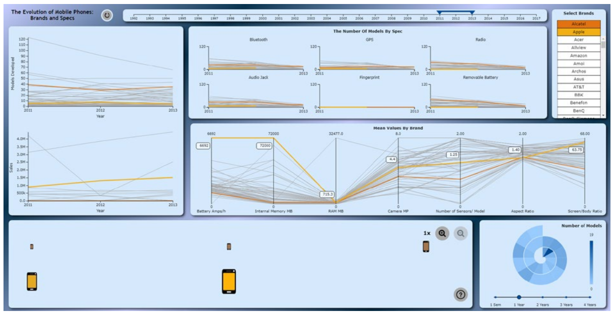


Figure 16 - After changing the time range to clicking on the line that is highest on the

IMPLEMENTATION DETAILS

The various components of our visualization use the same global variables to store the information. This means that each one of the files is only read once, stored globally, then a copy is created which will be affected by changes and filters that affect all the components (for example the time selection). It is important to note that then each component might still need to preprocess the data, this does not mean filtering or changing the values themselves but changing its structure, for example *rollups* and *groups by*.

The **main logic of each one of the components** is stored in its own file to keep some separability between them. Some patterns started to emerge during the implementation phase, for example each component has a function which builds the component, one in which it updates itself, one where it updates a selected or unselected brand, etc

The **handling of events** is dealt in general through *dispatches*, *events* that are triggered globally, by one or multiple different scenarios, that allow for a better separation between components while allowing for them to interact with each other. The only exception to this is events which only affect the elements themselves and not the visualization as whole, for example the hovering and dragging of some elements like the time selection or period selection. The *event drag end* is still treated globally through dispatches but during the dragging itself only the position of the elements is changed so there is no need to dispatch events.

When it comes to the **algorithms and implementations used**, most of the idioms were implemented from scratch using examples only as a guideline and even then, most of the cases were only with the intention of clarifying some detail on the documentation of the library, the only exception to this was a well-known function used to wrap text when bounded to an element (*Mike Bostock's wrap function* [6]) and an *implementation of a spiral chart* [7] that we used as a basis and then altered to fit only our needs, this implementation only expedited our implementation of the spiral chart since most of the complexity came not from code but from a Mathematical and Trigonometry standpoint, where we needed to calculate all the points correctly.

One of the techniques/algorithms implemented across the visualization by us that was more out of the norm was the *hovering of lines*. Most of the visualizations create a 'ghost' of the element to be hovered slightly bigger and invisible that when hovered is shown, by doing this the user does not need to hover exactly over the element to trigger the hover event. Since we have such a high density of lines, instead of creating copies of each one, which would need to be updated and more than likely still end up on top of one another, we compute the closest line / path to the mouse cursor and if it is close enough, we dispatch an event.

One of the **most challenging components** was the Glyphs Chart. We found a bit burdensome to implement zooming both through the default of D3 (scroll wheel, using touchpad and double clicking) as well as though buttons (programmatically) and having both working together, but this difficulty was mainly due to the differences between D3 version 6 and the documentation that we found that was more directed to other versions.

We also found, although less than the glyphs, the parallel coordinates chart challenging. This challenge came from the high amount of interactivity that this component embodies. This component needed to allow for the selection/deselection and hovering of brands, alike some of the other components in our visualization, but with the addition of allowing for the reordering of axis, inclusion of filters (through brushing) that needed to affect all other idioms and that should be properly maintained even if the scale of each one of the axes changed (for example if the time selection was altered). In order to understand how all of this could be implemented and confirming what interactions should be allowed on a parallel coordinates chart we followed the structure of an example we found. [8]

CONCLUSION & FUTURE WORK

By the end of the project, we came to confirm what we were informed of in the beginning of the project. JavaScript is a good language and D3 is a good and extremely powerful library, but there is a decent learning curve on D3. Not from the perspective that is hard to code what we want to appear but what is in fact harder is to do it properly, in a way that will not bring consequences to what we wish to implement later. If we had the opportunity to start over, we would refactor the first idioms that we developed, places where the code might not be at the same level as the rest.

Back to the beginning, we thought that we answered all the questions that we proposed at the beginning, but if we had more time, we would probably improve the Glyphs Charts, since as explained before, due to scalability issues we did not implement what we had set out to do and we feel like it could have been a component on the visualization with potential to impress the user, just as an example implement forces in order to space out the glyphs instead of doing it manually.

One of the things that we also wished we could have spent a bit more time on would be on flushing out a few transitions / animations and a few visual details. As well as maybe change a bit the domain where we focused on, especially when it comes to the economical side, since we did not find has much data as expected.

Nonetheless we feel like we developed a good visualization, which accomplished what we expected, with a few changes from what we thought, but that we adapted well to the problems that occurred when they occurred. We even utilized parts of D3 that we were not expecting to use, like for an example zooming and panning.

REFERENCES

1. Data Is Beautiful, 2019. Most Popular Mobile Phone Brands 1993 – 2019. Video (22- 09- 2019) Retrieved December 21, 2020 from https://www.youtube.com/watch?v=IdDEVIfbGEA&t=16
2. Marco De Nadai. 2019. Mobile phone activity in a city. Retrieved December 21, 2020 from https://www.kaggle.com/marcodena/mobile-phone-activity
3. Paul Datasets, 2020. Cell Phones Brands and Models. Retrieved December 21, 2020 from https://www.back4app.com/database/paul-datasets/cell-phone-dataset/list-with-all-cell-phone-models
4. Wikipedia, 2020. List of best-selling mobile phones. Retrieved October 12, 2020 from https://en.wikipedia.org/wiki/List_of_best-selling_mobile_phones#Annual_sales_by_manufacturer
5. Macrotends. 2020. Nokia Revenue 2006-2020 | NOK. Retrieved December 21, 2020 from https://www.macrotrends.net/stocks/charts/NOK/nokia/revenue
6. Mike Bostock. 2018. Wrapping Long Labels. Retrieved December 21, 2020 from *https://bl.ocks.org/mbostock/7555321*
7. Tom Shanley, Basti Tee. 2016. d3-spiral-heatmap. Retrieved December 21, 2020 from https://github.com/tomshanley/d3-spiral-heatmap
8. Kai. 2020. Nutrient Parallel Coordinates. Retrieved December 21, 2020 from http://bl.ocks.org/syntagmatic/3150059