Checkpoint IV: First Prototype
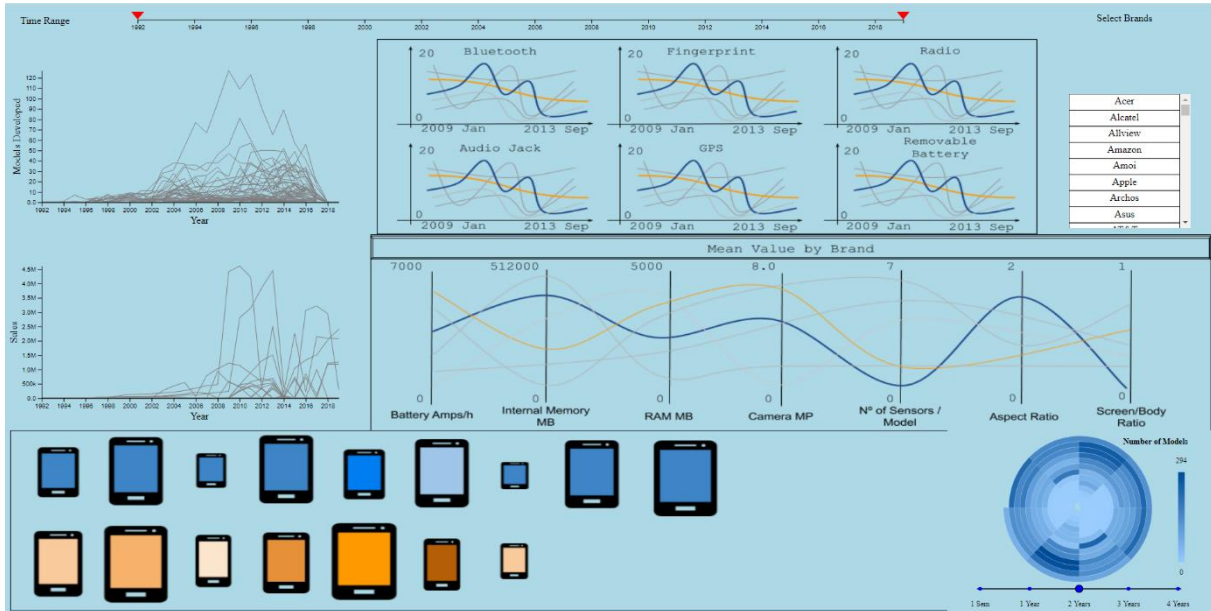
Group: G16
Date: 2020/11/28

Layout

Figure 1 – Initial layout

As proposed in the previous checkpoint our interface will be composed of several views. For this checkpoint we implemented three of them (two of them being of the same idiom). We implemented the two line charts (on the middle on the left), to represent the number of models developed and sales, both throughout time; and the spiral chart (on the bottom right), where we can explore the existence of temporal cycles in the production of phone models.

We felt the need to implement straight away the modules where the user is able to select the time period it wishes to explore (on the top left) and where the user can select and deselect brands (on the top right), since without them most of the interactivity is undermined.

Visual Encoding
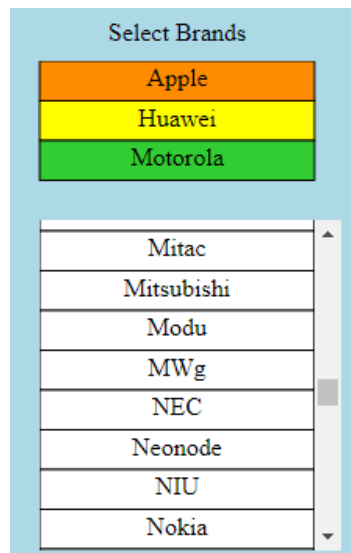

Figure 2 – Time range selected

Figure 3 - Brands selected

The **line charts** (in the figures 4a and 4b below) represent each brand's models developed and sales over time. When a brand gets selected on the brand selection box (figure 3), its corresponding line gets highlighted with a different color. It is also possible to highlight a line by hovering over or close to it and clicking it (as in the second figure, the red ball is displayed in the closest point on the closest line to the position where the mouse is being held and the orange line is the line that was clicked / selected). If a line is hovered, a tooltip is displayed next to the red ball, displaying the brand, along with the models developed and sales at that point in time. When changing the time range (figure 2), the time scales and lines on the line charts get stretched to represent a smaller range of time.
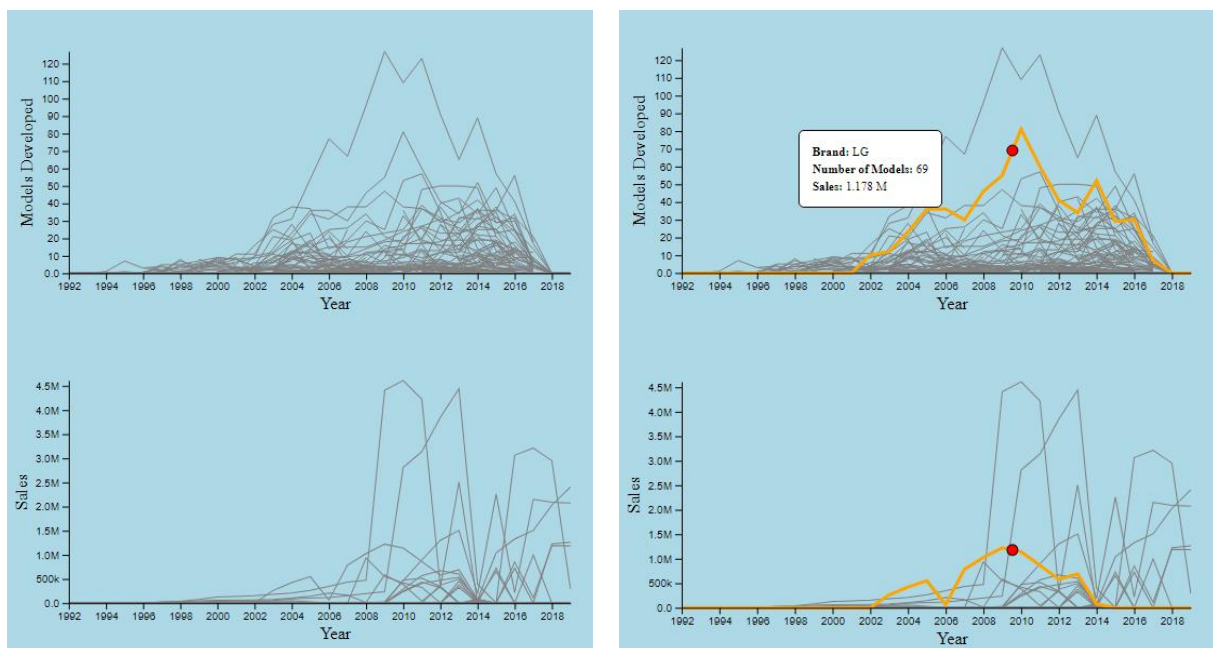

Figures 4(a) and 4(b) – Line charts (4a - left) and showing a brand selected and being hovered (4b - right)

The **spiral chart**'s (in the figures 5a and 5b below) purpose is to study the existence of periods in the release of phone models. When no brands are selected, it shows the data of all brands, while only showing data for specific brands if one or more brands are selected. There is a scale below it to define how much time each loop represents. Depending on the loop value selected, the amount of time each arc represents also changes (to represent a month, quarter, or semester). To the right there is a color scale mapping each arc's value of developed models. When hovering an arc, a tooltip is displayed with

the date and the number of models released in the time represented by that arc. When a brand or a period is selected, the spiral chart is updated according to the new selection.

One interaction between the line chart and spiral chart is that clicking on a line in the line chart will select the brand represented by that line, which in turn will also update the spiral chart.
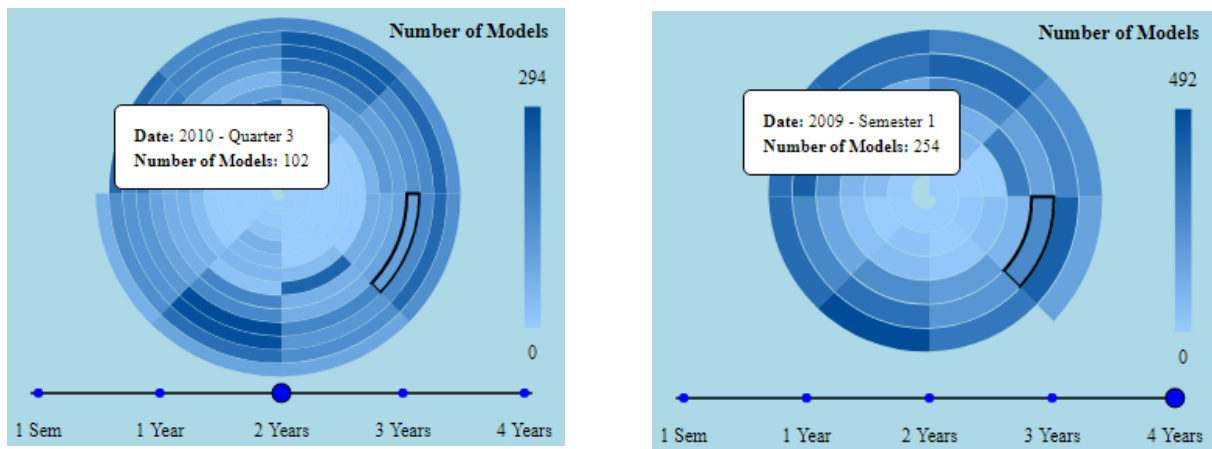
Figures 5(a) and 5(b) – Spiral chart showing a quarter with one revolution corresponding two years (5a - left) and showing a semester with two revolutions corresponding four years (5b - right)

Implementation of Linking Mechanism

Each one of our views is coded in a separate file in order to keep some organization and separability between modules, nonetheless the interactivity between them is not lost or undermined since all interactivity is achieved through **event dispatches**, treated in the main and common module (main.js). This means that in the future when we start implementing other views for events already defined, we just need to trigger them; for new ones we simply need to define them and implement the changes that they trigger (like we did for the ones on this checkpoint).

The only exception to the mentioned before is during the event of dragging, when we update pointers such as the time selection (on the top left) and the period selection (on the bottom right, in the spiral chart). The event drag end is still processed using event dispatches.