Personal Details

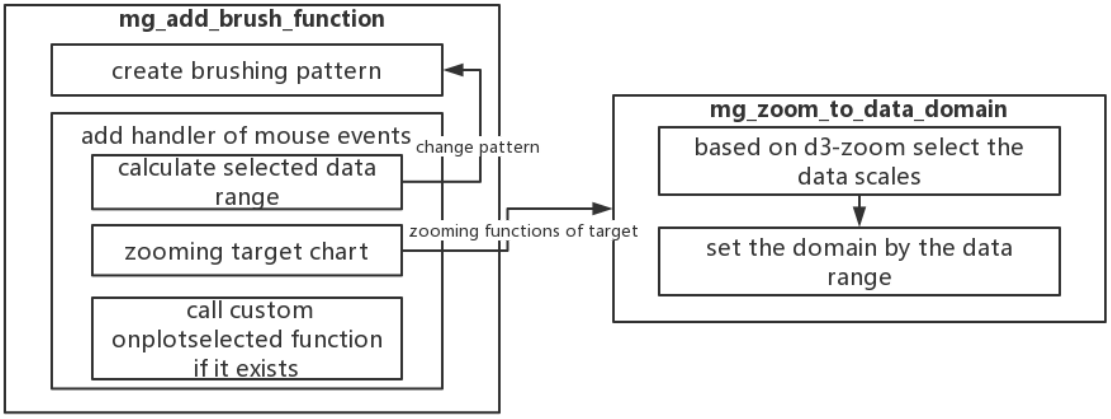
- Name: Yunhao Zhang
- Email:
- IRC nick: wangjie
- Telephone:
- Other contact methods: Telegram@wangjie000
- Country of residence: China
- Timezone: GMT+8
- Primary language: Chinese

Project Proposal

Brushing and zooming

src/js/common/zoom.js: In order to provide the zooming feature, we can design an object, it can construct based on args and provide some methods can be used to zoom the chart. In the main function of line.js and point.js, we will construct this object according to the args. This object would provide a zoom_to_data_domain() method. It receive a parameter that represents the data range. Inside use d3-zoom to zoom the chart.

src/js/common/brush.js: It provide a add_brush_function() method. This method adds brush function to the chart. It adds a few mouse-related event handlers to the chart, such as mousedown, mouseup, mousemove etc. A rectangle indicate the range of the selected data will be created in the handlers. And we calculate the range of the selected data in the handlers and emit a plotselected event include the range we calculated when the status of the mouse changes. We will then call the custom callback function here for the user to customize the performance of the event.



We can add an option to parameter of the data_graphic() to decide whether to add brush function for the chart, such as brush (a boolean that indicates the brush function's switch), brush_mode (set which axis to allow brush on), brush_allow_drag (we can allow drag the selected range), brush_allow_resize (we can allow resize the range when mouse is at the border of the range), brush_allow_deselect (we can allow deselect the range when click), onplotselected (we can allow custom behavior when selected), zoom_chart (we can scale the specified target chart based on this).

Overview plots

The user can manually create an overview plot using the options provided above. Just need to zoom the target chart in the callback function of onplotselected option.

Highlighting data points

This part is expanded by the original highlighting hovered element in different type charts. We can separate the code into a mg_highlight_data() method for control active data in the rollover method in "src/js/chart/line.js" and "src/js/chart/point.js". And allow the user to manually control which data should be highlighted.

Hover and click events

This part is based on the original functionality for highlighting hover elements. Keep the original algorithm and trigger the corresponding event when the hovered element changes or it is clicked.

If the user wants to create a tooltip can use Popper.js. It can automatically locate the tooltip element based on a DOM element or an SVG element. In the event callback we will provide data point element as parameters and the user can use this as a reference to locate popper. We can change the reference in the handler of hover event.

This is an example of using Popper.js in the current version:
https://jsfiddle.net/wangjie/6cp98pcp/

Schedule of Deliverables

April 24 - May 6: (about two weeks) Implement API for highlighting data points feature and write tests to ensure that the highlighted data points are the same as expected when the API is called.

May 7 - May 20: (two weeks) Rebuild rollover methods to emit hover and click events and write tests to ensure that events could be emitted correctly when mouse status changes.

May 21 - May 27: (one week) Debug it, write wiki for these new API, update example page.

*milestone: Implement two relatively independent features first.

May 28 - June 10: (two weeks) Implement zoom.js and write tests to ensure that the chart zooms correctly.

June 11 - June 24: (two weeks) Implement brush.js for all chart types and write tests to ensure that brushing layer is created and emits correct events when mouse status changes.

June 25 - July 8: (two weeks) Implement the functionality of calling zooming API when brushing event emitted for point and line charts separately. Then tests to ensure it works fine.

July 9 - July 15: (one week) Debug, write wiki, update examples and documentation.

*milestone: Implement brushing and zooming feature basically and then we can develop overview plots feature on it.

July 16 - July 29: (two weeks) Write tests for creating overview plots.

July 30 - August 5: (one week) Debug, update examples and documentation.

August 6 - August 12: (one week) Final wrap-up.

Open Source Development Experience

I've developed some personal projects with JavaScript or other languages and open source on GitHub, and I often use other open source projects as part of my work, if I find some bugs in the process of using it I will create a issue to tell the author. If I can solve these problem I will create a pr to solve it.

Work/Internship Experience

I work for a local Internet Company to develop internal CRM and OA system in the summer of last year. I also helped some companies make websites during my spare time. During the period include some work using Highcharts, D3 or other framework for data display.

Academic Experience

I am currently pursuing a BS degree (Computer Science) at Anhui University in China.

Why Me

I am familiar with JavaScript, HTML5 with 3 years experience and I have learned Node.js, PHP, Java and use them to do some websites. I was deeply interested in web development and start to learn it when I was in high school so although I have not have graduated now, I already have some programming experience. I have done some animation effects based on canvas and svg and tried to achieve some JS library. I think it is helpful for this project.

Why Mozilla

I used to learn web technology at MDN frequently, this project is very useful for data visualization and I am really want to implement it myself. I also think Mozilla is a great and friendly organization, I would be very honored if I can contribute to it.