

# Auto-adaptive web layout of Acemap based on mobile client

With the growth of smart phone use, more and more web-based applications start to offer access from a mobile client, whether it is a browser or a native application.

Front-end Design for Acemap is based on traditional PC client. Therefore, when a smart phone with a smaller screen loads the webpage, the show of it is not too satisfied. Many practices have revealed that self-adaption layout pattern makes the page to display more flexible and more portable when target some modern devices. In this project, I corrected some dissatisfactions on the webpage to transfigure the appearance. The website will change its layout according to the size of user interface.

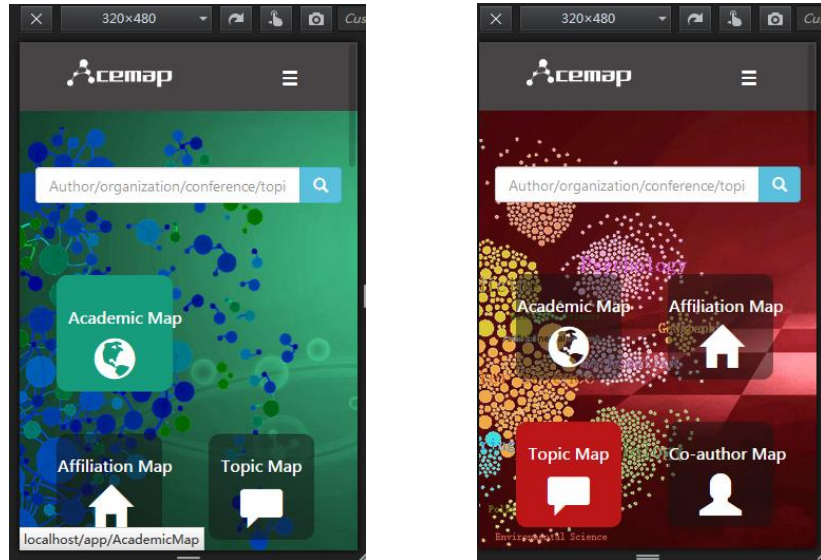
In many webpages, fluid layout is broadly adopted. Compared to traditional layout with fixed position, this layout seems more flexible and more easy to design. But this layout still has disadvantages. When windows is too small, the page will become unreadable. To solve this problem, a new auto-adaptive web layout – CSS+DIV is created and widely utilized.

Cascading Style Sheets (CSS) is a computer language to express document style such as HTML,XML. It allows webpage designer design webpage content and style separately. CSS can control any element in webpage in pixel level, and it supports almost all fonts. Division Tag(DIV) is a html tag, and it can divide documents into independent blocks. CSS+DIV has lots of advantages in web layout: shorten the code, lower the difficulty to rebuild the page, speed up the web access, and make the page more browser-compatible. Therefore, CSS+DIV layout schema has gradually become the core technology when designing html page. Currently, most CSS+DIV layout is aimed at pages with fixed width, which is more common in practice. In order to show webpage auto-adaptively in mobile client, more and more websites adopt auto-adaptive layout using CSS+DIV.

In this project, I mainly use CSS+DIV layout to make Acemap webpage more adaptive in different screen.

## 1. Home page

On home page, there are 4 buttons as entrances to other sub-pages. And when you move mouses on it, it will expand. If your screen is small and you move your mouse on a button, it expands and squeeze the others into next line. Such as:



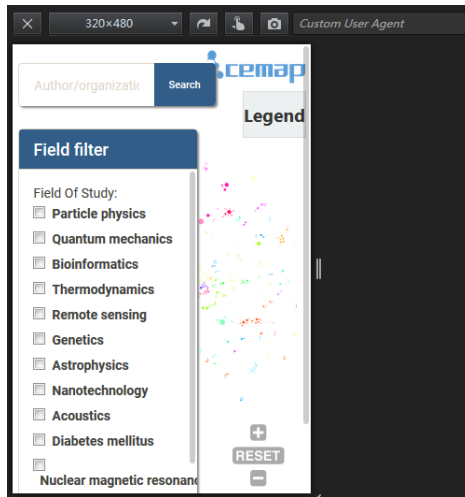
before

after

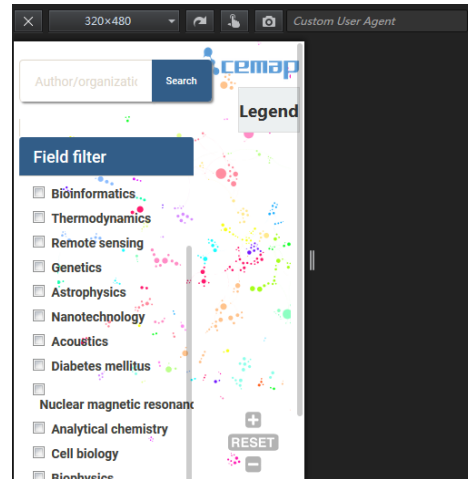
What's worse, if you move your mouse on the button on the second column, the button expands and squeezes itself into the next line. Then the mouse leaves it and it comes back. As a result, you will see the button "jumps" back and forth. To solve this problem, the expansion action should be canceled when the screen is too small. I modified *SlideMainpage.css* to judge if the width of the page is less than 450 pixels. If so, the button don't expand when selected.

## 2. author map page

- i) On the left on this page, there is a field filter bar to help readers quickly find topics they are interested in. But when the page is small, this bar occupies most of the page, and the map is almost obscured. However, the field filter bar contains lots of information, it's not easy to shrink. I chose a compromising method: to make the background of the bar transparent, which retains origin information in the bar and exposes most part of the map.



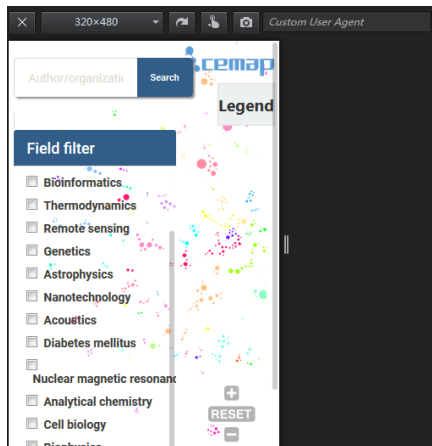
before



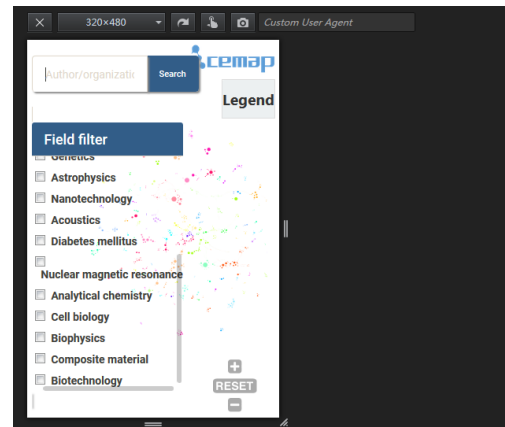
after

- ii) On PC client, field filter bar uses vertical slider to contain more text. But in mobile client, owing to the limit on page height, there is another vertical slider to show the whole page.

If you want to see the bottom of the field filter bar, you have to move both two vertical sliders down to bottom, which is not convenient. I modified mapPanel.css to judge if the height of the page is less than 600 pixels. If so, the height of the bar is set to 300 pixels.

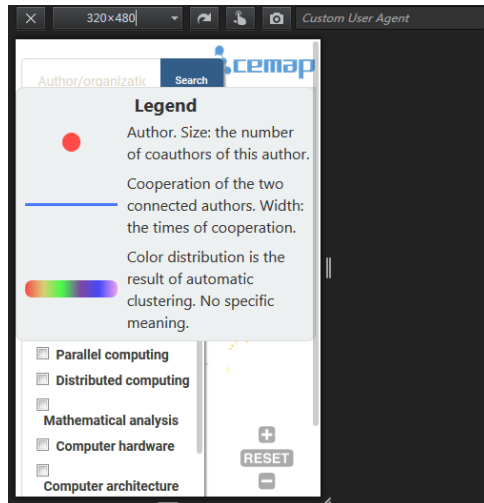


before

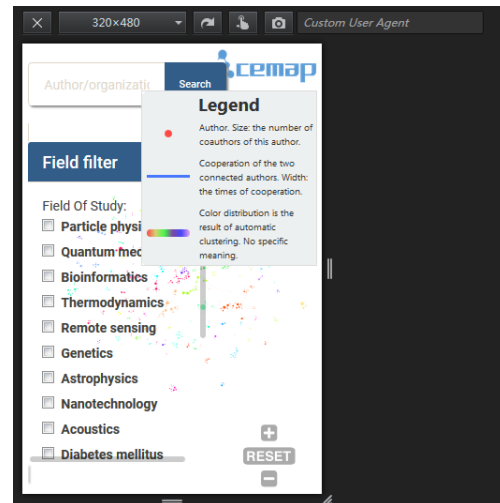


after

- iii) When you move your mouse on Legend bar, the map instruction will show up. But the size of instruction won't automatically change when the screen changes. As a result, the instruction seems to large on the webpage. I modified mapPanel.css to judge the height of the page and adjust the shape and the text's size.



before

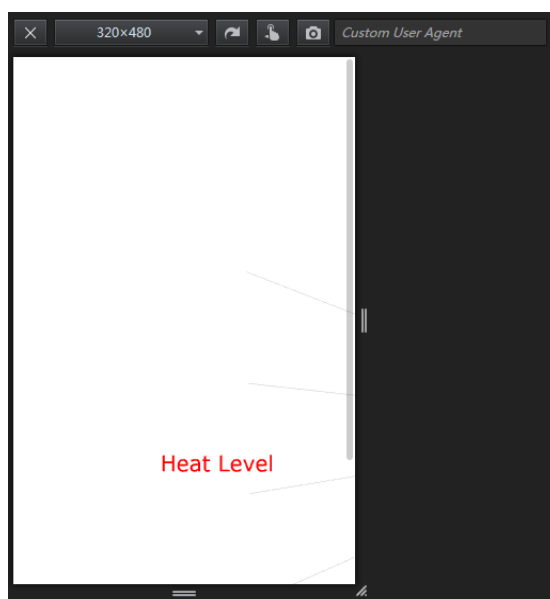


after

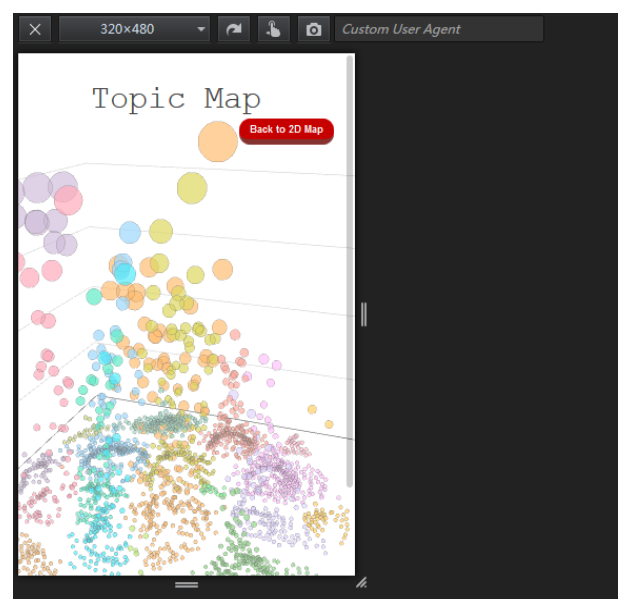
### 3. 3D topic map

When we load 3D topic map on mobile client, we find only blank on the webpage, because the definition of the map's location is absolute, and most part of the map is out of the border.

I modified topicMapStyle.css to adjust the position and the size of the map. In original version, there is a topic list on the right side. I ignored it when the width of the page is too small as it is too large. I only retain the title and *the Bank to 2D map* button. Then I modified 3DTopicMap.js to adjust the viewpoint, so the initial viewpoint will change along with the size of the webpage.



before



after

## 4. conclusion

In this project I firstly know about front-end design language such as html,css,js...

With my senior's advise , I realized that although the Acemap webpage looks great on PC client, it doesn't perform well on mobile phone. So what I need to do is to find the problems on the page and solve it. Finding problem is more important than solving it. This is what I learn most in this project.