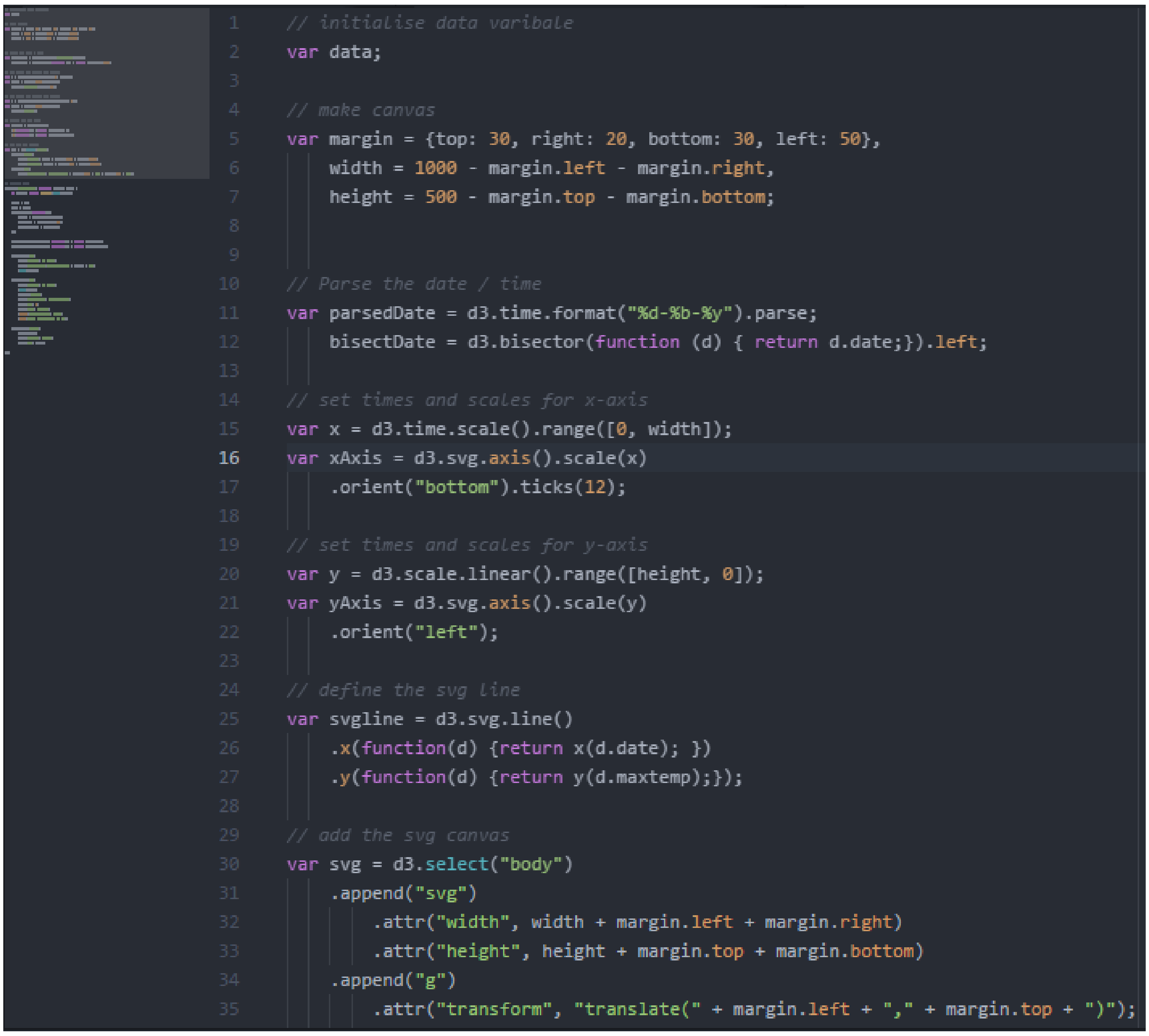
Reading 6 – Bart Quaink – 11121424

The three chosen interaction concepts are *linked navigation, filtering* and *zooming.* They will be described by examples and reasons why these concepts are necessary for a good visualization. Their taxonomy will be explained and discussed.

First of all, *linked navigation*. Linked navigation with navigation is an aspect that's widely debated if it should be included in a navigational visualization or not. The key aspect of linked visualization is to link the whole visualization to a smaller image/thumbnail. By doing so, the user easily spots his or her location on the whole map and can easily move to other different places on the map. Not only is this seen in geographical maps, but other types of mapping as well. My current preferred text editor is *Atom*, which on itself is already a great text editor, but the extensions make the programming experience with atom the best part. There is an extension/add-on that creates a mini-map of the whole code. See below.



```
// initialise data varibale
var data;

// make canvas
var margin = {top: 30, right: 20, bottom: 30, left: 50},
    width = 1000 - margin.left - margin.right,
    height = 500 - margin.top - margin.bottom;


// Parse the date / time
var parsedDate = d3.time.format("%d-%b-%y").parse;
    bisectDate = d3.bisector(function (d) { return d.date;}).left;

// set times and scales for x-axis
var x = d3.time.scale().range([0, width]);
var xAxis = d3.svg.axis().scale(x)
    .orient("bottom").ticks(12);

// set times and scales for y-axis
var y = d3.scale.linear().range([height, 0]);
var yAxis = d3.svg.axis().scale(y)
    .orient("left");

// define the svg line
var svgline = d3.svg.line()
    .x(function(d) {return x(d.date); })
    .y(function(d) {return y(d.maxtemp);});

// add the svg canvas
var svg = d3.select("body")
    .append("svg")
        .attr("width", width + margin.left + margin.right)
        .attr("height", height + margin.top + margin.bottom)
    .append("g")
        .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
```

Now, this code is pretty small, only 100 lines max. But what if you'd have a gigantic file of code that's 5000 lines in size? Then a mini map is really useful to navigate through the whole code and easily place your location relative to the code in whole.

Second, *filtering*. The taxonomy of filtering is mostly what do you want to show and how are you going to easily show the wanted data? The rest of the data should be easily accessible somewhere, but by filtering the data you'll get exactly what you're looking for. And most importantly, fast and easy. The focus is being shifted from the whole dataset to different data subsets.

In navigation, filtering is to me most noticeable while looking for routes. Navigational applications have lots of options to filter out certain data and show data that you actually want to see. What if you're driving a truck of a certain height? You'll need to filter out certain roads, if they're simply inaccessible for trucks or perhaps there are tunnels on a way that the truck can't go under. By filtering in navigation the most optimal route is easily calculated.

Other instances of navigating through data and having to filter out certain parts of the dataset are for example looking for people with a certain zip code, age, country. What if you explicitly have to search for people born in 1960, in Cambridge, who got a vaccination at dr. Gellers? The dataset probably exists for this, but you won't go look through all the data and people to find this select group of people. Filtering the data on these criteria makes it easier to navigate through the data and pick out this specific group.

And lastly, *zooming*. Zooming is kind of a broad concept, but then again not at all. The idea and concept of zooming is pretty understandable, go into more detail the more you zoom in. But there are so much different ways of zooming, specifically zooming on *what?* Zooming on photos, zooming on a map of the earth, semantic zooming. All are made for going deeper and deeper into detail, but the desired outcome is often different. In photo's you'd probably want to see a minor detail in the back, in comparison to the whole of the picture. Let's say you have a family photo, and just one little naughty nephew is pulling a weird face. You'd want to zoom in on that to see that silly face. And now let's take zooming in on a map of the earth, on google maps for example. You are zooming in on the same way, looking for detail, but now you're wondering what the fastest walk is from your work to home even though the entire walk is only 2 km. You don't really care anymore about the whole picture, the earth, but just that tiny frame of where you can see your work, and the route to your home. And what do you notice when you zoom in going from an overview of the world to your work in your small town? The details keep growing and growing, by each layer of zooming you see more and more detail of the surrounding area. You start out at a map of the earth, zoom in to the Netherlands, the provinces show up, you *navigate* to your province, find your town in this province, not only that town will show up but all the nearby towns as well, keep zooming in and you'll see the single houses and roads in your town. Markers will show up, showing historical sites, important locations and more. All of that in a couple of layers of zooming. All that detail in just 7/8 scrolls with your mouse wheel. That's the beauty of zooming in, there was no way to show all this data in the first overview of the world, but in the end you'll see so much details of your surrounding area.

Zooming can be considered a method of view manipulation and navigation. Patterns, hypotheses and specific details can grab your attention by the manipulation of the view and zoom level. Panning and zooming are all part of navigating through data.