Visualisation de données

06 - Intéraction et animation (partie 1)

7 mars 2024

Noemi Romano
noemi.romano@heig-vd.ch

Semaine passée

Semaine passée

Echelles `.scaleLinear().domain().range()`

Semaine passée

Echelles `.scaleLinear().domain().range()`

Axes `.axisBottom(), .axisTop(),.axisLeft(),.axisRight()`

d3-transition

Installation

```
npm install d3-transition
```

Transitions

Etat 1: r = 10, Etat 2: r = 50

Etat 1: width = 0, Etat 2: width = 200

Transitions

```
monSvg.append('element')
    .attr('attribute', Etat_1)
    .transition()
    .attr('attribute', Etat_2)
```

JS index.js

```js
const monSvg = select('#monSvg')
  .append('svg')
  .attr('width', width + margin.left + margin.right)
  .attr('height', height + margin.top + margin.bottom)
  .append('g')
  .attr('transform', 'translate(0 ,' + margin.top + ')');

let radius = 10;
const circle = svg3
  .append('circle')
  .attr('r', 10)
  .attr('cx', 400)
  .attr('cy', 100)
  .on('click', function () {
    radius = radius + 20;
    circle.transition().duration(2000).attr('r', radius);
  });
```

Méthodes

```
transition().duration(dureeTransition)
```

↳ d3 | Transition timing

```
transition().ease(fonctionEasing)
```

↳ d3 | Easing animations

```
transition().delay((d,i) => i*tempsMillisecondes)
```

↳ d3 | Transition timing

Exemple

↳ Observable | d3-transition

d3-zoom

Installation

```
npm install d3-zoom
```

Initialisation

```
const myZoom = d3.zoom()
```

- Initialise un objet de zoom pour activer le zoom et le panoramique

Événements

```
d3.zoom().on(typenames, listener)
```

- **typenames**: Les types d'événements à écouter, par exemple, `'zoom'`, `'start'`, `'end'`
- **listener**: La fonction de rappel (callback) appelée lorsqu'un événement spécifié se produit.

JS index.js

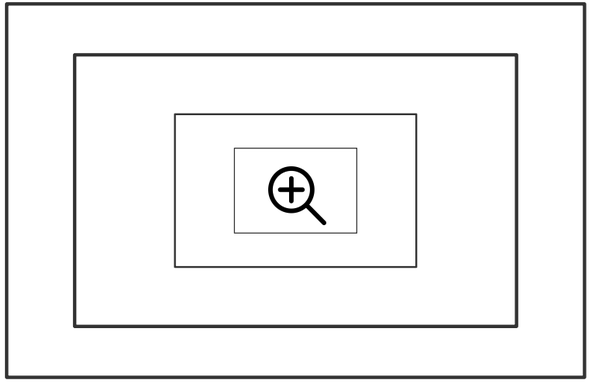
```js
// Créez un objet de zoom
const myZoom = d3.zoom();
const selection = d3.select('element');

// Fonction de rappel pour gérer les événements de zoom et
// Appliquer l'event.tranform {k, x, y} aux éléments graphiques
// k facteur d'échelle, x translation horizontale, y translation verticale
function handleZoom(event) {
    const { transform } = event;
    selection.attr('transform', transform);
}

// Associez la fonction de rappel à l'événement de zoom
myZoom.on('zoom', handleZoom);

// Appliquez le zoom à l'elemént selectionné
selection.call(myZoom);
```

↳ d3 | d3-zoom

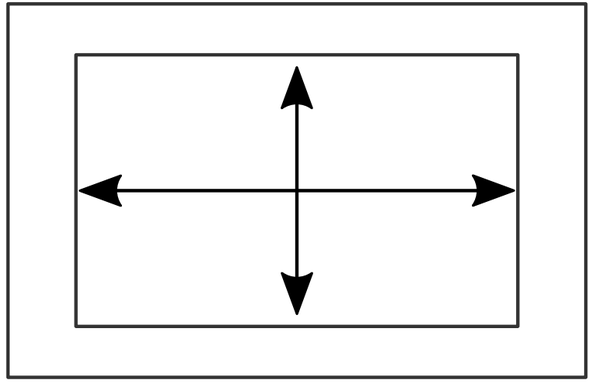
Contraintes

```
zoom().contrainte(props)
```

`scaleExtent([min, max])`



`translateExtent([minX,minY],[maxX, maxY])`



↳ d3 | d3-zoom

Méthodes

```
selection.call(zoom.methode, props)
```

Zoom

```
(zoom.scaleBy, facteur_échelle)
```

Multiplie l'échelle actuelle par le facteur d'échelle

```
(zoom.scaleTo, échelle)
```

Change l'échelle à échelle définie

Pan

```
(zoom.translateBy, x, y )
```

Translation de x, y

```
(zoom.translateTo, x, y)
```

Translation jusqu'à x, y

↳ d3 | d3-zoom

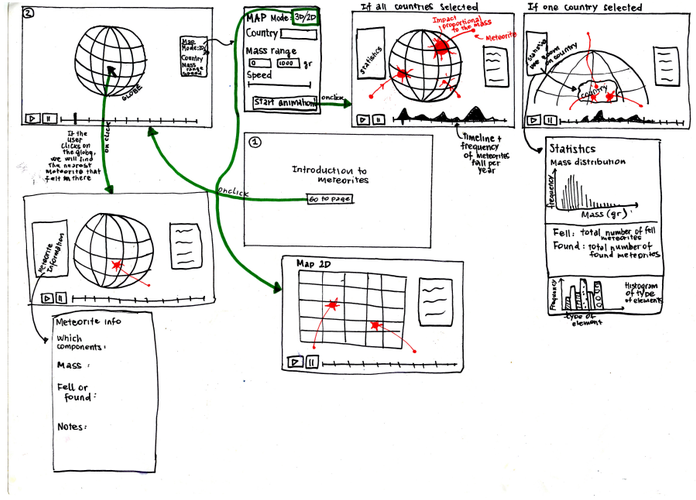
Exemple

↳ Observable | d3-zoom

Projet

Thématique (8 mars)

Wireframe (15 mars)



Figma

invision