

Visualisation de données

07 - Interaction et animation 2

9 avril 2025

Noemi Romano

noemi.romano@heig-vd.ch

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

// 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.transform {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'élément sélectionné
selection.call(myZoom);
```

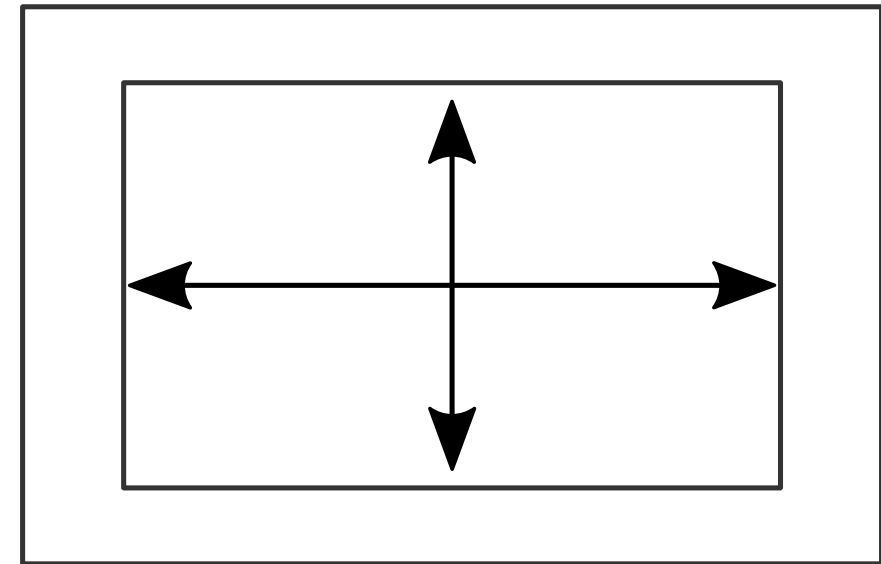
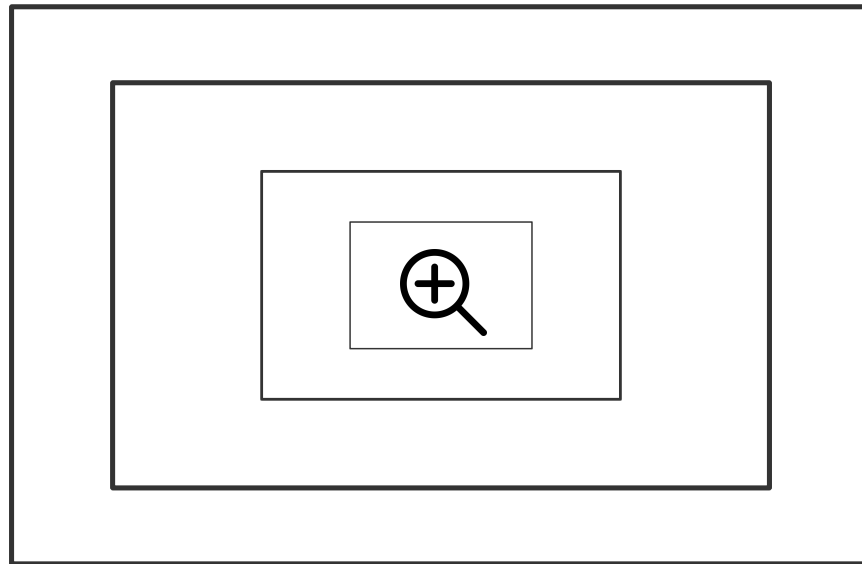
↳ d3 | d3-zoom

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

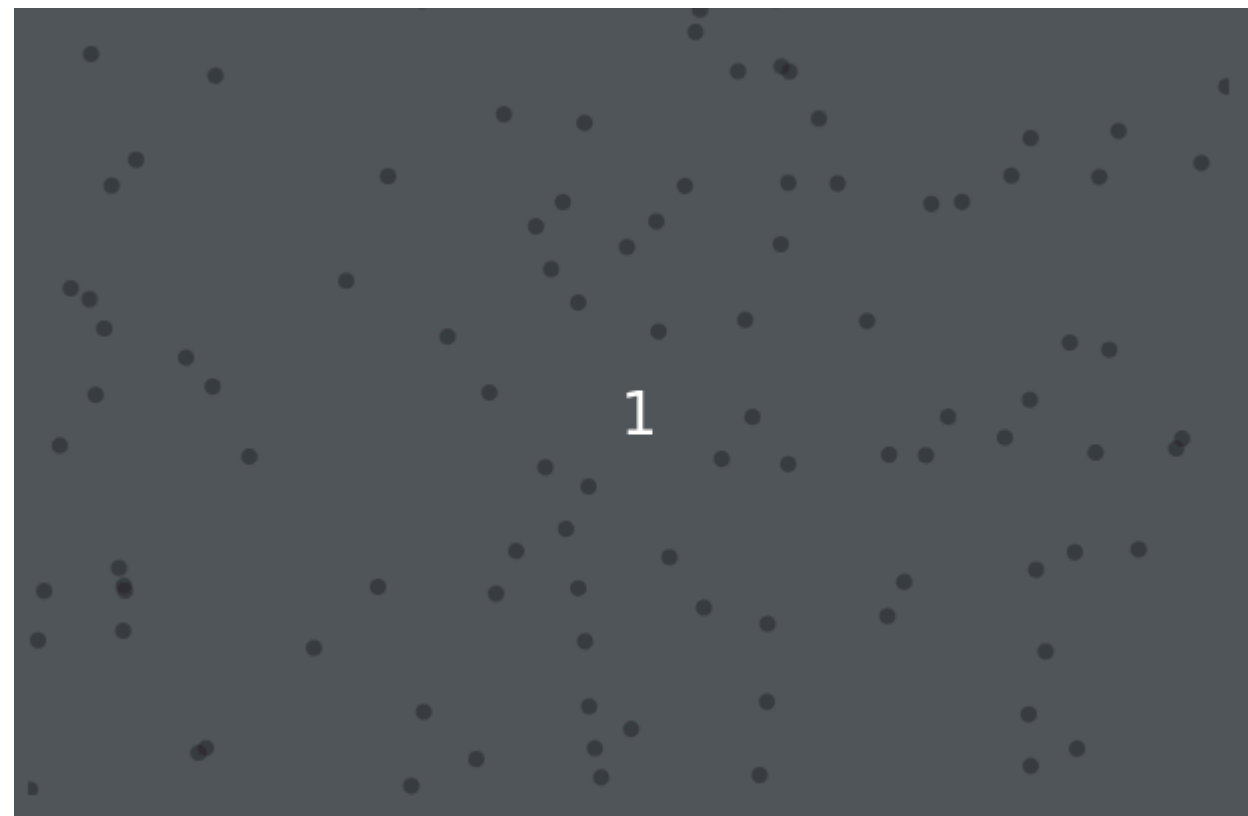
Example

↳ Observable | d3-zoom

d3-brush

Installation

```
npm install d3-brush
```



Initialisation

```
const myBrush= d3.brush()
```

Événements

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

- **typenames**: Les types d'événements à écouter, par exemple, 'brush', 'start', 'end'. Retourne: **event.selection** qui correspond à l'étendue du brush [[x0,y0],[x1,y1]]
- **listener**: La fonction de rappel (callback) appelée lorsqu'un événement spécifié se produit.

```
JS index.js

// Initialise la brosse sans domaine spécifique
const myBrush = d3.brush();

// Fonction de gestion pour l'événement de brosse
function handleBrush(event) {
  // Récupère la sélection de la brosse
  const { selection } = event;

  // Faire quelque chose avec la sélection
  // Dans cet exemple, nous affichons
  // simplement la sélection dans la console ([x0, y0], [x1, y1])
  console.log('Sélection de la brosse :', selection);
}

// Attache la fonction de gestion à l'événement 'brush' de la brosse
myBrush.on('brush', handleBrush);

// Sélectionne l'élément SVG où la brosse sera ajoutée
const monSvg = d3.select('svg');

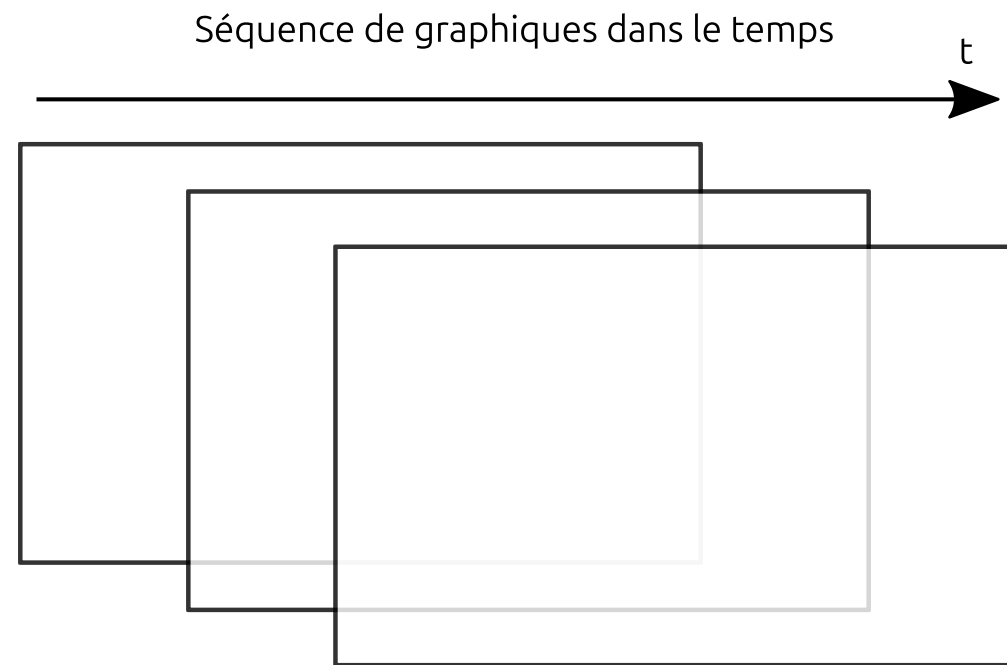
// Ajoute la brosse à l'élément SVG
monSvg.call(myBrush);
```

↳ d3 | d3-brush

Example

↳ Observable | d3-brush

Animation



Animation

setInterval(*fonction*, *intervalleEnMs* [, params])

exécute une fonction de manière répétée avec une intervalle fixée. Retourne un identifiant unique (*intervalId*).

clearInterval(*intervalId*)

arrête l'intervalle

↳ Doc MDN : [setInterval\(\)](#)

Example

↳ Codepen | Animation

Exercices

gapminder

1. Visualisation statique
2. Cartographie
3. Interaction et animation

