



**POLYTECHNIQUE  
MONTRÉAL**

LE GÉNIE  
EN PREMIÈRE CLASSE

# INF8808 – Visualisation de données

## Travail pratique 1

Auteurs:

Adrien Dessinges et Antoine Béland

Chargé de laboratoire:

Jean-Philippe Corbeil

Hiver 2019

Département de génie informatique et génie logiciel

# 1 Objectifs

Le but de ce travail pratique est de présenter les différents langages utilisés par le web actuel, soit le HTML, le CSS et le JavaScript, tout en introduisant la bibliothèque [D3.js](#) (Data-Driven Documents). Ce travail consiste principalement à une mise à niveau afin d'introduire les différents éléments qui seront utilisés dans les prochains travaux pratiques. Une emphase particulière sera accordée au langage JavaScript qui vous sera utile pour manipuler les éléments d'une page HTML (le DOM [*Document Objet Model*]) et pour utiliser la bibliothèque D3. Il est à noter que **ce travail n'est pas évalué**.

Avant de commencer ce travail pratique, il est recommandé d'avoir lu les chapitres 1, 2, 3 et 10 (partie « *Tooltips* ») du livre de Scott Murray [1]. Pour ceux qui ne sont pas familiers avec le langage JavaScript, le livre de Kyle Simpson [2] constitue une bonne introduction à ce langage.

## 2 Introduction

Tout au long de la session, vous aurez à réaliser des visualisations de données à l'aide de la bibliothèque D3.js. Pour ce faire, vous devrez utiliser le langage JavaScript pour être en mesure d'utiliser cette bibliothèque.

D3 est une bibliothèque JavaScript orientée données réalisée par Mike Bostock qui permet la visualisation de données sous forme de dessins vectoriels (SVG). Dans le cadre de ce cours, la **version 5** de D3 sera utilisée.

En ce qui a trait à ce travail pratique, celui-ci est une bonne occasion de se familiariser avec l'utilisation de D3 pour manipuler des éléments simples d'une page HTML. En effet, il est important de bien comprendre les fondements de D3, puisque cette bibliothèque sera très utilisée dans les prochains travaux pratiques.

## 3 Travail à réaliser

Pour ce premier travail pratique, vous aurez à compléter le code d'un générateur de cercles SVG qui permet la création de cercles de taille et de position aléatoires. Ce générateur possède plusieurs fonctionnalités, telles que générer un ou plusieurs cercles, obtenir les informations

d'un cercle en particulier et supprimer tous les cercles qui ont été dessinés. Le générateur est d'ailleurs illustré à la Figure 1.

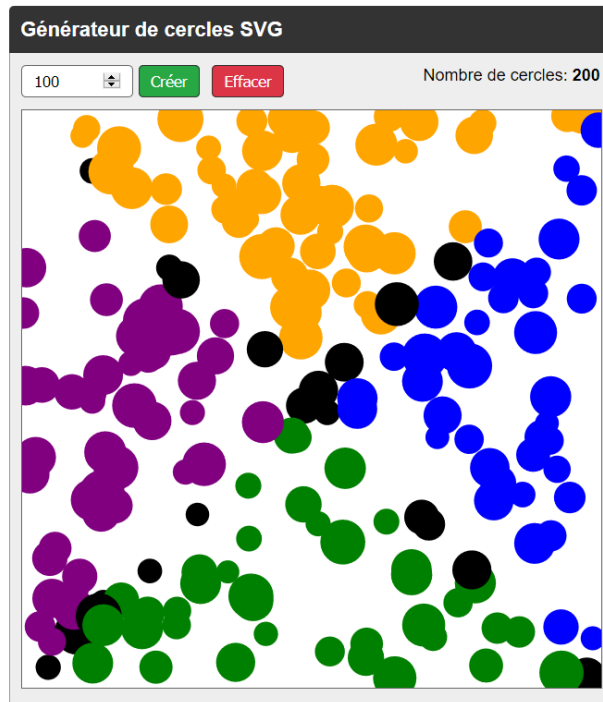


FIGURE 1 – Générateur de cercles SVG ayant créé 200 cercles

Les sous-sections qui suivent décrivent les fonctionnalités manquantes qui doivent être réalisées. Assurez-vous de compléter les différents « TODO » qui se trouvent dans le fichier « assets/scripts/main.js » afin que le générateur soit fonctionnel.

### 3.1 Affichage et mise à jour du nombre de cercles dessinés

Le premier élément à réaliser est l'indicateur affichant le nombre de cercles qui ont été dessinés dans l'élément SVG. Pour trouver le nombre de cercles déjà dessinés, vous devez d'abord accéder aux différents cercles présents dans l'élément SVG puis les stocker dans une variable pour pouvoir y accéder dans d'autres parties du code. En ce sens, vous devez sélectionner tous les cercles dessinés en utilisant les outils de la bibliothèque D3 puis enregistrer ce résultat dans la variable « circles ».



#### Attention

Bien qu'il soit possible de sélectionner des objets du DOM autrement, il vous est demandé d'utiliser la bibliothèque D3 pour manipuler ces éléments.

Une fois la liste des cercles enregistrée, vous devez sélectionner avec D3 l'élément du DOM affichant le nombre de cercles dessinés puis mettre à jour le nombre indiqué. Vous devez sélectionner cet élément avec son identifiant CSS, soit « `#circles-count` ».

Il est nécessaire de réaliser cette mise à jour chaque fois qu'un cercle est créé ou est effacé. Ainsi vous devez implémenter la fonction « `update()` » qui est responsable de réaliser cette mise à jour. Assurez-vous d'appeler cette même fonction lors de la création ou de la suppression d'un cercle.

## 3.2 Suppression des cercles dessinés

La deuxième fonctionnalité à réaliser est la suppression de tous les cercles qui ont été dessinés (voir la fonction « `deleteCircles()` »). Pour ce faire, vous devez confirmer que l'utilisateur souhaite bel et bien supprimer tous les cercles à l'aide d'une boîte de confirmation (voir [window.confirm](#)). Une fois que la suppression aura été confirmée, vous devez supprimer tous les cercles se trouvant dans la variable « `circles` » en utilisant les fonctions de D3. N'oubliez pas d'appeler la fonction « `update()` » une fois que la suppression est complétée.

## 3.3 Génération de plusieurs cercles simultanément

La troisième fonctionnalité à implémenter est la création de plusieurs cercles simultanément. Pour le moment, il est possible de créer uniquement un cercle lorsque le bouton « Ajouter » est cliqué. En effet, la valeur indiquée dans le champ permettant de spécifier la quantité de cercles à créer n'est pas prise en compte. Vous devez donc compléter la fonction « `createCircles()` » afin de prendre en compte la quantité spécifiée dans le champ. Cette fonction devra faire appel à la fonction « `generateRandomCicle()` »  $n$  fois, où  $n$  correspond au nombre de cercles à créer.



### Attention

---

Vous devez vous assurer que la valeur spécifiée dans le champ permettant d'indiquer le nombre de cercles à générer est un nombre valide (entier supérieur à zéro). Dans le cas où une valeur invalide serait spécifiée, vous devez afficher un message d'erreur à l'aide d'une boîte de dialogue (voir [window.alert](#)).

---

### 3.4 Affichage des informations d'un cercle

La dernière fonctionnalité à réaliser est l'affichage des informations d'un cercle lorsque ce dernier est survolé par la souris. Ainsi, lorsqu'un cercle sera survolé par la souris, une infobulle (*tooltip*) contenant les informations suivantes devra faire son apparition :

- le rayon du cercle ;
- la position du cercle ;
- la couleur du cercle.

Vous devez compléter les fonctions « `d3.tip()` » et « `textTip()` » pour réaliser cette fonctionnalité. L'infobulle affichée devra être similaire à celle illustrée à la figure 2.



FIGURE 2 – Infobulle à afficher lorsqu'un cercle est survolé

## 4 Remise

Comme cela a été mentionné au début de ce document, ce travail pratique n'est pas évalué. Ainsi, aucune remise n'est demandée.

## Références

- [1] S. Murray, *Interactive Data Visualization for the Web : An Introduction to Designing with D3*. O'Reilly Media Inc., 2013.
- [2] K. Simpson, *You Don't Know JS : Up & Going*. O'Reilly Media Inc., 2015.