

Architecture et programmation du web (HLIN510)

TP 4

Rendus 2D en Javascript avec la bibliothèque **D3.js**

Pierre Pompidor

Le but du TP est de créer un damier constitué **d'hexagones** un peu à l'image de celui affiché sur la page <http://d3js.org/>. Le rayon (car un hexagone est inscrit dans un cercle) et le nombre d'hexagones seront paramétrables. Les hexagones pourront être sélectionnés en affichant dans la console un id qui sera attribué dynamiquement. Dans tout ce qui suit, ... décrit du code à définir.

Utilisation de la librairie D3 :

Téléchargez le zip sur <https://d3js.org> puis évidemment dézippez-le (avec par exemple *unzip*) et liez la bibliothèque à votre page :

```
<script src="d3.min.js"></script>
```

Génération dans une liste des six points d'un l'hexagone centré en 0 :0 :

Ecrivez une fonction qui appelée avec un rayon en paramètre, renvoie une liste six points :

```
function creeHexagone(rayon) {
  var points = new Array();
  for (var i = 0; i < 6; ++i) {
    var angle = ...
    var x = Math.sin(angle) * rayon;
    var y = -Math.cos(angle) * rayon;
    points.push( ... ); // ajouter à points une liste
                        // contenant l'abscisse et l'ordonnée du point
                        // arrondies à deux décimales
  }
  return points;
}
```

Essayez-la en l'appelant par exemple avec 30 pixels pour le rayon.

Création de la balise SVG (sous une division identifiée par "damier") :

dans le body de votre page, créez une division :

```
<div id="damier" />
```

puis créez une fonction **genereDamier** qui appelée lors du chargement de la page, va créer via la bibliothèque D3 une balise SVG dimensionnée par rapport aux nombres de lignes et de colonnes d'hexagones :

```
function genereDamier(rayon, nbLignes, nbColonnes) {
  d3.select("#damier").append("svg").attr("width", ... ).attr("height", ... );
  ...
}
```

Dessin des hexagones :

Dans la fonction **genereDamier()**, appelez une seule fois **creeHexagone()** pour créer un modèle d'hexagone dont vous allez ensuite modifier les coordonnées pour créer chaque hexagone du damier :

```
var hexagone = creeHexagone(rayon);
for (let ligne=0; ligne < nbLignes; ligne++) {
  for (let colonne=0; colonne < nbColonnes; colonne++) {
    ... // création de chaque hexagone du damier via une balise path ou polygone :
        // voir indications sur la page suivante
  }
}
```

Dessin d'un hexagone via la balises SVG path ou polygon :

Chaque hexagone doit être créé avec D3.js qui va générer une balise SVG *path* ou *polygon*.

Structure d'une balise path en SVG :

En SVG la balise **path** a la structure suivante :

```
<path d="liste des coordonnées des points à relier" fill="white" stroke="black" />

Structure de la liste de points : d="Mx1,y1 Lx2,y2 x3,y3 x4,y4 x5,y5Z x6,y6Z"
      M : MoveTo, L : Line, Z : fermeture du tracé
```

Structure d'une balise polygon en SVG :

En SVG la balise **polygon** a la structure suivante :

```
<polygon points="liste des coordonnées des points à relier" fill="white" stroke="black" />

Structure de la liste de points : points="x1,y1 x2,y2 x3,y3 x4,y4 x5,y5 x6,y6"
```

Construisez la liste de points à relier :

Dans la balise *path* ou *polygon* les coordonnées des points à relier par des segments de droites sont à décrire dans un attribut nommé resp. *d* ou *points*.

```
let listeCoord = "";
let x, y;
for (h in hexagone) {
  // Si ligne impair, décalage des hexagones d'un rayon
  if (ligne % 2) x = ... // calcul de l'abscisse (ligne impaire)
  else          x = ... // calcul de l'abscisse (ligne paire)
  y = ... // calcul de l'ordonnée
  listeCoord += ... // ajout des coordonnées du point (abscisse et ordonnée séparés par une virgule)
}
...
```

et injectez *listeCoord* en valeur de l'attribut *d* ou *points* de la balise *path* ou *polygon*.

Attachez des écouteurs d'événements sur la sélection des hexagones :

La méthode **on()** permet d'attacher des écouteurs d'événements à un élément du DOM. Voici ci-après un exemple de son utilisation pour modifier la valeur d'un attribut d'un élément du DOM. Sur la sélection d'un hexagone :

- faites afficher dans la console l'id attribué à cet hexagone;
- modifier la couleur de fond de l'hexagone en rouge.

```
<élémentDuDOM>.on("click", function() {
  d3.select(this).attr(..., ...);
});
```