

Architecture et programmation du web - TP n.5

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 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 et liez-le 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( ... );  
  }  
  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**, appelée autant de fois que nécessaire **creeHexagone** :

```
for (var ligne=0; ligne < nbLignes; ligne++) {  
  for (var colonne=0; colonne < nbColonnes; colonne++) {  
    var hexagone = ...  
  }  
}
```

Construction des balises SVG **path** :

Structure d'une balise **path** en SVG :

En sachant qu'une balise SVG **path** a la structure suivante :

```
<path d="liste des coordonnées des points à relier" style="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 : pour refermer le tracé

Construisez la liste de points "d" :

```
var d = "";
var x, y;
for (h in hexagone) {
  // Si ligne impair, décalage des hexagones d'un rayon
  if (ligne % 2) ...
  else ...
  ...
}
d += "Z";
```

et construisez la balise **path** qui utilise cet attribut.

Attachez des écouteurs d'événements aux hexagones :

La méthode **on** permet d'attacher des écouteurs d'événements à un objet, exemples :

```
... .on("click", function() { ...
    console.log(d3.select(this).attr(...));
});
```