

Memory en JavaScript

BTS SIO :: AP



Memory

- Vous aller réaliser un jeu de memory en JavaScript avec l'élément <canvas> de HTML5.
- Le mieux sera de procéder par étapes.
- **Etape 1** : création des images dont vous aurez besoin
- **Etape 2** : afficher les images côtés faces de manière non aléatoire
- **Etape 3** : afficher de manière aléatoire les images
- **Etape 4** : les cacher et gérer le click de souris pour les retourner
- **Etape 5** : gérer les couples et les faire disparaître
- **Etape 6** : gérer les scores et les détails du jeu



Etape 1 : Générer les images

- J'ai préparé 10 images d'une taille assez importante quitte à redimensionner après.
- Mon memory fera **4 x 4** soit 16 cartes mais en fait 8 images différentes.
- Plus une image dos de carte
- Une vide pour les cartes retirées



0



1



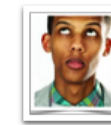
2



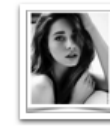
3



4



5



6



7



8



vide.jpg



Nom 5
Type Image PNG (Portable Network Graphics)
Taille 55 Ko
Création avant-hier 12:12
Modification avant-hier 12:12
Dernière ouverture avant-hier 12:12
Dimensions 180 x 211

Etape 2 : Afficher les images de manière non aléatoire de face

- on saute une ligne tous les 4 images
- On génère un petit espace entre chaque élément



Etape 2 : Afficher les images de manière non aléatoire de face

Initialisation

```
// Paramétrage
// Taille des images en pixels
var tailleXImage = 90;
var tailleYImage = 105;

// Initialisations de décalage
var decalage_x = tailleXImage+(tailleXImage/10);
var decalage_y = tailleYImage+(tailleYImage/10);

// Variables pour gérer le canvas de la page HTML
var canvas = document.getElementById('canvasMemory');
var context = canvas.getContext('2d');

// Déclaration d'un tableau de 16 images
var photos = new Array(16);

var images_chargees = 0;
var nb_images = 16;
```



Etape 2 : Afficher les images de manière non aléatoire de face

création du damier

```
// Création du damier avec les images
function creationDamier()
{
    var num=0;
    for (var i=0; i<16; i++)
    {
        // Formule pour associer l'indice i à un nom de photo valide
        // Les photos vont de 1 à 8
        num = (i%8)+1;
        // Création des images
        photos[i] = new Image();
        // On associe à la source
        photos[i].src = "./images/"+num+".png";
        // On appelle la fonction affichage que quand tout est chargée !
        photos[i].onload = function(e)
        {
            images_chargees++;
            if(images_chargees >= nb_images) affichage();
        }
    }
}
```



Etape 2 : Afficher les images de manière non aléatoire de face

affichage

```
function affichage()  
{  
    console.log ("Entre dans affichage !");  
    var x=0 , y=0;  
    for (var i=0; i<16; i++)  
    {  
        context.drawImage(photos[i], x, y, tailleXImage, tailleYImage);  
  
        x += decalage_x;  
        if ((i+1)%4 == 0)  
        {  
            y += decalage_y;  
            x = 0;  
        }  
    }  
}
```

```
// Initialisations de décalage  
var decalage_x = tailleXImage+(tailleXImage/10);  
var decalage_y = tailleYImage+(tailleYImage/10);
```

