



# Projet

## Chasse au Trésor

Rédigé par

**David ROUMANET**  
Professeur BTS SIO

Changement

Date	Révision

## Sommaire

A Objectifs.....	1
B Présentation.....	1
C Partie JavaScript.....	2
C.1 Arrondir un nombre.....	2
C.2 Nombre aléatoire.....	2
C.3 Emplacement d'un élément.....	2
C.4 Tableau à deux dimensions.....	3
C.5 Interaction sur clic.....	4
C.6 Changer un attribut.....	4
D Partie HTML.....	5
E Partie CSS.....	7
E.1 Définir le tableau.....	7
E.2 Définir les cellules.....	7
E.3 Définir un bloc.....	7
E.4 Utiliser les couleurs.....	8
F Structure du code.....	9
G Résultat.....	10

---

## A Objectifs

---

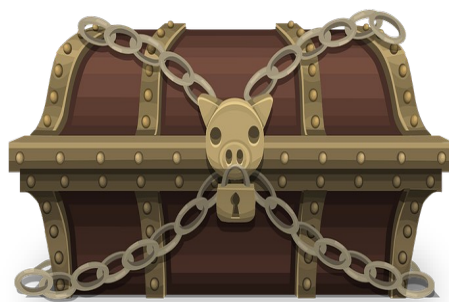
Cette activité doit permettre le développement d'un jeu utilisant les langages HTML/CSS, JavaScript et PHP.

---

## B Présentation

---

Le jeu est très simple : un trésor est caché sur une île et il faut le trouver.



L'île sera représentée par une grille de  $10 \times 10$  cases, et le trésor occupera une seule case, qui sera choisie au hasard en début de partie. L'île aura également des événements – positifs ou négatifs – qui pourront avantager ou désavantager le joueur.

Ce dernier devra trouver le trésor le plus rapidement possible et pourra enregistrer son score (date, nom, score) dans une base commune. Les résultats seront lisibles sur une page score.

Le projet sera créé en plusieurs étapes :

- Créer automatiquement une carte et des cases cliquables
- Placer un item dans une coordonnée de la carte
- Créer des items positifs ou négatifs (pourcentage de présence) et placement
  - On place le trésor en dernier pour éviter un recouvrement
- Créer un score (comptable, etc)
- Créer un formulaire d'enregistrement de score et récupération en PHP pour placement dans une base de données.

## C Partie JavaScript

Un certain nombre de fonctions seront nécessaires en JavaScript, pour créer la table, pour recevoir les événements (clic sur les cases), pour interagir, etc.

JavaScript sera au centre de notre projet. Le code total ne sera pas donné mais des morceaux de codes plus complexes sont proposés ci-dessous.

### C.1 Tableau à deux dimensions

Avant d'afficher un tableau à l'écran, il faut créer un tableau dans la mémoire de l'ordinateur. Pour cela, il faut utiliser une variable qui contiendra un tableau en deux dimensions. Cependant, il n'est pas possible de déclarer et d'utiliser simplement un tableau à 2 dimensions finies.

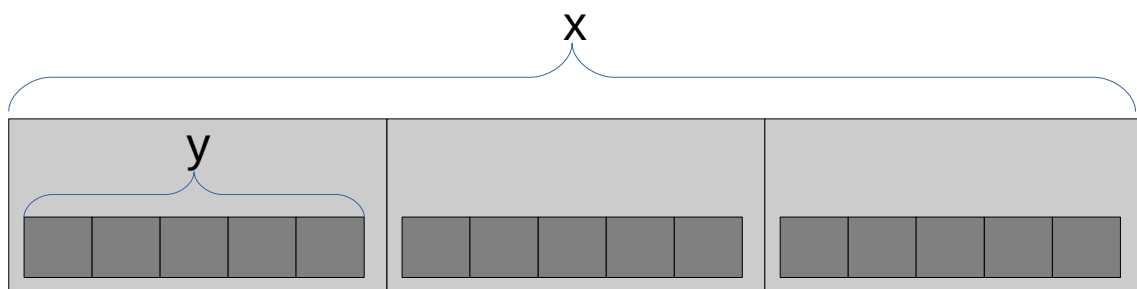
La fonction suivante permet de créer un tableau en 2 dimensions et de le renvoyer :

```
function Tableau2D(x, y) {
  var array2D = new Array(x);
  for (var i = 0; i < array2D.length; i++) {
    array2D[i] = new Array(y);
  }
  return array2D;
}
```

On crée un premier tableau

A l'intérieur de chaque case, on crée un nouveau tableau

Le tableau retourné ressemble à ceci : il s'agit d'un tableau à une dimension qui contient un tableau à une dimension.



Pour utiliser la fonction, il suffit de placer le résultat dans une variable :

```
let monTableau = new Tableau2D(10, 10);
```

L'intérêt de créer un tableau 2D, est de pouvoir cacher d'autres surprises que le trésor dans les cases : un bonus, une potion ou au contraire, du poison ou un malus.

Un autre avantage est de pouvoir y placer une valeur si la case a déjà été visitée.

L'écriture dans le tableau est simple :

```
monTableau[5][2] = "Tresor" ;
```

ou encore

```
let coordonneeX = 5 ;  
let coordonneeY = 2 ;  
//...  
monTableau[coordonneeX][coordonneeY] = "Tresor" ;
```

## C.2

### C.3 Arrondir un nombre

La fonction permettant d'arrondir un nombre décimal est **Math.floor()** : elle arrondit au nombre entier inférieur le plus proche.

Ainsi, `Math.floor(1,9999)` affichera 1.

### C.4 Nombre aléatoire

Pour créer un nombre aléatoire, il faut utiliser la fonction **Math.random()**.

Cette fonction renvoie un nombre décimal entre 0 et 0,999999. Pour simuler un dé (qui va de 1 à 6) il faudrait écrire :

```
De6 = Math.floor(Math.random() * 6 + 1) ;
```

En effet, puisque la fonction `Math.floor()` arrondi à l'entier inférieur et que `Math.random()` ne sera jamais égal à 1, écrire `Math.random() * 6` serait au maximum égal à 5,99999 qui serait arrondi à 5. à l'inverse, si `Math.random()` renvoie 0, la multiplication avec 6 reste toujours égale à 0. Il faut donc ajouter 1 pour avoir un résultat compris entre 1 et 6.

### C.5 Emplacement d'un élément

JavaScript est capable de lire le DOM (Document Object Model) : chaque balise y est référencé. Grâce à cela, il est possible de lire ce qu'une balise contient, ou bien d'en modifier le contenu.

L'instruction suivant recherche le contenu d'une balise dont l'ID est « compteur » :

```
document.getElementById("compteur").innerHTML
```

Exemple de lecture de valeur :

```
monTexte = document.getElementById("emplacementTable").innerHTML ;
```

Exemple de modification de valeur :

```
document.getElementById("emplacementTable").innerHTML = monTexte :
```

## C.6

## C.7 Interaction sur clic

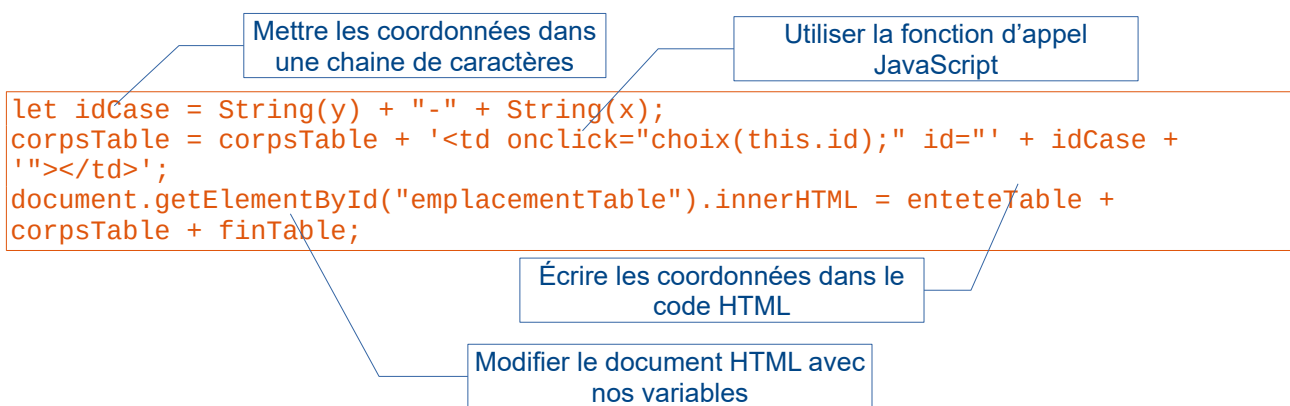
Cette dernière fonction permet (dans le code HTML) de placer un appel à une fonction JavaScript.

```
<td onclick="choix(this.id);" id="5-2"></td>
```

La fonction JavaScript choix(variable) recevra l'ID de l'endroit où il se trouve, ici : 5-2

Cette ligne peut être générée par du code JavaScript : il suffit d'écrire dans une variable JavaScript l'ensemble du code HTML, puis d'utiliser l'écriture à un emplacement de JavaScript.

Ainsi, un exemple d'utilisation serait le suivant :



## C.8 Changer un attribut

JavaScript peut modifier les attributs d'une balise, d'une classe, d'un ID. Voici comment changer la classe d'un élément :

```
caseTable = document.getElementById(idCellule);
caseTable.setAttribute('class', 'good');
```



## D Partie HTML

Elle sera réduite à sa plus simple partie : le code standard et une balise `<div>` ayant comme ID « emplacementTable » et appartenant à la classe « cadre » (nous utiliserons du code CSS sur cette classe).

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>L'ile au trésor</title>
  <script type="text/javascript" src="SI6-IleAuTresor.js"></script>
  <link rel="stylesheet" type="text/css" href="SI6-IleAuTresor.css">
</head>

<body>
  <H1>L'ile au trésor</H1>

  <p>Nombre de coup(s) : <span id="compte">0</span></p>
  <hr>
    <div id="emplacementTable" class="cadre">
      Tableau
    </div>
    <div id="emplacementCommentaires" class="cadre">
      Vous vous apprêtez à creuser avec votre perroquet sur votre
chapeau...<br>
    </div>
  <hr>
</body>
</html>
```

Permet à JavaScript de retrouver où dessiner la table

Permet à CSS d'appliquer un style

Le code sera amélioré par la suite, pour permettre à l'utilisateur de savoir s'il est sur la bonne colonne ou la bonne ligne... mais nous pouvons ajouter une simple légende pour la couleur de la bonne ou mauvaise case :

```
<table>
  <tr>
    <td class="bad"></td>
    <td>Mauvaise case</td>
    <td class="good"></td>
    <td>Bonne case</td>
  </tr>
</table>
```

Notez l'utilisation d'une classe pour définir la couleur de la case !

Au passage, ce code rappelle le fonctionnement d'un tableau en HTML : Table, ligne puis colonne (ou disons plutôt, « cellules »).

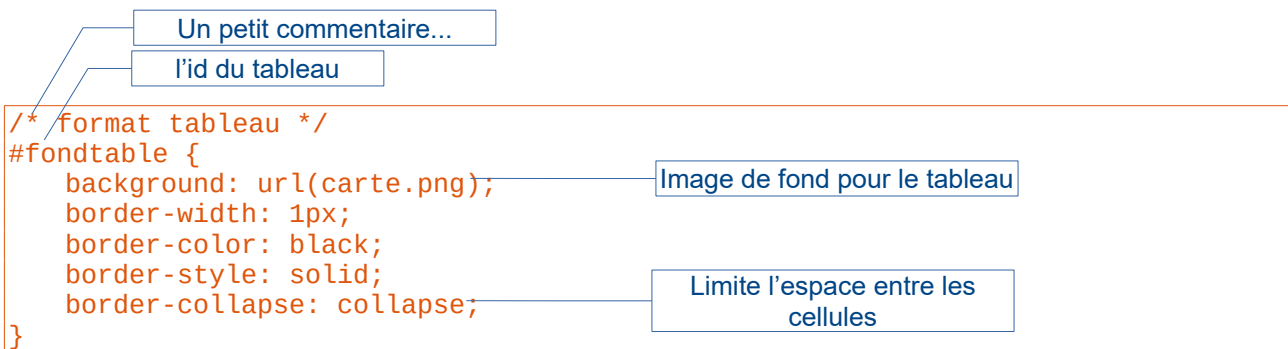


## E Partie CSS

Ce fichier va permettre de placer les éléments et rendre l'aspect visuel plus attractif :

### E.1 Définir le tableau

Ici, l'idéal étant de ne modifier que le tableau généré par JavaScript, il est recommandé d'utiliser un ID :



```
/* format tableau */
#fondtable {
  background: url(carte.png);
  border-width: 1px;
  border-color: black;
  border-style: solid;
  border-collapse: collapse;
}
```

Un petit commentaire...  
l'id du tableau

Image de fond pour le tableau

Limite l'espace entre les cellules

### E.2 Définir les cellules

Cette section montre comment créer des cases carrées dans un tableau :

```
/* Largeur et hauteur des cases de la grille */
td {
  width:40px;
  height:40px;
  border-width: 1px;
  border-color: black;
  border-style: solid;
}
```

### E.3 Définir un bloc

Pour le `<div class="cadre">` on favorise un bloc dont on fixe la largeur et qui accepte d'autres blocs à côté de lui.

```
/* Gestion des blocs de classe "cadre" */
.cadre {
  display: inline-block;
  width: 450px;
  /*margin: 1em;*/
  font-size: 18px;
}
```

Inline : les blocs peuvent être côte-à-côte  
Block : les blocs seront dans la même hauteur

## E.4 Utiliser les couleurs

L'idée, lorsque le joueur clique sur une case, est de changer la couleur de celle-ci. Comme JavaScript peut modifier tous les attributs du DOM, il pourra changer la classe d'une case. Il faut donc définir les deux classes de jeu : celle d'une mauvaise case, en rouge ; celle de la case gagnante, en cyan ;

Nous allons cependant utiliser le système rgba() pour gérer l'opacité de la case.

```
/* Couleurs des cases, avec opacité de 0.9 */
.bad {
  background:rgba(200,0,0,0.9);
}
.good {
  background:cyan;
}
```

Opacité : 0,9

Bleu : 0

Vert : 0

Rouge : 200

Nous aurions pu utiliser le système hexadécimal habituel : #C80000E5

---

## F Structure du code

---

Désormais, nous avons tous les éléments de codes pour créer notre jeu. Il faut cependant définir le fonctionnement.

Voici les fonctions JavaScript, à vous de les remplir :

```
// Tableau2D renvoie un objet tableau en 2D
function Tableau2D(x, y) {
}

// choix() récupère l'ID de la case cliquée et traite le résultat
function choix(that) {
}

// AfficherCompteur() permet d'afficher la variable compteur à l'emplacement
// voulu.
//                               affichera plus tard des commentaires.
function AfficherCompteur() {
}

// onload vérifie que la page soit complètement chargée avant de lancer la
// fonction
window.onload = function() { initTab(); }

//
-----
// initTab() affiche le tableau et choisit les coordonnées du trésor
function initTab() {
```





---

## G Résultat

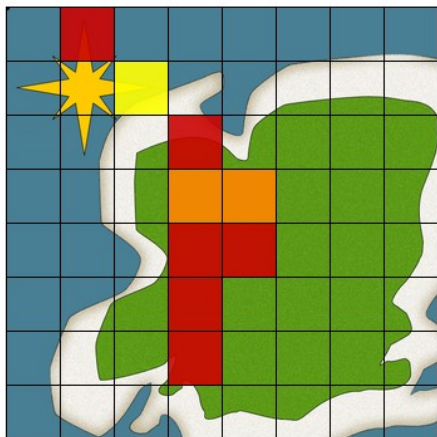
---

## L'île au trésor

Pour jouer, il suffit de cliquer sur une des cases du tableau. La couleur indique le résultat suivant :

	Mauvaise case		Bonne case		Bonne colonne		Bonne ligne
---	---------------	---	------------	---	---------------	---	-------------

Nombre de coup(s) : 9



Vous vous apprêtez à creuser avec votre perroquet sur votre chapeau...

**Coco** : Hey Capt'ain, bonne chance !

**Coco** : Hey Capt'ain, ça creuse ?

**Coco** : Hey Capt'ain, t'es sûr de ce que tu fais ?

**Coco** : Hey Capt'ain, t'es pas un vrai boucanier, n'est-ce pas ?