

# L'API Canvas – PROJET DE JEU

## JEU SNAKE

L'objectif de votre projet est de créer un clone du jeu *Snake*

### Avertissement

Il existe de nombreux tutoriels en ligne pour créer un jeu de Snake en Javascript.

Il n'est pas interdit de vous en inspirer, mais on attend de vous que vous écriviez votre *propre* code, pas que vous recopiez passivement le code du tutoriel.

Notamment, votre note dépendra largement du respect des recommandations ci-dessous (les tutoriels optent généralement pour des solutions différentes).

Votre application comportera au minimum deux “écrans” :

- un écran d'accueil, présentant le jeu et permettant de choisir parmi différents niveaux (cf. ci-dessous),
- un écran de jeu, où se déroule le jeu à proprement parler, en l'affichant dans un canvas 2D, et en affichant le score du joueur.

Le passage d'un écran à l'autre doit se faire sans changer d'URL (vous devez donc *modifier* la page par manipulation du DOM, et non en chargeant une nouvelle page HTML).

# L'API Canvas – PROJET DE JEU

## JEU SNAKE - REPRESENTATION

L'état du monde est représenté dans un tableau JS à deux dimensions. Chaque cellule du tableau représente une “case” du monde, et indique (par sa valeur) ce que contient cette case (mur, serpent, nourriture, vie, mine...). Il vous est recommandé de définir des constantes correspondant à chaque type d'objet, pour améliorer la lisibilité du programme.

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

```
// La représentation qu'aurait cette grille en
Javascript
let WORLD = [
  [EMPTY, EMPTY, EMPTY, EMPTY, EMPTY, EMPTY],
  [EMPTY, EMPTY, EMPTY, FOOD, EMPTY, EMPTY],
  [EMPTY, SNAKE, EMPTY, EMPTY, EMPTY, EMPTY],
  [EMPTY, SNAKE, EMPTY, EMPTY, EMPTY, EMPTY],
  [EMPTY, SNAKE, SNAKE, EMPTY, EMPTY, EMPTY],
  [EMPTY, EMPTY, EMPTY, EMPTY, EMPTY, EMPTY],
];
```

# L'API Canvas – PROJET DE JEU

## JEU SNAKE – REPRESENTATION

Par ailleurs, l'état du serpent est également représenté par un tableau, dont chaque cellule contient les coordonnées d'une case du serpent. Le début de ce tableau correspond à la queue du serpent, et la fin du tableau correspond à sa tête. Lorsque le serpent se déplace, on ajoutera la nouvelle position de sa tête à la fin et on retirera l'ancienne position de sa queue.

```
// La représentation qu'aurait le serpent  
dans la figure ci-dessus.  
let SNAKE = [[4,2], [4,1], [3,1], [2,1]];
```

L'information sur le serpent est donc stockée de manière *redondante* dans ces deux tableaux. Cette redondance rendra la gestion du jeu plus simple par la suite.

# L'API Canvas – PROJET DE JEU

## JEU SNAKE – CONTROLE

L'événement `keydown` est produit chaque fois que l'utilisateur enfonce une touche.

L'objet événement passé en paramètre du `listener` possède un attribut `key`, qui représente la touche enfoncée. Lorsque la touche correspond à un caractère imprimable, `key` contient ce caractère. Sinon, il contient le nom de la touche, comme décrit dans ces tableaux.

En particulier, les flèches sont représentées respectivement par les chaînes `"ArrowDown"`, `"ArrowLeft"`, `"ArrowRight"` et `"ArrowUp"`.

Chaque fois qu'une touche pertinente pour le jeu est enfoncée, vous mémoriserez dans une variable la valeur de l'attribut `key`. Cette valeur sera utilisée dans la fonction `step` (cf. ci-dessous).

# L'API Canvas – PROJET DE JEU

## JEU SNAKE – DEROULEMENT DE LA PARTIE

Toute la mécanique du jeu sera gérée dans une fonction nommée `step`, qui sera appelée à intervalle régulier, grâce à la fonction `setInterval`.

Dans cette fonction, vous devrez effectuer les étapes suivantes.

- Vérifier si l'utilisateur a enfoncé une touche, et modifier la direction du serpent en conséquence (si cela est compatible avec sa direction actuelle).
- Calculer la nouvelle position de la tête du serpent en fonction de sa direction.
- Vérifier si la tête du serpent rencontre de la nourriture, un mur, ou un morceau de son corps.
  - Dans le premier cas, le score augmente, et une autre nourriture est ajoutée dans une case vide aléatoire.
  - Dans les autres cas, la partie se termine.
- Mettre à jour le tableau `SNAKE` en faisant avancer le serpent ; s'il a mangé de la nourriture, son corps doit s'allonger (ce qui revient à ne pas réduire sa queue). Mettre également à jour le tableau `WORLD` en conséquence.
- Effacer intégralement le canvas, et re-dessiner l'état de `WORLD`. (On pourrait envisager de ne redessiner que les parties qui ont changé, mais cette méthode est plus simple et plus évolutive).

# L'API Canvas – PROJET DE JEU

## JEU SNAKE – GESTION DES NIVEAUX

Chaque niveau est décrit par un fichier JSON ayant la structure suivante:

```
{  
    "dimensions": [80, 40],  
    "delay": 200,  
    "food": [ [10,10] ],  
    "snake": [ [60,60], [60,59], [60,58], ]  
}
```

où

dimensions donne la taille de la grille dans laquelle évolue le serpent,  
delay donne le délais (en ms) entre deux appels à la fonction step,  
food donne la position initiale de la nourriture  
snake donne la position initiale du serpent

La page de garde doit donner le choix entre plusieurs niveaux ; lorsque le joueur en choisit un, le fichier JSON correspondant est chargé, et les variables WORLD et SNAKE sont initialisées conformément.

# L'API Canvas – PROJET DE JEU

## JEU SNAKE – BONUS

Modifier le fichier JSON des niveaux pour permettre de traverser les murs, si il traverse le mur le serpent doit réapparaître de l'autre côté du monde

Gestion des vies. Nous pouvons partir du principe qu'un joueur a seulement 3 vies et si il perd les 3 vies il y aura alors un game over.

Toutefois, le joueur pourra gagner des vies durant les partie garde à un « cœur » qui apparaîtra sur le monde du SNAKE. Attention toutefois à faire apparaître ces cœurs de manière aléatoire et temporaire et surtout avec un intervalle de temps assez important entre chaque apparition.

La gestion visuel des pommes et des vies qui seront de simples carrés, essayer de faire en sorte que les éléments ressemblent à de vraies pommes et de vraies cœur. Si possible ne pas utilisé d'image, essayer de les dessiner

Affichage du score et du nombre de vie

A partir d'un certains score (ex tous les 100 points) la vitesse du serpent augmente

Au même titre de les vies, le serpent du manger un fruit (ex une orange) qui apparaîtra rarement sur le jeu et qui fera rétrécir le serpent plutôt que l'allonger. A noter, il est intéressant de prévoir une taille max pour le serpent.