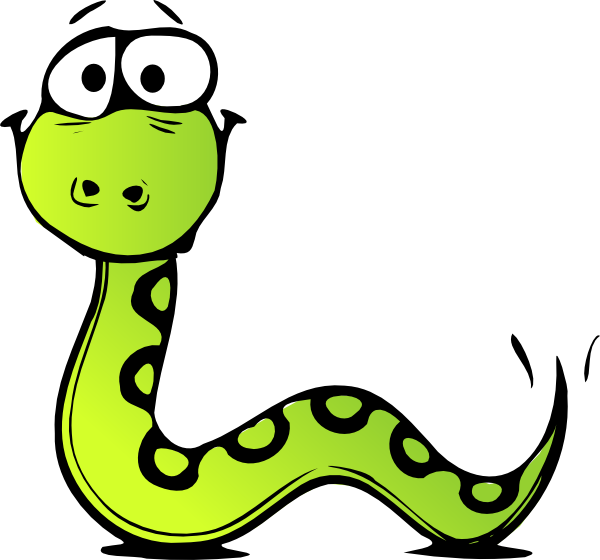
|  |
| --- |
| Evin Paramanathan  09/01/2024 |

|  |
| --- |
|  |
| ETML |
| Rapport de Pr🍎jet |
| Snake en JS |
|  |



Rapport de projet

# Table des matières

[Table des matières 1](#_Toc155689399)

[A. Introduction 2](#_Toc155689400)

[B. Apprentissage JS 2](#_Toc155689401)

[a) Variables 2](#_Toc155689402)

[b) Classes 2](#_Toc155689403)

[c) If 2](#_Toc155689404)

[d) Import / Export 2](#_Toc155689405)

[e) Canvas 2](#_Toc155689406)

[f) Math. 2](#_Toc155689407)

[g) ArrowkeyPressed 3](#_Toc155689408)

[h) Fonction fléchées 3](#_Toc155689409)

[i) ECMAScript 4](#_Toc155689410)

[C. Projet Snake 4](#_Toc155689411)

[a) Snake 4](#_Toc155689412)

[b) Pomme 4](#_Toc155689413)

[c) End Game 5](#_Toc155689414)

[D. Test 5](#_Toc155689415)

[E. Conclusion 6](#_Toc155689416)

[F. ChatGPT 6](#_Toc155689417)

[G. Sources 6](#_Toc155689418)

# Introduction

Dans le cadre de ce projet nous devons faire un snake en Javascript afin de développer nos connaissances dans ce langage, pour se préparer pour les prochains modules qui sont des modules de Javascript avancé. Pour atteindre nos objectifs nous devons nous aider d’Internet pour faire des recherches sur des forums ou des sites dédiés au JS, ou même ChatGPT (son Utilisation a été décrit dans le point 3.f). En plus de faire le projet Snake, nous devons aussi faire un aide-mémoire personnel pour avoir un doc qui contient des informations sur le JS qui nous semble important. Et pour finir l’introduction, si on avait fini le projet Snake en avance comme ça était le cas dans notre classe, on pouvait faire un autre projet qui est similaire à un futur projet qui est un self-service machine. Ce deuxième projet a pour but de faire une borne automatique avec différents types de produit d’un célèbre fast-food, et il fallait faire toute la partie JS pour sélectionner des produits, enlever des produits, additionner les prix, etc…

# Apprentissage JS

### Variables

Il y a 4 façons de déclarer une variable :

* Var : cette variable permet de déclarer une variable et même de lui donner une valeur.
* Let : Perme de déclarer une variable dont la portée est limitée celle du bloque ou elles sont déclarées
* Const : permet de déclarer une constante nommée qui est seulement accessible en lecture.
* Automatiquement : Ça créer automatiquement une variable

Mais le mieux c’est d’utiliser des const et let.

### Classes

Les classes sont juste des fonctions spéciales. Pour définir des classes il suffit de mettre class suivie par le nom que l’on souhaite donner.

Export class Snake {}

Il y a un export juste avant classe parce que comme citer de le point 4 cela va nous permettre de lier avec une autre page comme la main

### If

Un if permet d’effectuer différents type d’actions en fonction d’une condition

**Opérateur ternaire(?)**

L’opérateur ternaire ou point d’interrogation permet de faire une condition comme le for mais de façon plus simple et plus rapide

### Import / Export

En js on utilise des imports et export pour lier les diffèrent page de js, dans le cadre du projet snake on utilise l’export sur la class snake export class Snake {} et pour l’importer dans la page main grâce à import {Snake} from './Snake.js';

### Canvas

### Math.

‘Math n’est pas un constructeur, après le point on peut mettre de propriétés statique.

Telle que :

* PI : 3.14
* LN10 : logarithme naturel de 10 = 2.303
* Ln2 : logarithme naturel de 2 = 0.693
* SQRT2 : racine carrée de 2 = 1.414

Sinon on peut aussi mettre des méthodes statiques

* Abs : renvoie la valeur absolue de x
* Ceil : renvoie le plus petit entier supérieur ou égal à x
* Floor : Renvoie le plus grand entier inférieur ou égal à x
* Random : renvoie une valeur aléatoire

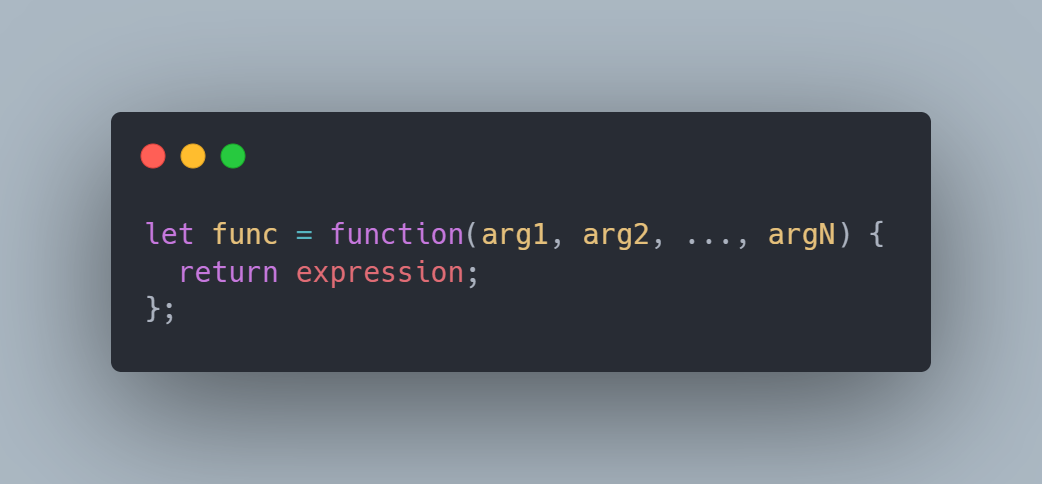
Et il y en a plein d’autre qui permette de faire diffèrent opérations

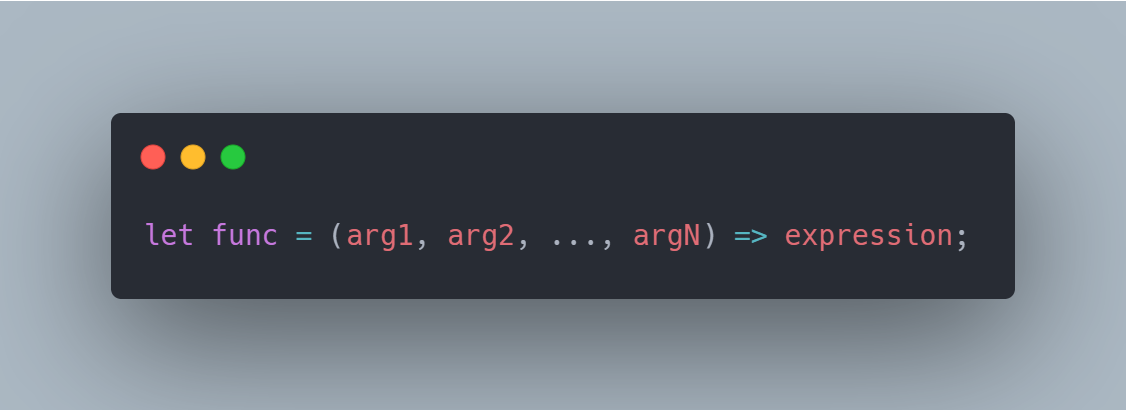
### ArrowkeyPressed

Pour le projet on doit utiliser les flèches pour déplacer le snake pour cela on doit c’est comme en C# on utilise des Switch Case

### Fonction fléchées

C’est une syntaxe plus courte et plus simple pour créer des fonctions.

Par exemple au lieu d’écrire une fonction comme ça : 

On va raccourcir en une seule ligne comme cela :

(Exemple reprit sur le site developer.mozilla.org)

Tout ça va créer une fonction qui accepte les argument Arg1, Arg2, …, ArgN. Ensuite elle va évaluer l’expression et va retourner le résultat.

Si on souhaite faire des fonctions plus compliquée on peut créer crée une fonction ensuite on met les arguments et ensuite mets des accolades, cela va permettre d’ouvrir une fonction fléchées multi ligne. Et si on utilise des fonctions fléchées avec des accolades on doit utiliser un Return à l’intérieur de celles-ci.

En gros il y a deux type de fonction fléchées, une avec des accolades ou sur une ligne il y a les arguments à gauches et à droite un expression (ça évalue et renvoie le résultat.) Et la deuxième est celui avec les accolades, on met les arguments ensuite on met les accolades pour créer une fonction multi ligne, et on n’oublie pas de mettre un return.

### ECMAScript

ECAMSCRIPT est une norme de programmation sur laquelle le JavaScript est basé. ECMA est l’abréviation de European, Computer, Manufactures, Association.

C’est une organisation qui développes de norme. Fun facts ECMA se situe à Genève.

# Projet Snake

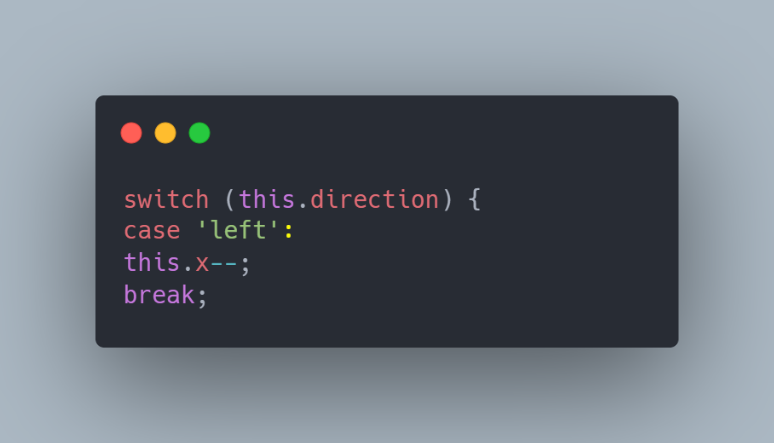
### Snake

Pour les Snake il y a plusieurs méthodes la première est le

* **Constructor**

Cette méthode va nous permettre d’initier la postions de base du snake donc x = 10 et y = 10. Ça va aussi nous permettre d’afficher la taille du snake this.size = 40. Mais le constructor vas aussi nous servir pour déclarer une variable pour le canva

* **Move**

Dans cette méthode on va faire deux types déplacements du snake le premier c’est pour la tête du snake et le deuxième est pour le corps. Pour la tête du snake on va faire que quand on appuie sur une des flèches que cela fasse que la tête du Snake va dans la direction indiquer. Pour cela on va utiliser comme au C# des switches case, d’abord on mettre le switch avec une variable qu’on va appeler direction, ensuite on met case avec la direction « Gauche » et ensuite on met la direction (x--) et on finit par un break. Il faut faire ça avec toute les directions

Ensuite il y a partie pour le corps. Pour le déplacement du corps

* **Grow**

Pour le grow il faut d’abord faire en sorte que chaque fois que le Snake mange une pomme une case s’ajoute et que cette case suit la tête du Snake, ensuite le reste du corps suit à chaque fois la case qui le précède. Pour ajouter la case on utilise un « push » cela permet d’ajouter un ou plusieurs éléments à la fin du tableau.

Une image contenant texte, capture d’écran, Police, conception

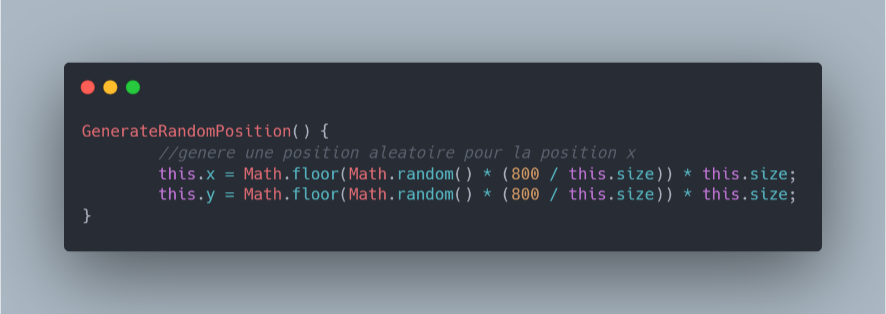
Description générée automatiquement

### Pomme

* **Constructor**

La pomme aussi a un constructeur ou il y a le canvas et la taille mais aussi une variable pour générer une position pour la pomme.

* **Positionnement**

Il y a une méthode qui va permettre de d’afficher la pomme de façon aléatoire sur le terrain de jeu. Pour cela il faut qu’on utilise math.random afin de créer une position x et y on va aussi utiliser un math.floor pour que la position aléatoire soit un nombre entier et s’il y a des virgules c’est arrondi au chiffre inférieur.

### End Game

* + **Mur**

Pour perdre il faut que le Snake touche un mur. Donc pour cela il faut faire une boucle if qui fait que si la position de la tête du snake et égal aux bordures ça arrête le jeu. Et ça affiche Game Over avec le score du joueur et ça dit d’appuyer sur F5 pour rafraîchir la page et recommencer le jeu.

# Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom du test | Description | Etape de test | Résultat Attendu | Résultat actuelle |
| Direction | Changement de direction | 1) Presser sur l’une des flèches de direction | Le snake doit aller du coté ou le testeur a appuyé | En appuyant sur l’une des flèches le snake va bien dans la direction souhaitée. |
| Score | Lorsque le snake mange une pommes les points augmente dans le conteur. | 1) déplacer le snake  2) manger une pomme | Le conteur qui se situe en haut à gauche de l’écran augmente | Le conteur augment bien à chaque fois qu’une pomme est mangé |
| End-Game | Lorsque le snake touche les bords du terrain de jeu, le jeu s’arrête et affiche Game over avec le score | 1) déplacer le snake  2) toucher le bord du terrain | Lorsque le snake touche le bord, le jeu s’arrête et au milieu de l’écran il y a le message Game-over et le score et aussi le moyen de recommencer le jeu | Le jeu s’arrête et le Game-Over est bien afficher avec le score et le moyen de relancer une partie. |

# ChatGPT

Dans le cadre de ce projet les IA plus précisément ChatGPT ma surtout servit pour m’aider à comprendre des choses en JavaScript que je ne comprenais pas. Par exemple en cherchant comment faire grandir mon Snake j’ai trouvé une méthode qui s’appelle push, et sur internet plusieurs personnes l’utiliser donc j’ai demandé à ChatGPT qu’est-ce que ça fait et comment l’utiliser. Cela ma permit de comprendre un peu comment utiliser le push.

# Conclusion

Ce projet a permis d’apprendre le node JS afin de développer le jeu Snake. Pour cela on a dû faire pas mal de recherches

# Sources

<https://fr.javascript.info/>

<https://developer.mozilla.org/fr/docs/Web/JavaScript>

ChatGPT