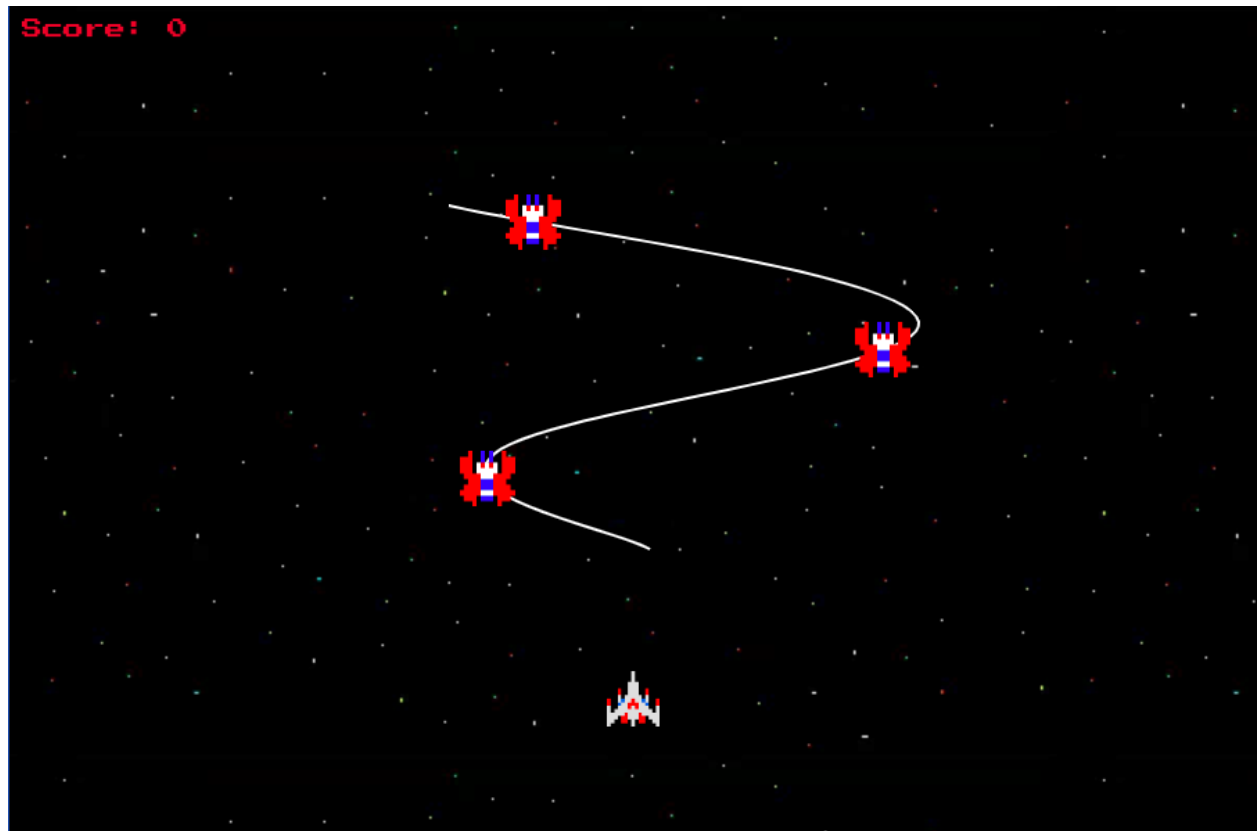


# RAPPORT DE PROJET TUTORÉ

## 2020/2021

---



Réalisé par Neffati Youssef.

---

## Introduction

Le 8 juillet 2021, lors d'une réunion zoom avec M. Buffa, il nous a été demandé de réaliser un projet individuel. Il s'agit de développer en JavaScript orienté objet, un ensemble de classes utilitaires et un éditeur interactif pour animer à 60 images seconde des objets suivant des courbes (quadriques, courbes de bézier), définies à la souris.

Scénario : on dessine une courbe à l'écran, avec des points de contrôle, la courbe apparaît. Puis ensuite on clique sur "animer un objet le long de cette courbe", et on a un objet qui va suivre la courbe en suivant la courbure.

- On ajoutera un temps pour aller du point de départ au point d'arrivée, plus le temps est court, plus la vitesse est rapide.
- On essaiera de faire ralentir l'objet dans les virages prononcés (en gros, ajuster la vitesse en fonction de la courbure)
- On ajoutera une option pour "chaîner les courbes" et donc générer des animations plus complexes.
- Un ensemble de courbes aura un nom, ce sera le nom de l'animation définie. Par exemple "attaque", "rejoindreFormation", "bougerEnFormation".

---

## Techniques utilisées

Pour développer ce site internet, j'ai utilisé :

- le langage CSS pour la partie visuelle du site.
- le langage HTML pour construire les pages web.
- le langage JAVASCRIPT pour développer le jeu, et rendre la page plus dynamique avec les animations.

J'ai travaillé avec l'IDE Visual Studio Code. Git m'a aidé pour la gestion de projet, c'est-à-dire à publier une mise à jour de mon code chaque semaine.

## Architecture logicielle

**L'architecture du site internet est constituée de :**

- 1 page HTML (index.html)
- Un répertoire Assets contenant tous les objets externes nécessaires au projet.
- Un répertoire CSS contenant le style.css de la page web.
- Un répertoire JS qui contient toutes les classes JS nécessaires aux fonctionnalités de la page web.

**Dans le dossier JS nous avons :**

**Script.js** : On va créer les instances de classe dans la fonction init() qui s'exécutera à window.onload , et passer les bons paramètres aux différentes instances.

**Canvas.js** : C'est la classe qui va gérer la partie visuelle du jeu, on lui passe en argument le contrôleur. Les animations se déroulent ici. À sa construction, cette classe va initialiser le canvas avec la méthode init() ainsi que l'animation. C'est dans cette classe que sont définies les méthodes permettant l'affichage des objets comme : drawAlien(), drawPlayer(),

---

drawBullets() et draw(). Ces méthodes seront appelées dans la méthode update(), qui récupère les nouvelles coordonnées des objets du contrôleur et dessine les formes.

**Outils.js** : C'est la classe qui contient toutes les fonctions du projet. On peut y retrouver les fonctions liées aux clics des boutons.

**ecouteurs.js** : C'est la classe qui va contenir les écouteurs. Elle utilise les écouteurs keydown et keyup pour le déplacement du joueur par les flèches et la barre d'espace pour tirer. Elle utilise aussi les écouteurs click dont les fonctions et mouseclick() pour sauvegarder les coordonnées de la position de la souris à chaque clic.

**controleur.js** : C'est la classe qui va représenter la logique du jeu, on va lui passer dans le constructeur une instance de l'objet Canvas. Les méthodes qu'il contient permettent de contrôler l'état du jeu. On peut y retrouver checkCollisionsJoueur(), checkCollisionsBullet(), addBullets(), move().

**Alien.js** : Cette classe permet simplement d'établir le constructeur de l'alien. Elle contient seulement le constructeur de la classe. L'objet sera dessiné dans la classe Canvas avec la méthode drawAlien().

**joueur.js** : Cette classe permet simplement d'établir le constructeur du joueur. Elle contient seulement le constructeur de la classe. L'objet sera dessiné dans la classe Canvas avec la méthode drawPlayer().

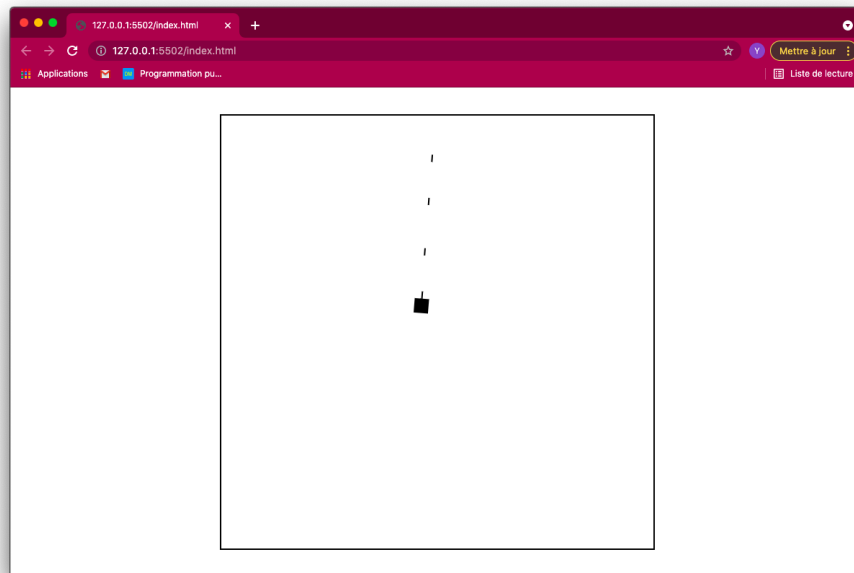
**Bullet.js** : Cette classe permet simplement d'établir le constructeur du joueur. Elle contient le constructeur de la classe, la méthode draw() et la méthode move(). L'objet sera dessiné dans la classe Canvas avec la méthode drawBullets() qui utilisera la méthode draw() de la classe Bullet.

---

## Planning de réalisation des scénarios

### 9 juillet au 16 juillet :

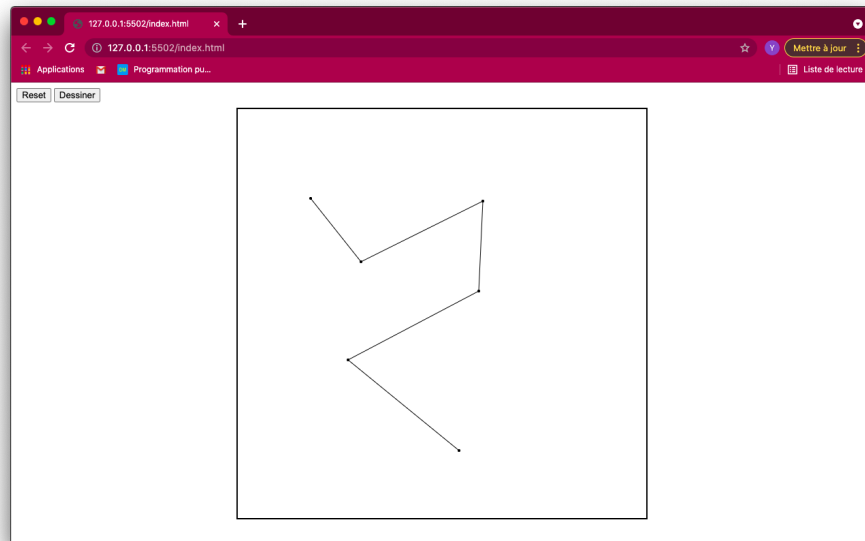
Au cours de cette semaine j'ai créé mon repository Github. J'ai listé les premières tâches à réaliser pour démarrer le projet. J'ai commencé par prendre la base du code d'un char qui tire des missiles en appuyant sur la barre d'espace et qui suit les coordonnées de la souris.



### 16 juillet au 23 juillet :

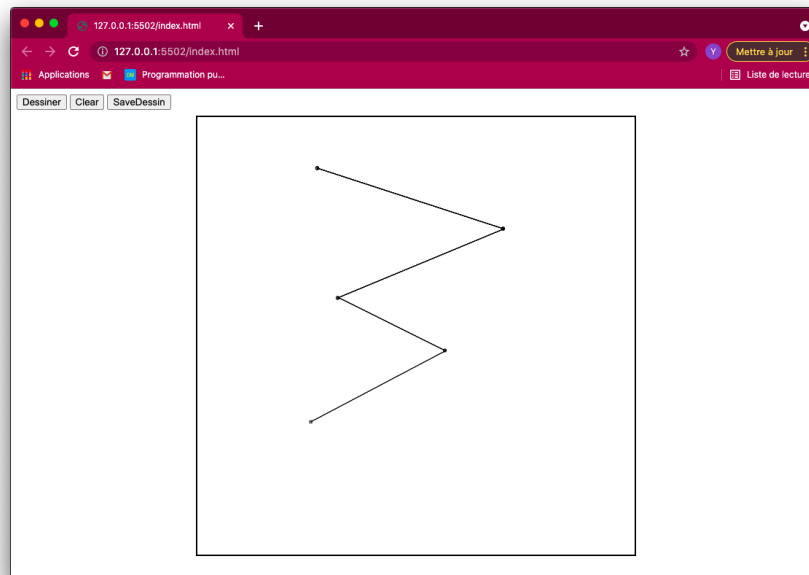
Au cours de cette semaine je suis parti sur une nouvelle base en commençant par créer un canva qui dessine des points à chaque clics et qui relie les points entre eux.

J'ai commencé à créer deux boutons "Reset" et "Dessiner" mais qui ne fonctionnaient pas encore lorsque j'avais commit mon travail. En effet, la prochaine étape était de pouvoir sauvegarder mon dessin et de le redessiner si besoin.



### 23 juillet au 30 juillet :

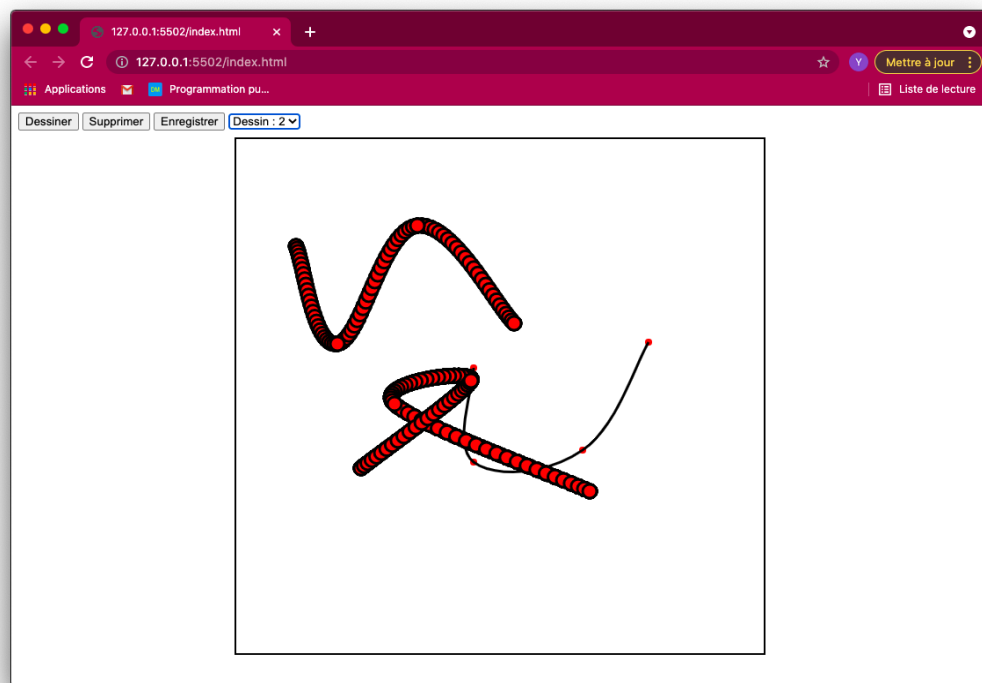
Cette semaine, visuellement on ne voit pas grand chose car le travail s'est plus orienté dans la partie de sauvegarde du dessin. En effet, j'ai ajouté un bouton "SaveDessin" qui permet de sauvegarder les points sur le canvas dans un tableau. À ce moment-là, mon prochain objectif était de pouvoir sauvegarder tous ces points dans un seul index d'un tableau. Ainsi, je pourrai sauvegarder plusieurs dessins.



---

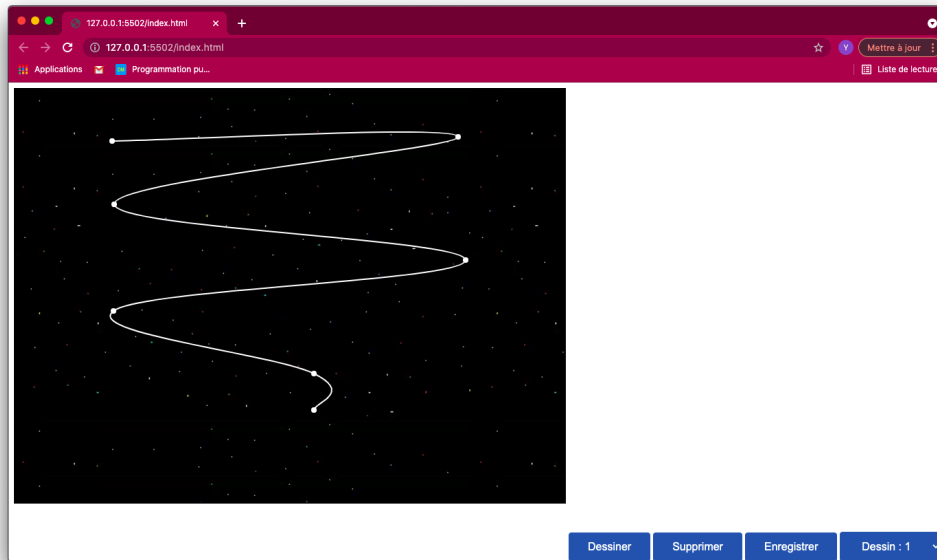
### **30 juillet au 6 août :**

Cette semaine, j'ai réussi à sauvegarder plusieurs dessins dans un tableau multidimensionnel et à les mettre dans une sélection pour pouvoir les ré-afficher. J'ai aussi réussi à transformer les droites en courbes grâce à une fonction que j'ai trouvée sur [stackoverflow](#). Cependant, lorsque j'ai commencé à faire de l'animation il y avait pas mal de bugs et je ne comprenais pas mes erreurs ce qui m'a un peu démotivé car j'y ai passé pas mal d'heures dessus.



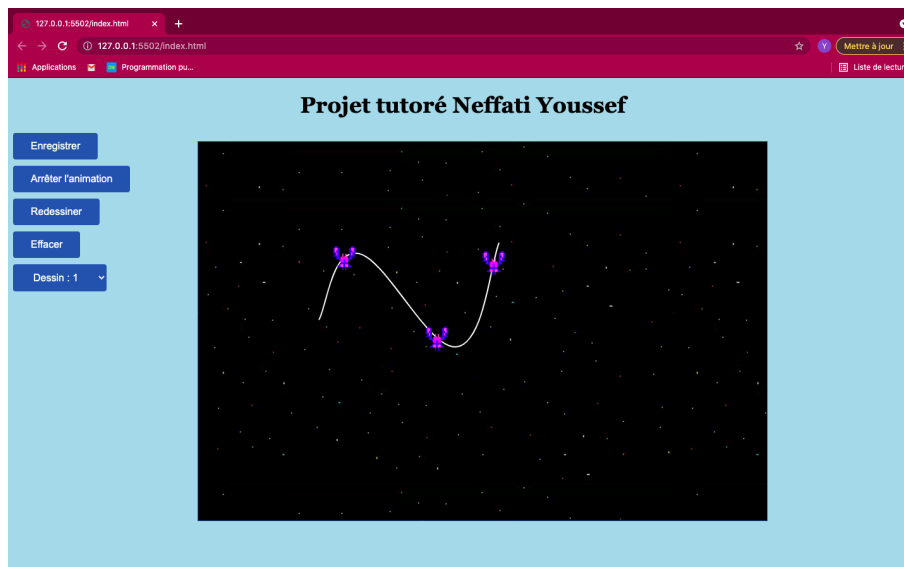
### **6 août au 13 août :**

Cette semaine, j'étais toujours bloqué sur les animations. Pour être honnête, j'ai un peu moins coder que d'habitude. Tout ce que j'essayais ne fonctionnait pas ou apportait plus de bug. J'ai donc décidé de travailler sur la structure du code et de supprimer tout ce qui était inutile histoire de bien comprendre ce que je faisais car j'avais l'impression de m'être noyé dans mon propre code. J'ai aussi ajouté une image de fond à mon canvas et enlevé les animations.



### 13 août au 20 août :

Cette semaine j'ai réussi à résoudre mon problème d'animation. Par la suite, j'ai donc remplacé le dessin d'un point rouge par une image d'alien galaga et j'ai ajouté un bouton permettant de stopper/reprendre l'animation. J'ai aussi fait du CSS pour avoir quelque chose de plus agréable à regarder. J'ai aussi modifié mon code pour avoir une structure moderne avec des classes (ecouteurs.js et alien.js). Il faudrait que j'ajoute un joueur (vaisseau), que j'ajoute les collisions avec l'alien et aussi pouvoir tirer des missiles.





---

## 20 août au 27 août :

Cette semaine s'est bien déroulée. J'ai (encore) restructuré mon code, ajouté un joueur et des tirs de missiles, j'ai encore modifier la présentation de la page pour qu'elle soit plus esthétique. Cependant, je note qu'il reste certains détails d'animation à corriger mais que je n'arrive pas à résoudre comme le fait de faire disparaître l'alien lorsqu'il est touché par une balle. Il est aussi possible de modifier la courbe mais il y a une petite erreur : lorsque je drag & drop un point de contrôle, c'est toujours le 1er point de contrôle du tableau de point qui se déplace.



## Conclusion

Pour conclure, ce projet a été une expérience très enrichissante pour moi. Il m'a permis de tester mes limites et de voir jusqu'où j'aurais pu aller. J'ai fait de mon mieux et j'ai essayé d'être le plus régulier possible. Le plus difficile pour moi à été au début car il faut partir de zéro et aussi la partie animation où j'ai passé plus d'une semaine à résoudre mes problèmes. Néanmoins, cela m'a permis de mieux comprendre le fonctionnement des animations en javascript. Pour le reste, je suis plutôt satisfait du travail que j'ai fourni même si il y a des choses à améliorer. Ce travail à été réalisé sans demander aucune aide même si j'aurais dû le faire mais d'un autre côté cela m'a permis de tester mes capacités.