

# Compte rendu projet Macron49.3

---

Nicolas Ainouz Info1



Ce compte rendu décrit les spécificités du projet Macron49.3, ainsi que les choix techniques et les difficultés rencontrées. Ce projet a été réalisé dans le cadre du cours de Programmation C de première année de l'ESIGE de la formation INFO. Il a été entièrement pensé et développé par Nicolas AINOUS.

Je tiens particulièrement à remercier les enseignants M. Borie ainsi que M. Revuz pour leur aide et leur disponibilité.

## Présentation du projet

L'objectif de ce projet est de réaliser entièrement en programmation C un jeu de type "Shoot'em up". Ce type de jeu est un jeu de tir dans lequel le joueur contrôle un personnage dans un niveau à défilement horizontal. Le joueur doit éviter les obstacles et tirer sur les ennemis qui apparaissent à l'écran. Le jeu se termine lorsque le joueur perd toutes ses vies ou lorsqu'il termine le niveau.

Le but du jeu est d'obtenir le meilleur score en restant en vie le plus longtemps possible. Le score est calculé en fonction du temps de jeu.

## Spécificités du projet

Le jeu est conçu avec un design pattern utilisant des vues et des contrôleurs. La vue principale du jeu est appelée par des contrôleurs qui ont des rôles spécifiques. Le jeu est composé de 5 contrôleurs principaux :

- `enemies_controller` : s'occupe de la gestion des ennemis.
- `key_listener` : s'occupe de la gestion des entrées clavier.
- `player_controller` : s'occupe de la gestion du joueur.
- `powerup` : s'occupe de la gestion des bonus.
- `rocket_controller` : s'occupe de la gestion des missiles.

Chaque entité du jeu est représentée par une structure. Pour les entités qui sont présentes en nombre, un tableau de 100 éléments est initialisé au démarrage du jeu. Les entités sont ensuite ajoutées ou supprimées du tableau en fonction de leur présence à l'écran ou de leur état. Ce choix d'implémentation n'est pas optimal car je dois parcourir le tableau entier pour chaque entité à chaque frame. J'avais résolu partiellement ce problème en utilisant une variable indice qui me permettait d'obtenir la position dans le tableau du dernier ennemi créé. Cependant, cette variable a été supprimée car elle posait des problèmes de gestion de mémoire. Une bonne solution aurait été d'utiliser des listes chaînées, mais je n'ai pas eu le temps de l'implémenter. Toutes mes structures ainsi que des variables importantes sont stockées dans ma structure "game", ce qui me permet d'y accéder facilement depuis n'importe quel contrôleur.

Au-delà du principe de base du jeu de type "Shoot'em up", d'autres mécaniques ont été ajoutées.

### L'inertie

L'inertie a été ajoutée pour ajouter de la difficulté au jeu. Le joueur ne peut pas s'arrêter instantanément et doit anticiper ses déplacements. C'est la fonction `void apply_inertie(Game *game)` dans le `key_listener` qui se charge de l'appliquer. Cette fonction décrémente la vitesse du joueur progressivement jusqu'à ce qu'elle soit nulle.

### Les bonus

Les bonus sont des entités qui apparaissent aléatoirement à l'écran. Ils sont représentés par la structure "Powerup" et s'activent en pressant la touche shift de son clavier. Il existe 3 types de bonus :

1. Dragon Ball : L'utilisateur se transforme en super saiyan, devenant ainsi invincible et envoyant une puissante attaque.
2. Le bonus de vitesse : Bien qu'il puisse être un bonus comme un malus, il permet d'augmenter drastiquement la vitesse du joueur pendant un court instant.
3. Le bonus de vie : Le joueur récupère 1/5 de sa vie.

### Shoot too much

Lorsque le joueur tire trop de missiles, il ne peut plus en tirer pendant un court instant. Cela permet d'éviter que le joueur spamme la touche de tir et rend le jeu plus difficile. Le joueur peut voir le nombre de missiles restants à tirer grâce à une barre de chargement qui s'affiche en haut de l'écran.

## Barre de vie

Certains ennemis étant plus coriaces que d'autres, il est important de pouvoir voir la vie restante de certains ennemis. Certains ennemis sont donc pourvus d'une barre de vie qui s'affiche au-dessus d'eux.

## Missiles spéciaux

Toutes les trajectoires des missiles ne sont pas rectilignes. Des missiles à tête chercheuse doivent être évités par le joueur pendant quelques secondes avant d'exploser.

## Améliorations possibles

Le jeu est fonctionnel, mais il reste encore beaucoup d'améliorations possibles. Voici une liste non exhaustive des améliorations que je souhaiterais apporter au jeu :

1. Implémenter des listes chaînées pour les entités.
2. Ajouter des niveaux.
3. Ajouter des boss.
4. Ajouter des ennemis plus variés.
5. Ajouter des bonus plus variés.
6. Ajouter des obstacles à éviter.
7. Animer mes entités avec des sprites.
8. Afficher les meilleurs scores (les scores sont pour l'instant écrits dans un fichier .txt).

## Conclusion personnelle

Ce projet a été très enrichissant pour moi. Il m'a permis de penser et de concevoir un projet entièrement seul et d'apprendre de mes erreurs. D'un point de vue technique, il m'a permis de mieux appréhender le principe des structures, de la gestion de mémoire, mais aussi de la programmation événementielle. J'aurais aimé approfondir davantage ce projet pour livrer un projet dont je suis fier qui reflète tout ce que j'aurais aimé faire. Le thème du projet m'aurait permis de créer une variété d'ennemis et de bonus. Je vais par la suite continuer à améliorer ce projet pour le rendre le plus complet et le plus intéressant possible. Je pense qu'avec une meilleure organisation ainsi qu'en travaillant avec un binôme, ce projet aurait pu être encore plus complet. Je suis tout de même satisfait du résultat final et je suis content d'avoir pu réaliser ce projet.