

Compte Rendu Shootem-up



Ce compte rendu synthétise le projet de Shootem-up de DJEBLOUN Yacine et RODDIER Corentin. Ce projet a été donné aux étudiants de la promotion de 2022 de l'ESIPÉ en 1^{ère} année, dans le cadre du cours pédagogique de programmation C.

Nous remercions nos professeurs de programmation C ainsi que les personnes tierces qui ont contribué au projet.

Sommaire

Sommaire	2
Présentation	3
Structuration du projet	3
Quelques cas concrets	4
Points d'améliorations	4
Avis personnels	5

Présentation

L'objectif de ce projet est de réaliser un jeu de type Shoot'em up, dont les attendus sont spécifiés dans le sujet disponible [ici](#)

Pour réaliser ce projet en binôme, nous avons développé l'ensemble des attendus obligatoires, tout en les adaptant à notre objectif de rendu. Afin de travailler correctement en binôme, nous avons utilisé l'outil de versionning et de stockage GitHub. Le répertoire utilisé pour le développement est disponible [ici](#)

Afin de personnaliser un maximum le projet, nous avons fait le choix de demander à des personnes tierces de nous assister dans la direction artistique du projet. C'est pourquoi, les décors du jeu ainsi que la musique ont été entièrement réalisés pour ce projet.

Structuration du projet

Dans ce projet, nous avons choisi de construire notre code en suivant au maximum le design pattern MVC (Model-View-Controller). Concrètement, cela signifie que notre code est organisé de manière à ce que le Controller soit le fichier qui dirige le projet et appelle le Model et la View. Le Model stocke et initialise l'ensemble des différentes données utiles au projet (le joueur, les ennemis, etc). Concernant la View, elle s'occupe de dessiner l'intégralité de la fenêtre de jeu et ne fait aucun calcul.

Concernant l'algorithme global du projet, l'ensemble des allocations de mémoire sont effectuées avant le lancement du jeu. Ceci nous permet de fluidifier le jeu et de ne faire aucune allocations durant la partie (les allocations sont coûteuses en termes d'utilisation mémoire vive). L'ensemble des allocations sont libérées à la fermeture ou à la fin du jeu.

Durant ce projet, nous avons eu besoin d'utiliser de nombreuses structures (Player, Position, Scenery, etc) qui sont toutes incluses dans la structure globale de notre projet : Party. Ceci nous permet de passer notre structure Party à toutes les fonctions, et ainsi accéder à toutes les informations du jeu, n'importe où dans le code du projet.

Quelques cas concrets

Comme dit précédemment, toutes les allocations sont effectuées au lancement du projet. C'est donc à ce moment là que l'ensemble des ennemis sont créés, ainsi que les missiles du joueur et des ennemis. Ces derniers sont stockés dans un tableau d'une taille égale à 100. Au moment de leurs créations, tous les missiles et les ennemis sont rendus invisibles (avec un entier utilisé en booléen ; 1 = visible ; 0 = invisible). Au moment où un nouvel ennemi/missiles doit être créé, nous parcourons l'ensemble de son tableau jusqu'à tomber sur un élément qui à l'entier visible à 0. Dans ce cas, nous mettons visible à 1 et nous réinitialisons l'ensemble des ses caractéristiques. Dans le cas où nous ne trouvons pas d'élément ou visible égal 0, aucun ennemis ou missiles n'est ajouté.

Concernant la détection des touches, au lieu de détecter les événements et de faire des actions en fonction de cet événement, nous avons opté pour une autre méthode. Nous stockons les événements dans un tableau pk (pour pressedKey), inclus dans la structure Party. Ceci nous permet d'accéder à l'état de nos touches n'importe où dans le code.

Enfin, dans notre projet, nous avons fait le choix d'afficher les meilleurs scores à la fin d'une partie. L'ensemble de ces scores sont stockés dans un fichier score.txt, qui contient les 10 meilleurs scores du jeu enregistré. Pour gérer l'affichage, l'algorithme commence par lire le contenu du fichier et mettre les informations dans un tableau contenant 10 ScoreBoard. Nous trions ensuite ce tableau en comparant avec la taille du score à la taille du tableau -1. Ceci nous donne donc un tableau trié que nous affichons dans la vue. Il nous reste ensuite plus qu'à écrire le tableau dans le fichier score.txt pour sauvegarder le nouveau score.

Points d'améliorations

Dans le cadre du projet, il serait intéressant d'ajouter certaines améliorations proposées dans le sujet ou que nous aurions aimé voir dans notre création.

C'est notamment le cas de l'inertie ou des chocs élastiques qui peuvent ajouter davantage d'immersion au joueur. Au vu de la direction artistique de notre jeu, nous ne nous sommes pas penchés vers cette fonctionnalité.

En ce qui concerne les bonus, nous avons encore des idées à explorer (multi-tir ou compétence spéciale) et nous aurions souhaité ajouter des effets sonores pour tous les bonus. Actuellement ce n'est le cas que pour certains d'entre eux (nous vous laissons découvrir lesquels).

En suivant le sujet, nous aurions aussi aimé ajouter les missiles têtes chercheuses aux ennemis afin d'ajouter une complexité pour le joueur (bien que le jeu soit assez dur comme ça).

D'un point de vue technique, actuellement nous avons un tableau d'une centaine d'ennemis ce qui limite le nombre d'entités simultanément. Il aurait été préférable de modifier le code afin de travailler avec une liste chaînée plutôt qu'un tableau et ne pas être limité bien que la liste chaînée nous aurait contraint à faire nos allocations en temps-réel et potentiellement créer une latence dans les pires cas.

Avis personnels

Yacine :

Ce projet m'a permis de me familiariser avec le langage C dans des conditions de production réelles. Il a été primordial d'assurer l'adaptabilité du code ainsi que sa segmentation ce qui n'est pas toujours le cas dans nos séances de TP. Je demeure tout de même frustré de ne pas avoir davantage ajouté d'éléments tangibles au projet et il est fort probable que je revienne dessus à l'avenir. Je me dois de remercier mon binôme avec qui cela a été un plaisir de travailler.

Corentin :

Ce projet a été pour moi une occasion pour moi de me familiariser et de mieux comprendre le système de pointeurs et d'adresses. Il m'a également permis de me familiariser avec git que des points que je ne connaissais que très peu. Cependant, je regrette de ne pas avoir eu plus de temps pour travailler sur ce projet car je pense qu'on aurait pu ajouter de nombreuses fonctionnalités intéressantes (missiles à têtes chercheuses, choix du joueur au démarrage du jeu).