

Rapport Projet Tempest

Gregory Adam - Esther Lenôtre

Intervenant pour la soutenance : Anna Leonteva

I. Présentation du projet

La visée de ce projet était de créer une reproduction du jeu **Tempest** sorti sur *Atari* aussi fidèle que possible.

Le projet dans son état actuel propose un menu de départ, un niveau avec un joueur mobile pouvant tirer des munitions sur un seul type d'ennemi, lui aussi mobile. Il y a également une gestion du score et les ennemis peuvent mourir ainsi que le joueur, donnant alors lieu à un écran de *game over*. La taille des ennemis et projectiles est variable et le comportement des ennemis est complet (apparition, montée jusqu'au bord externe, poursuite du joueur sur ce dernier).

II. Organisation

Nous avons organisé des réunions régulières afin de se mettre d'accord sur l'architecture du projet (les premiers fichiers *include* ont été programmés à deux). Je me suis occupé de la répartition des tâches. Entre chaque réunion, nous avançons sur nos parties respectives (c'est-à-dire les *.cpp* et d'éventuelles classes supplémentaires si nécessaire).

Au cours de ce projet, j'ai pu implémenter les fonctionnalités suivantes :

- **Couleur** : classe `Color` permettant de créer des couleurs pour simplifier l'implémentations d'éléments graphiques par la suite
- **Entités** : classe `Entity` regroupant munitions et ennemis (toute entité se déplaçant dans un couloir)
- **Jeu** : classe `Game` permettant de définir l'espace de jeu et de le lancer
- **Lignes** : classe `Line` permettant de tracer des traits entre deux points
- **Carte** : classe `Map` permettant de générer un niveau sur lequel les entités pourront se déplacer
- **Munitions** : classe `Munition` permettant de regrouper l'ensemble des munitions en jeu
- **Joueur** : classe `Player` regroupant les attributs du joueur (statut, score, position...) et qui gère ses actions
- **Contrôle du joueur** : classe `PlayerController` permettant de gérer les entrées clavier qui commandent le joueur
- Tests dans un main temporaire
- *CmakeList*
- *README.md*

Et les fonctionnalités que j'ai ajouté à des classes prototypées par ma camarade :

- **Texte/Font** : aide pour régler la taille des lettres
- **Enemy** : ajout de `move()` et `trackPlayer()`

III. Choix et difficultés

J'ai eu des difficultés à manipuler les différents types de pointeurs comme les `shared_pointer` ou les `unique_pointer`. Choisir quel type de pointeur était approprié a été une autre difficulté. J'ai été amené à manipuler la géométrie et la trigonométrie pour la carte et les entités qui s'y déplacent. Ceci a également posé problème durant la conception de ce projet, notamment la gestion de la profondeur et de l'adaptation de la taille et de la vitesse en fonction de la position de l'objet dans le couloir. Certains choix opérés tôt dans la conception de l'architecture du projet ont été revus à la fin de ce dernier : plusieurs classes étaient devenues obsolètes et j'ai dû simplifier la structure du projet.