

# 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). Gregory s'est 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).

Ci dessous la liste des fonctionnalités que j'ai pu implémenter intégralement :

- **Boutons** : classe `Button` permettant de dessiner des boutons et de gérer les évènements relatifs à la souris
- **Texte** : classe `Font` et `Text` permettant de dessiner le texte et d'encapsuler ses propriétés (couleur, position...)
- **Scènes** : classe `Scene` permettant de définir un écran de jeu (menus, game over, HUD...)
- **Point** : classe `Point` permettant d'ensuite tracer les lignes et donc les éléments de jeu
- **Ennemis** : classe `Enemy` permettant de générer un ennemi et son comportement
- **Manager d'ennemis** : classe `EnemyManager` permettant de générer des ennemis aléatoirement, de gérer leurs déplacements et de les supprimer et incrémenter le score lorsqu'ils ont été tués par le joueur.
- Tests dans un main temporaire
- Dessin de l'ennemi
- Dessin des menus (boutons, couleurs et textes) et affichage/update du score en jeu

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

- **Entity** : ajout de `move()`
- **Game** : modifications dans `start()`
- **Player** : ajout de `move()`, `shoot()`, `update()`, `die()`, `getScore()`, `addScore()`, `getGameOver()`
- **PlayerController** : prise en compte de la barre espace dans `checkInput()`

### III. Choix et difficultés

La construction du menu m'a pris plus de temps qu'escompté : je n'étais pas familière avec la librairie SDL et j'ai donc mis du temps à comprendre le fonctionnement des textures et des évènements. L'utilisation de la police Hershey a été aussi assez laborieuse, il m'a fallu du temps pour comprendre comment bien tracer les différents caractères. Il a fallu ensuite faire des choix pour encapsuler les classes relatives au tracé du texte afin de simplifier la création des scènes qui comportent boutons et texte. J'ai eu quelques difficultés à trouver comment lier tout ces éléments : la solution fut la création d'une classe intermédiaire `Text`.

La réflexion autour du traitement des objets rotatifs (notamment les ennemis) a également occupé une partie conséquente de notre temps sur ce projet. La solution proposée dans le sujet a fini par fonctionner après de nombreux essais peu fructueux.