Une image contenant Graphique, Police, logo, capture d’écran

Le contenu généré par l’IA peut être incorrect.**one piece fight**

Sayf SBAI

Ziane Berreghis

Maxime-Sorith

Dans le cadre de notre projet de groupe, nous avons développé un jeu de combat au tour par tour inspiré de l’univers *One Piece*. L’objectif principal était de concevoir un système de jeu à la fois stratégique, équilibré et facile à prendre en main. Le joueur compose une équipe de trois personnages et affronte soit une autre équipe contrôlée par un joueur humain (mode PvP), soit par l’ordinateur (mode PvE), avec différents niveaux de difficulté.

Chaque personnage possède des statistiques (attaque, défense, agilité), une attaque de base, ainsi que plusieurs techniques spéciales. Ces techniques sont soumises à des délais de rechargement (cooldowns), afin d’éviter les abus et de forcer le joueur à réfléchir à l’ordre de ses actions. Les dégâts sont réduits par la défense de l’adversaire, et l’agilité influe sur les chances d’esquive. Lorsqu’un personnage est mis K.O., le système adapte dynamiquement les tours restants.

Le jeu repose sur une architecture modulaire en C, et les données des combattants sont chargées depuis un fichier texte structuré, ce qui permet de facilement ajouter ou modifier des personnages sans toucher au code source.

**Répartition des tâches**

Nous avons réparti les responsabilités de manière équilibrée selon les compétences de chacun :

**Sayf** a été en charge du système de combat, de l’implémentation du mode Joueur contre Joueur, ainsi que du développement du bot.

**Ziane** a travaillé sur l’interface utilisateur, la gestion des équipes et l’implémentation du mode Joueur contre ordinateur.

**Maxime** s’est occupé du chargement des données via fichiers texte, de la gestion mémoire et du débogage général.

Nous avons utilisé Git pour le versionnage du projet et organisé des réunions quotidiennes pour suivre l’avancement et répartir les tâches.

**Problèmes rencontrés**

**1. Gestion des pointeurs et de la mémoire**

Nous avons rencontré plusieurs plantages dus à des accès à des pointeurs non initialisés (NULL).  
**Solution :** nous avons mis en place des vérifications systématiques des pointeurs avant toute utilisation, et utilisé free() pour libérer correctement la mémoire.

**2. Équilibrage des techniques**

Certaines statistiques étaient trop puissantes, notamment dès le début du combat (ex. : Nami avait 110 d’agilité).  
**Solution :** nous avons rééquilibré les statistiques et introduit des cooldowns pour définir la durée des techniques et réduit les dégâts de certaines techniques.

**3. Affichage dans l’interface**

Le texte des menus n’était pas bien aligné, ce qui nuisait à la lisibilité.  
**Solution :** nous avons utilisé le formattage avec printf (%-20s) pour un affichage propre, et créé des fonctions dédiées à la présentation (afficher\_separateur, afficher\_ligne\_texte, etc.).

**Avancement par étapes**

Le développement s’est déroulé sur quatre semaines :

* **Semaine 1** : Conception des structures de base (Combattant, Équipe).
* **Semaine 2** : Implémentation des combats PvP et création de tests.
* **Semaine 3-4** : Développement du bot, ajout des niveaux de difficulté, optimisation mémoire.
* **Semaine 5** : Finalisation de l’interface, corrections de bugs, améliorations visuelles.

**Intelligence artificielle**

L’IA propose trois niveaux :

* **Noob** : Effectue des actions aléatoires, sans stratégie.
* **Facile**: Attaque constamment le plus faible.
* **Moyen** : Analyse les points faibles de l’équipe adverse, cible les personnages vulnérables, et utilise les techniques disponibles de manière optimale.

**Bilan**

Ce projet nous a permis de renforcer nos compétences en langage C, notamment sur la gestion mémoire, la manipulation de structures, l’optimisation du code et le travail en équipe. Nous avons aussi appris à équilibrer un gameplay, concevoir une IA basique mais fonctionnelle, et à structurer un projet logiciel de manière collaborative.