

Rapport de projet : Jeu de Combat de Personnages

Réalisé par Rayane, Lucas et Joao

1. Introduction

Notre projet consiste en un jeu de combat textuel développé en langage C, permettant aux joueurs d'affronter l'ordinateur avec différents personnages issus de la pop culture. Le jeu propose deux modes principaux :

- Un combat d'équipe 2v2
- Un mode tournoi où le joueur peut participer à une compétition contre des adversaires contrôlés par l'IA

Malgré quelques difficultés rencontrées pendant le développement, nous avons réussi à créer un jeu fonctionnel avec des mécaniques de combat intéressantes, des animations simples mais efficaces, et un système de tournoi complet.

2. Méthodologie de travail

Notre équipe était composée de trois membres ayant chacun des responsabilités spécifiques :

Membre	Responsabilités
Rayane	Développement du mode tournoi et gestion des combats dans ce mode, développement des modules combat.c/h et utilis.c/h
Lucas	Correction des bugs, implémentation des combats en équipe 2v2, développement des modules equipe.c/h et main.c
Joao	Interface utilisateur, animations et affichage, développement des modules tournoi.c/h et perso.c/h

Nous avons travaillé avec une approche modulaire, où chaque membre pouvait développer sa partie relativement indépendamment grâce à la séparation du code en plusieurs fichiers .c et .h. Cette organisation nous a permis de travailler en parallèle et d'intégrer facilement nos contributions.

3. Structure du projet

3.1 Fichiers principaux

Fichier	Description
main.c	Point d'entrée du programme, initialise le jeu et gère le menu principal
perso.c/h	Gestion des personnages, leurs caractéristiques et leurs attaques spéciales
combat.c/h	Système de combat, actions et calcul des dégâts
tournoi.c/h	Mode tournoi complet
equipe.c/h	Gestion des équipes pour le mode 2v2
affichage.c/h	Fonctions d'affichage et interface utilisateur
utilis.c/h	Fonctions utilitaires (gestion des entrées, animations, etc.)

3.2 Organisation du code

Le code est structuré autour de plusieurs structures de données principales :

```
c

typedef struct {
    char* nom;
    int cooldown_max;
    int effets;
    char* description;
    int cooldown_actuel;
} SpecialAttack;

typedef struct {
    char* nom;
    int PVmax;
    int PV;
    int att;
    int def;
    int agi;
    int vit;
    int jauge;          // Jauge d'attaque pour le système de tours
    SpecialAttack* special;
    int special_cd;
    int shield_active;
} Perso;

typedef struct {
    char nom_equipe[MAX_TEAM_NAME];
    int taille;
    Perso** membres;
} Equipe;

typedef struct {
    char nom[50];
    int nb_participants;
    int format;
    Perso** participants;
    int* points;
    int* victoires;
    int* defaites;
    int* matchs_joues;
} Tournoi;
```

4. Fonctionnalités implémentées

4.1 Système de combat

Le système de combat offre trois actions possibles :

- 1. **Attaque normale** : Inflige des dégâts basés sur les statistiques d'attaque et de défense
 - Coûte 10 points de jauge
 - Peut être esquivée selon l'agilité du défenseur
 - Génère 10 points de jauge pour le défenseur s'il est touché
- 2. **Bouclier** : Réduit les dégâts subis au prochain tour
 - Génère 5 points de jauge pour l'utilisateur
- 3. **Attaque spéciale** : Action plus puissante avec un temps de recharge
 - Coûte 20 points de jauge
 - Inflige des dégâts doublés ou active un effet spécial
 - Génère 15 points de jauge pour le défenseur s'il est touché

Chaque personnage possède des statistiques uniques qui influencent ses performances :

- **PV** : Points de vie
- **ATT** : Puissance d'attaque
- **DEF** : Capacité défensive
- **AGI** : Agilité (chance d'esquive)
- **VIT** : Vitesse (influence la génération de jauge)

Le nouveau système de jauge d'attaque permet un déroulement des tours basé sur l'accumulation d'énergie plutôt que sur un ordre prédéfini. Le personnage qui atteint 50 points de jauge peut agir, puis sa jauge redescend à 25 points après son action.

4.2 Mode Tournoi

Le mode tournoi permet au joueur de participer à une compétition avec 4 participants (le joueur et 3 adversaires contrôlés par l'IA). Le tournoi se déroule en format "poule" où chaque participant affronte tous les autres. Des points sont attribués pour chaque victoire, et un classement final est établi.

Exemple de classement de tournoi :

=== CLASSEMENT DU TOURNOI Tournoi des Champions ===				
Joueur	MJ	Vic	Def	Pts
Pikachu	3	3	0	9
Luffy	3	2	1	6
El Primo	3	1	2	3
Muffin Man	3	0	3	0

4.3 Mode Combat 2v2

Le mode 2v2 permet au joueur de créer une équipe de deux personnages pour affronter une équipe contrôlée par l'IA. Le niveau de difficulté peut être ajusté, ce qui influence les statistiques des personnages adverses.

Ce mode introduit une dimension stratégique supplémentaire :

- Les membres d'une équipe gagnent des points de jauge lorsque leurs coéquipiers agissent
- Il faut choisir judicieusement les cibles à attaquer

4.4 Interface et animations

Nous avons intégré plusieurs éléments pour améliorer l'expérience utilisateur :

- **Animations ASCII** pour les combats
- **Barres de vie colorées** avec emojis pour une visualisation claire des PV
- **Jauges d'attaque** avec représentation visuelle
- **Menus avec codes couleur** pour faciliter la navigation
- **Affichage du podium** à la fin du tournoi

5. Difficultés rencontrées

Au cours du développement, nous avons rencontré plusieurs défis :

1. **Gestion de la mémoire** : Les fuites de mémoire ont été un problème récurrent, notamment avec les personnages et les équipes. Nous avons dû implémenter des fonctions de libération (`liberer_persos`, `liberer_equipe`, `liberer_tournoi`).
2. **Bugs dans le système de combat** : Certains personnages étaient trop puissants, ce qui déséquilibrait le jeu. Nous avons dû ajuster les formules de calcul des dégâts et d'esquive.
3. **Système de tours** : L'implémentation initiale du système de tours favorisait trop les personnages rapides. Nous avons créé un système de jauge d'attaque qui offre un meilleur équilibre.
4. **Animations** : L'implémentation des animations ASCII dans un terminal a posé des problèmes de synchronisation et d'affichage.
5. **Saisie utilisateur** : La validation des entrées utilisateur a nécessité une attention particulière pour éviter les crashes du programme.

6. Améliorations possibles

Si nous avions plus de temps, nous aurions aimé implémenter :

1. **Sauvegarde et chargement** : Permettre de sauvegarder la progression dans un tournoi
2. **Plus de personnages** : Ajouter davantage de personnages avec des compétences uniques
3. **Interface graphique** : Remplacer l'interface texte par une interface graphique simple
4. **Mode multijoueur** : Permettre à deux joueurs de s'affronter
5. **Système d'expérience** : Faire évoluer les personnages après les combats
6. **Effets de statut** : Ajouter des effets comme poison, paralysie, etc.
7. **Équilibrage des personnages** : Affiner les statistiques pour un meilleur équilibre

7. Conclusion

Ce projet nous a permis de mettre en pratique les concepts de programmation en C que nous avons appris cette année. Malgré quelques difficultés, nous avons réussi à créer un jeu fonctionnel et amusant.

La structuration du code en modules nous a aidés à mieux organiser notre travail et à collaborer efficacement. Les structures de données et les algorithmes que nous avons utilisés nous ont permis de mieux comprendre leur application pratique.

L'implémentation du système de jauge d'attaque a considérablement amélioré l'expérience de jeu en rendant les combats plus dynamiques et stratégiques. Nous sommes satisfaits du résultat final, même si nous avons identifié plusieurs points d'amélioration pour une version future.

