

Rapport de Projet – CYFighters

Membres du groupe

- ABTTA Yanis
- WAJIH Mansour
- SLIMANE Malak

Pourquoi avons-nous choisi ce projet ?

Nous avons choisi ce projet car il se rapproche des jeux que nous connaissons, notamment les RPG, ce qui le rend très motivant à développer. Le sujet offre aussi une grande marge de progression et de créativité, ce qui nous a permis de l'adapter à notre manière.

Organisation

Nous avons mis en place un serveur Discord dédié pour organiser efficacement le travail. Chaque membre pouvait y publier les avancements, les bugs rencontrés, les idées de design, ou encore les nouvelles versions du code. Des salons spécifiques ont été créés : maps, personnages, version-code, musique, idées, etc. Cela nous a permis de mieux répartir les tâches et d'éviter les doublons.

Nous avons organisé des réunions tous les trois jours pour faire le point sur l'avancée de chacun, et parfois tous les jours lorsqu'une deadline approchait ou qu'un blocage technique était rencontré.

Semaine 1 : Mise en place et premières réflexions

- Installation des environnements de travail sur les PC : WSL, SDL, GitHub, VSCode.
- Préparation des fichiers de base : .h et .c.
- Réflexion autour du gameplay, des personnages et des attaques.
- Premiers choix sur les effets de statut et les maps.
- Création d'une première interface : menu principal (Jouer, Options, Quitter).

Semaine 2 : Ajustements et organisation

- Finalisation des effets de statut et des maps.
- Implémentation du système de langues.
- Réorganisation du dépôt GitHub.
- Résolution de conflits de version.
- Amélioration de la structure des fichiers.

Semaine 3 : Implémentation du gameplay

- Début du codage du gameplay.
- Page de sélection des personnages.
- Gestion des musiques et bugs de RAM.
- Création de la page Histoire.

- Amélioration du système de sélection.
- Implémentation d'une IA facile.

Semaine 4 : Finalisation et optimisation

- Refonte du gameplay, des attaques et des structures de personnages.
- Organisation du code en modules : attaque, data, interface, son.
- Ajout de toutes les attaques et personnages.
- Finalisation de l'affichage, de l'équilibrage, des effets de statut, du mur Incassable.
- Page de confirmation, page options, crédits, sélection de musiques.
- Refonte de menu.c et ajout de toutes les musiques.

Problèmes rencontrés

- Redirection des attaques sur "Incassable" avec sa compétence "Mur vivant".
- Cibles attaquables après la mort d'un personnage.
- Mauvaise gestion des attaques multiples.
- Sélection de cible par l'IA lors d'une attaque.
- SDL : problèmes d'organisation des fichiers, segmentation fault, bugs d'affichage, erreurs de rendu.
- SDL_mixer : musiques parfois interrompues ou mal lancées.
- Fuite de mémoire détectée à cause de la police non libérée.

Solutions apportées

- Rediriger les attaques sur "Incassable" avec "Mur vivant" : récupération de l'ID, condition dans chaque attaque.
- Vérification de l'état des personnages avant l'attaque.
- Nouvelle gestion des cibles multiples via boucle spécifique.
- Ajout d'un algorithme de ciblage automatique pour l'IA.
- Tests réguliers, recherches sur forums et wikis.
- Utilisation correcte de SDL_FreeSurface, SDL_DestroyTexture.
- Debugging mémoire et optimisation de la gestion de la RAM.