

Rapport de développement

MU Davy, PEREIRA Tony, QUETIN Julien

SOMMAIRE :

Architecture du projet	3
Répartition des tâches entre trinômes	3
Difficultés rencontrées	3
Ce qui a été implanté	4
Ce qu'il reste à implémenter	4
Perspectives d'évolution du projet	4
À destination d'un développeur externe	5

Architecture du projet

Le projet se découpe en 5 packages. Le package "bag" contient toutes les classes en rapport avec les objets. Le package "pokemon" contient toutes les classes en rapport avec les pokémons, ainsi que les joueurs. La classe "data" contient toutes les classes en rapport avec les données du jeu ainsi que le pokédex et la liste des capacités. Le package "fight" contient les classes en rapport avec les combats, c'est-à-dire la ligue ainsi que le multijoueur. Enfin le package "main" contient le main.

Répartition des tâches entre trinômes

Davy : Partie II. Prérequis, Partie III. Chargement et Sauvegarde, Partie VI. Gestion des types.

Julien : Partie III. Chargement : Création de la fonction permettant de créer sa team ainsi que la sélection des capacités. Création de la classe PokemonTeam. Partie V. Ligue : aide à la réalisation. Réalisation du fichier stats afin d'avoir les stats pour chaque pokémon, cependant ce sont les stats au niveau 1.

Tony : Partie IV. Combats, Partie V. Ligue, Partie VII. Multijoueur, Partie VIII. Objets.

Sur la fin du projet, tout le monde travaillait ensemble afin d'essayer de régler les différents bug. Chaque membre a finalement touché un peu à toutes les parties.

Difficultés rencontrées

Mise en commun du code réalisé par chacun. Nous avons donc créé un gitHub afin que chacun puisse mettre en commun le code.

Les nombreux bug rencontrés :

Lors des combats, lorsque nous devions changer de pokémon, le programme nous renvoyait une erreur disant qu'avec le scanner il n'avait pas trouvé de ligne. L'erreur venait du fait que l'on avait fermé le scanner.

Lorsque tous les pokémons de notre équipe étaient morts, le programme nous demandait de choisir le prochain pokémon à invoquer alors que nous n'avions plus de pokémon. Le problème fut réglé avec l'ajout d'une condition.

Dans le multijoueur, lorsque le premier joueur avait joué, il pouvait jouer encore et encore ce qui fait que le joueur 2 ne jouait jamais. Le problème venait du fait que l'on entrait jamais dans la condition pour le joueur 2 à cause d'une mauvaise variable utilisée.

Lors de la réalisation de la partie V. Ligue, nous avons eu des difficultés à se faire enchaîner les combats entre les différents adversaires. Problème résolu avec l'implémentation d'une boucle while.

Difficultés pour la réalisation de la partie VI.

Difficultés pour la structure des objets, à savoir si il fallait faire une interface, etc...

Disponibilités dûe aux vacances de Noël.

Difficultés à comprendre le code réalisé par autrui.

Ce qui a été implanté

Dans ce projet nous avons implémenté :

- La création des types;
- La création du pokédex;
- La création de la liste des capacités, ainsi que les capacités;
- La création des pokémons;
- La création de la sauvegarde ainsi que du chargement du partie lors de la ligue.
- La création de son équipe;
- Le mode original des combats;
- La ligue par défaut;
- La ligue "made in DUT2";
- La gestion des types;
- Le multijoueur;
- Les objets.

Ce qu'il reste a implémenté

- L'état des pokémons;
- Le système de niveaux;
- Les évolutions;
- L'acquisition de capacités;
- Les boost sur les capacités des pokémons;
- Les modes 2v2 et 3v3.

Perspectives d'évolution du projet

Pour faire évoluer notre projet et le rendre meilleur et plus intuitif, une partie graphique serait la bienvenue. Cela permettrait une plus simple utilisation du programme. Si la partie

graphique ne peut-être réalisée, alors rajouter des switch pour la sélection des capacités à utiliser afin d'éviter une faute de frappe lors de la sélection de ces dernières.

Il y a aussi une grosse amélioration de la classe Fight car certaines variables utiles sont créées dans une méthode et donc inutilisables en dehors de cette méthode. Cette simplification pourrait permettre de faire des méthodes plus simples.

On pourrait aussi plus commenter notre code.

A destination d'un développeur externe