

LouLilou - Imac 2022



CRÉONS POKIMAC



MENU

START



SOMMAIRE



PRÉSENTATION DU JEU



MÉTHODES D'IMPLÉMENTATION



DIFFICULTÉES RENCONTRÉES



MÉTHODE DE TRAVAIL



PRÉSENTATION DU JEU



★ Pokimac est un jeu de combat avec un affichage graphique créé grâce à la librairie sdl.

Le jeu commence par un menu dans lequel nous pouvons choisir nos conditions de victoire et afficher les règles du jeu en cliquant sur les icônes avec la souris.



Lorsque nous lançons une partie nous devons d'abord choisir notre "PokiStarter".





PRÉSENTATION DU JEU



Une fois le jeu lancé, notre petit personnage apparaît sur la carte devant une maison qui fera aussi office de centre de soin.

Des Pokimacs sauvages sont éparpillés sur cette carte et lorsque notre personnage va à leur rencontre un combat se lance. Lors d'un combat nous avons le choix d'attaquer (avec une attaque spéciale ou non), de capturer, de fuir ou de changer de Pokimac.





PRÉSENTATION DU JEU



Chaque Pokimac possède un type qui caractérise ses forces et ses faiblesses vis-à-vis des autres types. Si un pokimac meurt, il disparaîtra pour tout le reste de la partie.



Nous perdons lorsque tous les pokimacs dans notre équipe sont morts. Nous gagnons lorsque nous avons battu le nombre de pokimacs que nous avons défini avant de commencer la partie.





MÉTHODES D'IMPLEMENTATION



LOGIQUES PARTICULIÈRES



- ➡ Les collisions sont gérées selon les classes des entités et rangées dans un vecteur qui les traite une par une. Elles dépendent de la taille des Sprites.
- ➡ Pour récupérer des informations sur les Pokimacs, nous lisons un fichier texte puis nous stockons les informations de ce fichier dans des vecteurs de classes.
- ➡ Nous avons implémenté des animations et VFX.
- ➡ Dans la maquette console nous avons utilisé les librairies fstream (stdio.h pour le code final) pour ouvrir les fichiers .txt, iomanip pour avoir un affichage plus contrôlé (setw et setfill) et ctype (tolower/toupper) pour prendre en compte les casses.



DIFFICULTÉES RENCONTRÉES



- ✦ La syntaxe et les paramètres des fonctions.
- ✦ La manipulation et le stockage des informations avec les pointeurs et les références.
- ✦ Installer la librairie SDL et réussir à compiler sur CodeBlocks.
- ✦ Trouver pourquoi ça ne compilait pas. (à cause de Framework)
- ✦ Se rendre compte que la classe VFX ne se lance pas deux fois d'affilée.
- ✦ Trouver une façon assez intuitive pour faire fonctionner notre programme sur la machine d'un utilisateur lambda.



MÉTHODE DE TRAVAIL















Lilou étant la plus à l'aise en code, lors de la phase d'idéation sur le déroulement d'une partie, c'est elle qui déterminait ce qui était faisable ou non. Elle a travaillé sur toute la partie affichage graphique du code avec la librairie SDL et elle a aussi géré les événements et les animations.



En parallèle, Lou a codé une bonne partie du coeur du programme en C++. Son code était en quelque sorte une maquette qui simulait le déroulement d'une partie avec un seul pokimac. Elle a aussi créé certains éléments graphiques que nous retrouverons dans le jeu.





MERCI D'AVOIR JOUÉ !

VOULEZ-VOUS FAIRE UNE AUTRE PARTIE ?

Oui

Non

