

# Programmation et Algorithmique

## Projet POKIMAC

### Déroulé du jeu

Pour lancer le jeu il faut taper « make » dans le terminal, puis ENTREE pour poursuivre le dialogue.

Après un dialogue avec le professeur Chen, le joueur se retrouve sur une map dans laquelle il peut se déplacer avec Z, S, Q et D (puis ENTREE). Il peut rester sur la route 1, aller à la plage ou en ville et rentrer dans les bâtiments. Il peut se rendre dans les hautes herbes pour rencontrer des Pokémon qui y apparaissent aléatoirement, il a le choix de les combattre avec ses Pokémon ou de les attraper avec des Pokéballs. Chaque Pokémon a une caractéristique Attaque et une caractéristique Défense, générées aléatoirement quand le Pokémon est obtenu, qui influencent les dégâts infligés. Une fois affaibli, un Pokémon est plus facile à attraper.

Le joueur doit gérer les PV de ses Pokémon, il peut les soigner en se rendant au centre Pokémon ou en utilisant des potions dans son inventaire. Le joueur a de l'argent pour acheter les potions et les Pokéballs dans la boutique Pokémon, même s'il débute avec 1000 Pokédollars, 15 Pokéballs et 10 potions. Pour gagner de l'argent, le joueur peut aller au casino et jouer à deux mini-jeux : une machine à sous et le Juste Prix.

Il n'y pas de fin précise. Le but : tous les attraper !

### Structure du programme

Axel : CLion, Windows

Wendy : Visual Studio Code, Linux



#### main.h (Axel et Wendy)

- librairies
- structures Dresseur, Poklmac
- prototypage de toutes les fonctions

#### main.cpp (Axel et Wendy)

- initialise le joueur (structure Dresseur). Pointeurs : les fonctions prennent en argument un Dresseur \*joueur (&joueur à l'appel), on ré-affecte ainsi joueur->equipe[0] pour modifier le premier Pokémon de l'équipe, joueur->argent pour modifier son montant en poche, etc.
- appel du dialogue du début
- jeu avec MAP

### **f\_dialogs.cpp : Dialogues (Wendy)**

dialogue du début de jeu et le choix du starter (essentiellement cout et cin/getchar())

### **f\_map.cpp : Gestion de la map (Axel et Wendy)**

- `drawmap()` = remplit tableau de caractères à 2 dimensions `char* map[TX][TY];`

Deux boucles imbriquées (for x et for y) remplissent les x colonnes d'une ligne y, on affiche la y-ème ligne une fois remplie

Joueur = un élément du tableau à 2 dimensions (l'élément `map[posY][posX]`)

`Typemap` = entier pour identifier dans quel environnement (route 1, plage, ville, bâtiment) on se trouve.

- `displaycolor()` = affichage en couleur de tous les éléments du tableau

- `mapname()` = affiche le nom de la map ou bâtiment

- `MAP()` = appelée dans le main, appelle `drawMap` dans une boucle for infinie, permet le déplacement du joueur dans le tableau. Permet aussi d'accéder à l'équipe (E), au Pokédex (P) et à l'inventaire (I) du joueur.

### **Des boucles if permettent, selon la position du joueur et typemap :**

- de changer d'environnement si l'on « touche » les bords correspondants d'un environnement ou si l'on passe la porte d'un des bâtiments

- de ne pas sortir de la map

- de lancer une interaction (dans le centre de soins, la boutique et le casino)

- d'éventuellement lancer un combat, si le joueur se trouve dans les hautes herbes (repérées par la fonction `dansHT`)

### **f\_interactions.cpp : Gestion des interactions (Wendy)**

- chaque interaction/dialogue dans les bâtiments = gérée par une fonction, appelée par une boucle if dans `MAP` quand le joueur se trouve à la position (`posX`, `posY`) correspondante

- fonction pour la machine à sous

- fonction pour le jeu « Le Juste Prix »

### **f\_menus.cpp : Equipe, Inventaire (Wendy)**

chaque menu permet de parcourir les champs dans la structure `Dresseur` joueur pour afficher les `PokI`mac de l'équipe, les objets possédés, l'argent en poche. On peut aussi soigner ses Pokémon (la fonction qui affiche l'inventaire prend `Dresseur *joueur` en argument pour réaffecter les PV des `PokI`mac (`joueur->equipe`)[i]).

### **f\_pokedex.cpp : Génération de tous les Pokémon (Axel)**

- fonction qui à un entier n renvoie le n-ième Pokémon du Pokédex, avec Attaque et Défense aléatoires

→ A l'origine on souhaitait que le Pokédex soit un tableau de 151 `PokI`mac que l'on aurait passé en argument des fonctions en ayant besoin, en accédant au n-ième élément du tableau (mais on avait des problèmes de pointeurs...)

### **f\_fight.cpp : Rencontrer des Pokémon sauvages, les combattre/capter (Axel)**

une fonction génère aléatoirement des Pokémon selon leur rareté et `typemap` (passé en argument). `Ecrancombat` gère tout le combat une fois qu'il est lancé, permettant d'attaquer le pokémon sauvage (4 choix d'attaque, et le Pokémon sauvage riposte en utilisant aléatoirement une de ses quatre attaques), le capturer (il va alors dans l'équipe du joueur), changer de pokémon attaquant, ou fuir. Une fonction calcule les dégâts subis en fonction de l'Attaque, de la Défense et de leurs diminutions durant le combat.

### **makefile : Compiler, Nettoyer, Executer (Axel et Wendy)**

- compilations avec g++

- édition des liens

- suppression de tous les fichiers .o (`rm *.o`)

- exécution (`./pokimac`).

## Quelques fonctionnalités récurrentes

<code>system("clear");</code>	Efface la console
<code>val = rand() %11;</code>	Génère un entier aléatoire
<code>string suite; suite = getchar();</code>	Pour passer à la suite en appuyant sur ENTREE
<code>cout &lt;&lt; "O/N : "; cin &gt;&gt; choix; if (choix=="0"    choix=="o") {</code>	Choix Oui / Non (ou chiffre)
<code>sleep(1);</code>	Figé l'affichage (pendant 1s ici)

## Quelques chiffres

**25**

Fonctions

**325**

Conditions « if »

**2676**

Lignes au total

**73 ko**

Poids du programme

## Pistes d'amélioration

- + de Pokémon dans le Pokédex et un système d'évolution (actuellement il n'y a que les Pokémon non évolués)
- un Pokédex listant tous les Pokémon attrapés
- + de Maps
- ne plus appuyer sur ENTREE après Z/S/Q/D ou E/I
- prévalences des types prises en compte dans le calcul des dégâts en combat
- affichage des types et des représentations des Pokémon en couleur

## Et comme tricher, c'est gagner ...

Essayez :

- lors du choix du starter, de rentrer un autre nombre que 1, 2 ou 3
- lors du choix du starter, de rentrer 151
- sur les trois différentes Maps, de taper « L » puis ENTREE
- lors du jeu de la machine à sous, de taper « Motherlode » au lieu de « M »

