

# Rapport : Projet Prog 2

Mathis DEGRYSE / Gaspard REGHEM

March 29, 2022

## 1 Fonctionnalité du Code

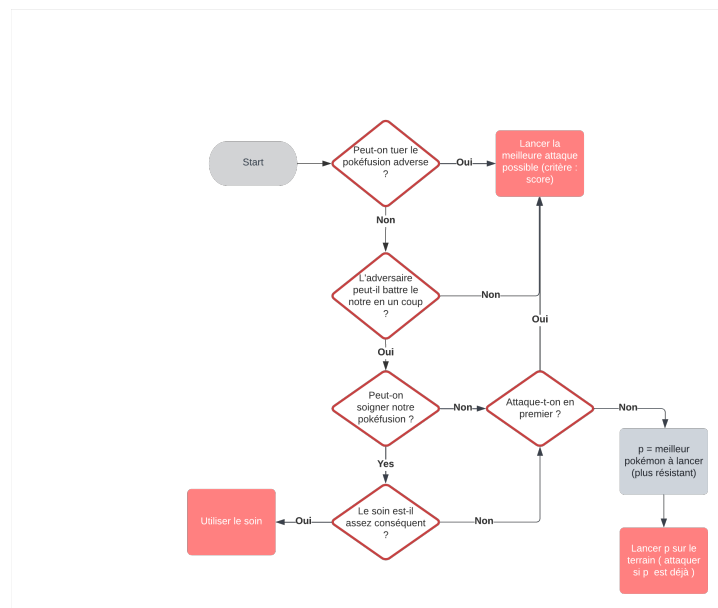
Au premier rendu, notre code permettait de réaliser un combat contre l'ordinateur. Durant ce combat tour par tour, le joueur avait le choix entre lancer une attaque et changer de Pokéfusion. L'ordinateur lançait toujours la première attaque de son Pokéfusion.

### 1.1 Types

Nous avons déjà implémenté le système de type pour le premier rendu. Nous avons repris les types des jeux Pokémon, il y en a donc 17 : Acier, Combat, Dragon, Eau, Electrik, Feu, Glace, Insecte, Normal, Plante, Poison, Psy, Roche, Sol, Spectre, Tenebres et Vol. Nous avons implémenté une classe `Ttype` dans le fichier `Ptype.scala` qui contient le nom du type, une chaîne de caractères qui permet de récupérer une image, et une fonction d'affinité. Les Pokéfusions et les attaques ont tous un et un seul type. Pour l'instant, les images sont des rectangles de couleur qui permettent de colorer les boutons afin de deviner le type des Pokéfusions et des attaques pendant la bataille. Les affinités sont celles des jeux Pokémon ce qui évite de devoir les réapprendre.

### 1.2 Intelligence Artificielle

Voici un schéma qui résume le fonctionnement de l'IA, qui est à priori assez simple. On a cherché à avoir une IA qui détermine son choix (non optimal) en  $O(1)$  sans arbre de Monte-Carlo ou n'importe quelle méthode similaire, car elles ne relèvent d'aucune stratégie déterminée en amont.



La partie la plus complexe de l'IA réside dans la détermination de "la meilleure attaque". En se mettant à la place du bot, on cherche l'attaque qui a le meilleur 'score'. Pour déterminer ce score, on cherche d'abord contre quel Pokéfusion il est le plus probable qu'on combatte, c'est à dire que le joueur peut en effet changer de pokémon. Pour déterminer cette probabilité, on se met à la place du joueur, et on cherche à déterminer son envie de lancer un autre Pokéfusion, pour cela, on calcule pour chacun des Pokéfusions du joueur une estimation de son temps de survie et le temps de survie de notre pokémon face à celui-ci. Le rapport des deux donne un score convenable (le pokémon sur le terrain possède un tour de survie supplémentaire).

Maintenant on obtient un tableau comme par exemple :

[1, 0.66, 0.3, 0.1, 0.4, 0.33]

Ces probabilités sont pondérées par les changements faits dans la partie par le joueur (s'il ne change que rarement de Pokéfusion spontanément, ces valeurs sont drastiquement diminuées).

Le score de l'attaque est alors la somme des scores de l'attaque contre le i-ème Pokéfusion multiplié par la probabilité que ce Pokéfusion soit lancé.

Pour le score d'une attaque contre un pokémon particulier, il s'agit de la somme de 7 sous-scores désignant :

- les dégâts
- les Buffs/Debuffs possibles
- les états

Dans toute cette procédure, une dizaine de scalaire permettent de modifier le comportement de l'IA dans son choix d'attaque, ou d'action.

- un scalaire par sous-score, permettant d'augmenter une caractéristique recherché dans les attaques utilisées
- un scalaire désignant l'indifférence face à la précision des attaques
- un scalaire désignant la marge de sécurité lors d'un soin (si on prend 20 dégats, on veut se soigner de 20 pv + une marge de sécurité)
- un pour l'importance du possible changement de pokémon adverse

### 1.3 Pokédex

Pour le Pokédex, nous avons décidé de ne pas trop le charger en information. Son implémentation se trouve dans le fichier *Pokédex.scala*. Le Pokédex se découpe en trois parties : les informations en haut, la liste des Pokéfusions au milieu et les boutons sont en bas. Pour trier les Pokéfusions dans le Pokédex, nous avons indiqué tous les Pokéfusions grâce au tableau `liste_pokemon`. Les informations affichées ne sont pas les mêmes si le joueur a ou non rencontré ce Pokéfusion (c'est l'intérêt du tableau `encountered`). Si le joueur ne l'a jamais rencontré, le Pokédex ne donne aucune indication sur le nom, le type ou la description du Pokéfusion. Toutefois, comme dans Pokémon, on peut voir l'ombre du Pokéfusion. Sinon, le joueur peut alors voir le nom, le type et une courte description du Pokéfusion. Pour afficher les descriptions, une fonction permettant d'écrire une longue chaîne de caractères sur plusieurs lignes a été écrite dans le fichier *general\_functions.scala*. En dessous des informations, on peut voir la liste des Pokéfusions précédents ou suivants en fonction du sens dans lequel on parcourt le Pokédex. Le Pokéfusion actuellement affiché est surligné en cyan. Il est possible d'utiliser la molette pour parcourir le Pokédex. Les boutons **Suivant** et **Précédent** permettent de naviguer à travers le Pokédex et le bouton **Retour** renvoie au menu principal du monde ouvert. En effet, le Pokédex est accessible depuis le monde ouvert.

## 1.4 Monde Ouvert

Nous avons construit notre monde ouvert à base de zones pour éviter à devoir gérer une caméra mobile. Le monde ouvert est donc composé d'un carré de 3 sur 3 zones. Chaque zone est quadrillée en 15x10 afin de se repérer. La zone est défini par une matrice de revêtement et par une matrice d'obstacles. Les classes de revêtement sont le vide, le sol, et l'herbe. Quant aux obstacles, ils peuvent faire partie de nombreuses classes : vide, mur, lac, buisson, porte, panneau, personnage et objet. Tous les tableaux doivent être remplis à la main, cela est fait dans le fichier *Area.scala*. L'affichage des zones se fait grâce à une image, ensuite on redessine dessus certains des obstacles comme les personnages ce qui nous donne plus de liberté. Pour se déplacer, il faut utiliser les touches Z/Q/S/D et la touche E permet d'interagir avec certains obstacles (la gestion des inputs se fait dans *touche.scala*). Si le joueur possède l'objet chaussure de running de la Kulte, il peut aussi courir en maintenant shift appuyé pendant qu'il se déplace. On peut par exemple interagir avec un personnage pour lancer un combat ou interagir avec un lac pour pêcher. Pour passer d'une zone à l'autre, il faut interagir avec la bordure de la zone. Quand on change de zone, le personnage se déplace comme si on faisait un pas dans la direction de la nouvelle zone. Si jamais un obstacle empêche de changer de zone, alors le déplacement n'a pas lieu. Lorsqu'on se trouve dans le monde ouvert, un menu est affiché dans la partie basse de la fenêtre. Il permet de lancer le Pokédex, d'accéder aux Pokéfusions de l'équipe, d'accéder au sac, et de faire une sauvegarde. La gestion de l'équipe et la sauvegarde ne sont pas encore implémentées mais elles seront dans le troisième rendu.

## 1.5 Objets

On peut séparer les objets du jeu en quatre catégories : les soins, les soins de statut, les résurrections, les repousses et les autres. Les soins sont des potions de différentes puissances (20/50/200 PV). Les soins de statut d'enlever les statuts particuliers aux Pokéfusions. Il en existe un pour chaque statut possible (**Awakening** pour **Sommeil** par exemple). Il existe aussi un objet **Full Heal** qui soigne tous les états possibles. Les résurrections, **Revive** et **Max Revive**, permettent de ramener un Pokéfusion KO à la vie avec la moitié ou la totalité de ses PV. Les repousses permettent de se prémunir des combats sauvages pour un certain nombre de pas. Pour l'instant, il n'en existe que deux objets à part : la classe **Fishing\_Rod** qui permet de pêcher et la classe **Chaussure** qui permet de se déplacer plus vite. Il existe une variable **is\_usable** qui oblige le joueur à utiliser les objets de façon utile. Ainsi, il est impossible de soigner un Pokéfusion KO ou un Pokéfusion qui a toute sa vie. De même, le joueur ne peut pas utiliser un **Awakening** sur un Pokéfusion qui n'a pas le statut **Sommeil**. De plus, en appuyant sur la touche A, le joueur peut activer une aide qui va lui indiquer les effets des objets quand il survole leur bouton (cette aide existe dans tous les menus).

## 1.6 Bonus

En plus des objectifs fixés par notre chargé de TD préféré, nous avons implémenté quelques fonctionnalités supplémentaires.

### 1.6.1 Pokéfusion Sauvage

Quand le joueur se déplace sur une case du monde ouvert dont le revêtement est **Herbe**, il a 1 chance sur 10 de déclencher un combat avec un Pokéfusion sauvage. Pour cela, nous avons créé une classe de personnage **Nature** et, pour chaque combat sauvage, on crée un nouveau personnage de cette classe et on lui attribue un seul Pokéfusion. L'espèce de Pokéfusion choisi dépend de la zone dans laquelle le joueur se trouve, même si, pour l'instant, il ne peut s'agir que d'un **Salatard**. Le niveau du Pokéfusion sauvage va dépendre de la moyenne des niveaux des Pokéfusions du joueur.

### 1.6.2 Pêche

Comme mentionné plutôt dans ce rapport, il est possible de pêcher en interagissant avec une case qui a un obstacle de classe `Lac` si le joueur possède un objet de classe `Fishing_Rod`. On peut arrêter de pêcher en appuyant à nouveau sur E. Lorsqu'on pêche, à chaque tour de la boucle while principale (environ 0.1 seconde), vous avez 1 chance sur 10 de déclencher un combat sauvage. La seule différence est que la classe de l'adversaire n'est pas `Nature` mais `Lac_opp`. La distribution des espèces de Pokéfusions dans le lac n'est pas la même que dans les herbes.

### 1.6.3 Fuite

Il est aussi possible de fuir le combat. Contrairement à Pokémon, la fuite réussit tout le temps et il est possible de fuir les combats non-sauvages.

### 1.6.4 Animation

Le personnage est animé lors de ses déplacements ce qui rend le jeu beaucoup plus fluide. Pour animer le personnage, on lance un thread qui affiche l'animation dès que le joueur se déplace. Pour éviter de nombreux problèmes, il est crucial que le thread d'évènement ne boucle pas pendant la durée de l'animation car le thread qui gère les appels à la fonction `repaint` est le même.

## 2 Organisation du Code

Une flèche du sommet *a* vers le sommet *b* indique un appel du fichier *b.scala* dans le fichier *a.scala*.

