Rapport du TP3
IA01- Automne 2015
Pokemon : le simulateur de combat

Marie Simatic & Agathe Guillemot

2 janvier 2016

Table des matières

1 Problématique

1.1 Un Simulateur de Combat

Les pokémons, de l'abbréviation de poket monsters, sont, dans les jeux vidéos, des petits monstres que l'on fait s'affronter lors de combat. Ces derniers sont régis par de nombreuses statistiques complexes. Pour le dresseur de pokémon débutant, il n'est jamais facile de savoir l'issu d'un combat.
Ainsi, nous avons créé un système expert permettant de simuler un combat pokémon entre deux pokémons.

2 Formalisation du problème

2.1 Besoins

Notre système est d'ordre 0+, par conséquent notre base de fait sera une liste associative (fait, valeur), et les prémisses devront parfois répondre à des équations complexes et s'écriveront alors comme une liste d'expression lisp à interpréter selon les valeurs de la base de faits.

Le système aura besoin pour fonctionner d'un set d'information simple venant de l'utilisateur à savoir les noms des pokémons se combattant entre eux et leur niveaux respectifs.

2.2 Des rappels sur les pokémons

Chaque pokémon est issue d'une race particulière et possède :
- des statistiques (nombre de point de vie, puissance d'attaque, puissance de défense, une vitesse)
- un type
- un niveau

L'amateur de pokémon avisé notera que nous avons choisi, dans notre modèle, de simplifier le système de pokémon original. En effet, un pokémon possède en plus une puissance d'attaque spéciale et une puissance de défense spéciale, pour les attaques non physiques que les pokémons se porteraient. Néanomins, afin de ne pas alourdir, nous avons choisi de les écarter.
De même, nous avons décidé de ne pas traiter le système d'EV. Les EV sont des caractéristiques spécifiques pour chaque pokémon qui font que deux salamèches de niveau équivalent n'auront pas forcément les mêmes statistiques.
Ces deux notions ayant peu d'influence sur le déroulement d'un combat, les résultats obtenus par notre système expert n'en patissent néanmoins pas.

2.2.1 Calcul des statistiques

Les statistiques d'un pokémon dépendent de sa race, possédant des statistiques de base, et de son niveau.
Le nombre de point de vie s'obtient avec la formule suivante : $PV = 2 * (StatistiqueDeBase * Niveau)/100 + (Niveau + 10)$
Le reste des statistiques s'obtiennent de façon à peu près similaire :$Stat = 2*(StatistiqueDeBase* Niveau)/100 + 10$
On notera qu'on ne gardera, à chaque fois, que la partie entière des statistiques

Source : http ://www.pokepedia.fr/index.php/Calcul_des_statistiques

2.2.2 Types et modificateur de dégats

Chaque race de pokémon possède un type qui sont des sortes de familles. Ainsi, salamèche et pyroli, par exemple, sont des pokémons de type feu.
Chaque type a des résistances et des faiblesses. Ainsi, l'eau est efficace contre le feu (on dira que l'eau est une faiblesse du feu) et au contraire le feu n'est pas très efficace contre l'eau (on dira que le feu est une résistance de l'eau).
Ces faiblesses et résistances ont un impact sur les combats. Ainsi, un pokémon de type feu fera deux fois plus de dégats à un pokémon de type plante, car le feu est une faiblesse du type plante. De façon analogue, un pokémon de eau ne subira que la moitié des dégats d'une attaque d'un pokémon de type feu, car le feu est une résistance du type eau.
Du type des deux pokémons combattants, il est donc aisé d'obtenir le modificateur de dégat de ces derniers.

Source : http ://www.pokepedia.fr/index.php/Table_des_types

2.2.3 Calcul des dégats, déroulement d'un combat, gagnant

Chaque pokémon attaquera son adversaire à tour de rôle. Le pokémon attaquant en premier est celui dont la statistique de vitesse est la plus élevée. Le combat s'arrête quand l'un des combattants n'a plus de PVs.
Les dégats reçus à chaque tour répondent à la formule suivante :
$PV_{perdus} = (Attaque_{attaquant}/Défenseur_{défenseur} + 2) * ModificateurDeDégats_{attaquant}$

De ce calcul, on peut aisément déterminer en combien de tours un pokémon mettra son adversaire K.O, au moyen d'une simple division.
Le gagnant du combat est tout simplement celui mettant son adversaire en le moins de tours. Dans le cas où un pokémon commence le combat, ce dernier gagnera un bonus de un tour lors du calcul.

Source : http ://www.pokepedia.fr/Calcul_des_dégats

2.3 Représentation en Lisp de la base de faits : base_faits.lisp

Un fait dans la base de fait est représenté de la façon suivante : (identifiant (nom valeur))

- L'identifiant : unique pour chaque fait. Généré à l'aide de la fonction lisp gentemp(). Pour plus de lisibilité, l'identifiant n'est pas qu'un nombre, mais pourra avoir la forme de type_factX ou bien faitX. Ainsi, la lecture de l'identifiant permet de catégoriser le fait. La lecture de la base de faits en est facilitée.
- Le nom : c'est avec le nom du fait que la valeur du fait sera utilisée, via la fonction valeur_fait_fromBase(). On notera qu'un nom de fait aura forcément la forme d'un symbole Lisp. Un fait ne peut donc pas se nommer 3 c̈harëtc.
- La valeur : si le fait a plusieurs valeurs (comme pour la liste des faiblesses), la valeur d'un fait aura alors la forme (list val1 val2 ...). Le mot clé lisp list est nécessaire pour pouvoir interpréter l'ensemble de valeurs comme une liste dans les prémisses de la base de règle.

Si la base de fait sera initialisée juste avant l'utilisation du moteur d'inférence dans le programme, elle contient de base l'ensemble des forces et des faiblesses de chaque type.

2.4 Représentation en Lisp de la base de règles : base_regle.lisp

Une règle dans la base de règle est représentée de la façon suivante : (identifiant (liste des prémisses) (liste des conclusions) texte). BLAHBLAHBLAH

3 Programme en Lisp

3.1 Moteur d'inférence avec chaînage avant

Notre chaînage avant applique la stratégie suivante : choix de la première règle applicable, exécution des ses conclusions et ainsi de suite jusqu'à ce qu'il n'y ai plus de règles applicables. Chaque règle est utilisée au maximum une seule fois.

`Algo`

3.2 fonctions de services

4 Utilisation