UNIVERSITE CATHOLIQUE DE LOUVAIN

RAPPORT LFSAB1402 : POKEMON GOZ

Bastin Julien – Detry Damien
81031400 91451400

Année académique 2016-2017

1) Structure du programme

a. Explications des différentes fonctions et sous fonctions

Fonction MyFunction

Dans cette fonction principale, nous retrouvons 8 sous fonctions qui sont les suivantes :

- *ValueToFloat* ➔ Fonction qui transforme une <value> en <float>. Si il y a des opérations à effecuer dans la <value> passée en argument, la fonction va calculer le résultat de manière récursive et renvoyer le résultat final en <float>.

- *FormulaToFloat* ➔ Fonction qui transforme une <formula> en <float>. Si il y a des opérations à effecuer dans la <formula> passée en argument, la fonction va calculer le résultat de manière récursive et renvoyer le résultat final en <float>.

- *AuxR* ➔ Fonction qui prend un <RealUniverseItem> et ses coordonnées de base en paramètre. Elle effectue de manière récursive des opérations sur les coordonnées en fonction de la transformation lue et renvoie au final une liste de fonction qui place les items sur la carte.

  Remarque : Nous nous somme rendu compte trop tard que nous ne respections pas la priorité des opérations. *Exemple:* si nous avons *scale(rx:2.0 ry:3.0 1:[translate(dx:20.0 dy:30.0 1:[primitive(kind:road)])])* la fonction va d'abord effectuer le *scale* et ensuite le *translate* sur les coordonnées de base.

- *Aux P* ➔ Focntion qui prends un <PokeUniversePOI> et ses coordonnées de bases en paramètre. Elle effectue de manière récursive des opérations sur les coordonnées en fonction de la transformation lue et renvoie au final une liste de fonction qui place les items sur la carte.

- *DoListR* ➔ Fonction qui parcours un élément de la liste Map.ru et qui renvoie la liste des fonctions générées par cet élément.

- *DoListP* ➔ Fonction qui parcours un élément de la liste Map.pu et qui renvoie la liste des fonctions générées par cet élément.

- *DoFinalListR* ➔ Fonction qui parcours la liste des éléments de Map.ru et crée une liste des fonctions à renvoyer

- *DoFinalListP* ➔ Fonction qui parcours la liste des éléments de Map.pu et crée une liste des fonction à renvoyer

Fonction CheckMap

La fonction CheckMap est la fonction qui va globalement vérifier que la map est bien définie avec les bons types et valeurs pour les champs des records. Dans cette seconde fonction, nous retrouvons 5 sous fonctions :

- *CheckP* ➔ Fonction qui teste la validité des éléments du PokeUniverse. Elle parcours la liste des PokeItem, si ceux-ci correspondent au types définit, la fonction renverra true.

- *CheckR* ➔ Fonction qui teste la validité des éléments du RealUniverse. Elle parcours la liste des RealItem, si ceux-ci correspondent au types définit, la fonction renverra true.

- *AuxR* ➔ Fonction qui parcoure récursivement le éléments de la liste RealUniverse et qui créer une liste de fonction qui place chaque éléments sur la map en fonction de ses coordonées pendant un certain laps de temps.

- *AuxP* ➔ Fonction qui parcoure récursivement le éléments de la liste RealUniverse et qui créer une liste de fonction qui place chaque éléments sur la map en fonction de ses coordonées.

- *CheckTrueOrFalse* ➔ Fonction qui parcoure récursivement le éléments de la liste PokeUniverse et qui créer une liste de fonction qui place chaque éléments sur la map en fonction de ses coordonées pendant un certain laps de temps.

b. Décisions de conceptions

Tout le programme se base sur l'idée que nous séparons le RealUniverse et le PokeUniverse pour ensuite exécuter les différentes fonctions desssus. Nous avons écrit le programme de cette manière car les différents type de champs ne sont pas les même dans les deux cas. Il est également plus simple de créer des fonctions spécifiques pour chacuns de ces deux univers.
Nous avons donc créer une fonction spécifique à chaque univers. Peut être aurions-nous pu trouver une solution pour rassembler ces deux univers mais nous n'avons pas trouvé cette solution.
Toutes nos fonctions ont été écrites récursivement pour une optimisation complète du programme.

c. Complexités des différentes fonctions

La plupart des fonction que nous avons définies se font en une complexité de $\theta(n)$ car il s'agit de fonction qui parcoure une liste de records (n étant la taille de la liste).
Les fonctions de cette complexité étant : *Append, Flatten, ValueToFloat, FormulaToFloat.*
Pour les autres fonctions, il s'agit également de parcours de liste mais cette fois-ci, en utilisant d'autre fonctions à chaque appel récursif. C'est pourquoi les complexités de ces autres fonctions sont en $\theta(n^2)$. Les fonctions concernées par cette complexité étant : *AuxR, AuxP, DoListR, DoListP, DoFinalListR, DoFinalListP*.

Nous pouvons donc conclure que les deux fonctions principales (*MyFunction* et *CheckMap*) s'exécutent avec une complexité de $\theta(n^3)$ où n est la taille de la map.

2) Extensions choisies

Pour ce projet, nous avons décidé d'implémenter toutes les extensions.

a. ExtendedFormula

Cette extension a été implémentée dans la fonction *ValueToFloat* pour les valeurs du RealUniverse et dans la fonction *FormulaToFloat* pour les formules du PokeUniverse.
Il nous a simplement fallu rajouter les différents cas dans le *case* et de les gérer de la même manière que les cas de bases.

b. IfThenElse

L'implémentation des "If Then Else" se trouve dans la fonction *FormulaToFloat*.

c. TimeWindow

Cette fonctionnalité permet aux items auquels un *spawn* est attribué de n'apparaitre sur la map qu'entre les temps *tmin* et *tmax*. Nous avons aussi fait en sorte de supporter les *spawn imbriqués*, s'il y a un *spawn* dans un *spawn* alors c'est l'intersection des *time* qui est prise en compte. Nous gérons aussi le déplacement à l'intérieure d'un spawn : si nous avons *translate(dx:X0 dy:Y0 1:[*spawn*(tmin:A tmax:B 1:[translate(dx:X1 dy:Y1 1:[primitive(kind:pokemon)]])])])* alors un pokemon se déplacera du point (X0,Y0) au point (X1,Y1) pendant un temps [A B].

d. CheckMapEasy et CheckMapComplete

Ces deux extensions se trouvent dans la deuxième grande fonction du programme, à savoir *CheckMap*.
Pour chacune des extensions créées dans *MyFunction*, il y a une fonction qui vérifie la validité de celles-ci dans *CheckMap.* Ces fonctions de vérifications se font toutes de manière récursives.