POKEMON GOZ

Dans le cadre du cours d'informatique enseigné par
Peter Van Roy

mozart

MISE EN SITUATION

Dans le cadre du cours d'informatique, nous avons été menés à travailler en binômes sur un programme informatique visant à améliorer la maîtrise du langage de programmation « Mozart Oz ». Ce rapport a pour but de décrire le déroulement du projet.

65441500 LAMOTTE Pierre
80721400 MEERTS Martin
LFSAB 1402

UCL Université catholique de Louvain

Table des matières

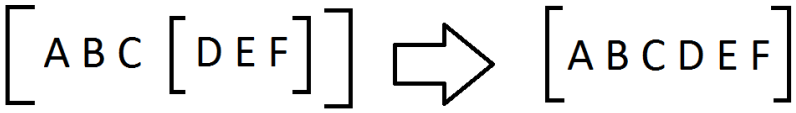
1. Introduction

Ce rapport a pour but de décrire l'entièreté du déroulement du projet, le code ainsi que les diverses extensions ajoutées. Vous trouverez le code complet en annexe.

Il nous a été donné pour instruction de créer une carte virtuelle en 2D constituée de divers éléments réels (points d'eau, bâtiments et routes) ainsi que d'éléments issus de l'univers de Pokémon Go (arènes, pokéstops et pokémons). L'objectif étant de déplacer des éléments sur la carte, il nous a fallu passer par plusieurs étapes d'apprentissage permettant de structurer au mieux le code.

2. Construction du code

Pour obtenir un code efficace, nous avons du définir certaines fonctions complémentaires telles que FlattenList qui permet de regrouper une liste d'éléments avec une sous-liste, pour n'en faire qu'une seule liste.



La fonction Transform prend des points, le temps et les formules Cos et Sin en argument. Elle retourne les points d'une primitive après y avoir appliqué une liste de modifications ordonnées (translate, rotate et scale).

RealUn crée une liste ordonnée d'opérations à effectuer sur les points d'une primitive. Une fois que cette liste est établie, il faut appliquer Transform qui retournera les points modifiés. On peut ensuite créer la primitive là où nous le souhaitons. Il faut ensuite la stocker sous forme d'un realitem, dans une liste définitive. La difficulté dans cette fonction, fut d'établir une liste ordonnée d'opérations. Il n'a pas été aisé de comprendre comment parcourir la liste du realuniverse. Mais après plusieurs tentatives, nous sommes parvenus à retourner une liste correcte.
PokeUn fonctionne de la même manière que RealUn, mais utilise le temps dans ses opérations. Seule l'opération translate est utilisée ici. La création de cette fonction n'a pas posé vraiment de problème, une fois que nous avions déjà RealUn
MyFunction regroupe PokeUn et RealUn. Elle retourne une liste contenant les deux listes de PokeUn et RealUn.

Nous avons décidé de rester dans la programmation déclarative. Nous aurions pu utiliser des cellules, mais nous nous sommes contentés des fonctions récursives.
Au niveau des extensions. Nous avons eu un peu de mal à comprendre leur définition. Mais nous avons définis les principales telles que plus, mult,'div', minus, cos, sin, tan, if-then-else, et exp, log, neg. Pour ce qui est de check map, nous n'avons pas eu le temps de la définir (avec ou sans extension)

Le gros problème est que nous n'avons pas réussi à rendre notre programme Depth-First. De ce fait, notre code n'est pas accepté par Inginious.
Au niveau des complexités temporelles et spatiales des fonctions, nous n'avons également pas eu le temps de les étudier correctement et de pouvoir les donner.

3. Conclusion

Ce projet fut vraiment enrichissant. Nous sommes déçus d'avoir perdu du temps dans la recherche d'erreurs , ou de ne pas avoir eu plus de temps pour améliorer notre programme. Nous aurions pu l'améliorer en définissant d'autres extensions comme checkmap, le spawn, etc, … Nous aurions pu aussi améliorer la syntaxe afin d'obtenir un temps de calcul plus court.