

IA41 – Teeko

Rapport de projet

Barbara SCHIAVI

Paul-Émile BRETEGNIER

Printemps 2015

Introduction

Dans l'UV IA41 nous sommes amenés à réaliser un projet de programmation en Prolog afin d'utiliser les connaissances acquises dans l'unité de valeur ce semestre. Nous avons choisi le jeu de plateau Teeko dont nous détaillerons les règles par la suite.

Dans ce rapport, nous regrouperons notre analyse du problème, la solution que nous y apportons, ainsi que l'ensemble des difficultés rencontrées, sans oublier les parties clefs du code.

Le jeu Teeko est un jeu pour deux joueurs, sur un plateau carré de 25 cases. Le jeu se déroule en une ou deux phases successives. L'objectif de chaque joueur est d'aligner successivement ses 4 pions sur une ligne, une colonne ou une diagonale ou de faire un carré (2*2). Lors de la première phase, chaque joueur dépose chacun son tour un de ces pions. Si aucun vainqueur n'émerge après que chacun d'entre eux ait posé ses 4 pions la seconde phase commence. À ce moment là, toujours à tour de rôle, les joueurs doivent déplacer leurs pions afin de réaliser une des conditions de victoire. Un pion peut uniquement être déplacé sur une case vide adjacente à sa position de départ au début du tour.

Sommaire

Introduction.....	2
1 Représentation des données.....	4
1.1 Plateau de jeu.....	4
1.2 Joueurs et pions.....	4
1.3 Conditions de victoire.....	4

1 Représentation des données

1.1 Plateau de jeu

Avant de commencer à réfléchir aux problèmes d'IA, nous avons cherché une solution afin de représenter le plateau. Petit à petit deux solutions ont émergé : la première, l'utilisation de deux coordonnées, une verticale et l'autre horizontale. La seconde, une liste successive de l'ensemble des positions, dans le cas présent, de 1 à 25. Chacune de ces solutions présentait des avantages et des inconvénients.

La mise en place d'un plateau sur deux axes ressemble à un jeu de plateau tel que les échecs, et semble une approche plus intuitive, en revanche sa mise en place est relativement lourde comparée à une succession de liste de 25 positions, en effet, il faut recueillir chacune des coordonnées à chaque traitement.

À l'inverse, en utilisant une liste de positions, le recueil des coordonnées est plus léger : on diminue ainsi le nombre de fonctions, et augmente la lisibilité du code.

Nous avons donc choisi de représenter le plateau comme une liste de 25 cases.

1.2 Joueurs et pions

Chaque joueur possède une liste contenant la position des 4 pions. Les pions sont une liste et possède dès le départ un nom (B1, B2, etc.).

1.3 Conditions de victoire

Afin de savoir si un joueur a gagné, nous devons trouver une façon de représenter l'ensemble des solutions gagnantes, ou bien vérifier si les 4 pions du joueur sont alignés ou en carré.

Le plus simple à nos yeux, était de renseigner l'ensemble des combinaisons gagnantes : lignes, colonnes, diagonales et carré.

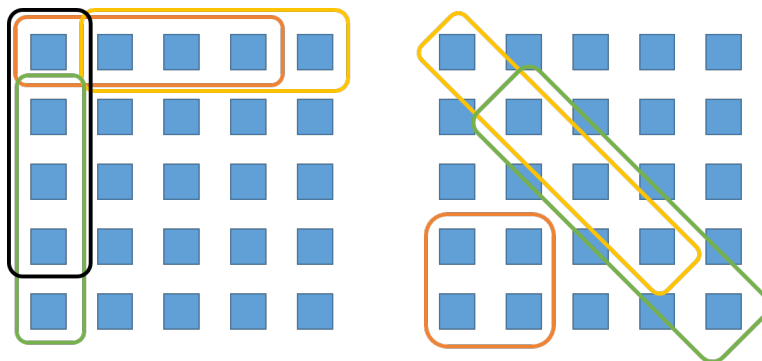


Illustration 1: Exemple de solutions gagnantes

un rappel de l'énoncé du problème,
votre spécification (formalisation) du problème,
l'analyse du problème,
la méthode proposée, avec en annexe le listing du programme commenté
la description détaillée d'une ou de plusieurs situations traitées par votre programme,
les résultats obtenus par votre programme sur ces situations,
les difficultés éventuellement rencontrées,
les améliorations possibles (méthodes de résolution) et les perspectives d'ouverture
possibles du sujet traité