1 Règles du jeu

Les règles du jeu des 7 couleurs sont :

— Un tableau d'une certaine taille est rempli de 7 couleurs aléatoirement.

— La case en bas à gauche (resp : en haut à droite) est de la couleur v (resp : ^) du joueur 0 (resp : 1).

— Chaque tour le joueur 0 (resp : 1) choisit une couleur parmi les 7 couleurs. Toute les cases de la couleur choisie qui sont juxtaposées à une case du joueur 0 (resp : 1) directement ou indirectement prennent la couleur du joueur 0 (resp : 1).

— Le jeu termine quand un joueur a rempli la majorité du tableau : il est le gagnant.

Nous allons d'abord aborder la réalisation du jeu en lui-même (et des choix de l'implémentation) puis parler de la réalisation de plusieurs IA pour ce même jeu.

2 Voir le monde en 7 couleurs

Le langage imposé fut le C. Il permet un contrôle rigoureux de la mémoire.

2.1 Remplir le monde

Le choix le plus intuitif était d'utiliser un tableau pour représenter le plateu de jeu. C'est ce que nous avons fait. L'autre choix cohérent et possible consistait à définir les cases en tant qu'objet et les voisins des cases mais le C n'est pas pratique pour faire de la programmation orientée objet. Nous avons fait le choix de représenter en interne les couleurs par les nombre de 2 à 9, et les joueurs 0 et 1 par le nombres 0 et le nombre 1. Des fonctions de traduction permettent ensuite d'afficher les couleurs correspondantes pour l'utilisateur.

La fonction *initgame()* permet d'initialiser le plateau de jeu. On remplit le tableau de chiffres de 2 à 9 aléatoirement, puis la case en bas à gauche (resp : en haut à droite) de la couleur du joueur 0 (resp : joueur 1). La fonction est en $\mathcal{O}(n)$ (on notera à partir de maintenant n le nombre de case du tableau)

2.2 Jouer un coup

Nous avons réaliser la méthode proposée pour jouer un coup. Elle consiste à parcourir le tableau de jeu et à trouver toute les cases de la couleur choisie par le joueur actuel qui sont adjacentes à l'une de ses cases. Si on a mis à jour au moins une case au cours de ce parcours, on effectue un nouveau parcours. On s'arrête lorqu'on à effectué un parcours complet sans changement. Le moyen le plus simple pour tester la correction de cette méthode est de l'appliquer quelques fois sur le tableau de jeu en affichant celui-ci et de vérifier que tout se passe comme prévu.

Dans le pire des cas, on effectuera $n-2$ parcours (si l'on modifie exactement une case à chaque parcours). Comme chaque parcours vérife les n cases du tableau, on a une complexité dans le pire des cas de $\mathcal{O}(n^2)$.

La plupart du temps on effetuera que quelques parcours (car il faut une grande zone de même couleur pour effectuer de nombreux parcours, ce qui est statistiquement peu probable), on a donc une complexité moyenne de $\mathcal{O}(n)$. Ce pendant, il est possible de faire mieux

2.3 Jouer un coup : meilleure alternative