

Projet 2011 – Think Ahead

Introduction

Le projet de cette année concerne la modélisation du jeu Think Ahead qui est un jeu basé sur une grille rectangulaire ayant le même nombre de lignes que de colonnes ; la taille de la grille pouvant être variables (notons m le nombre de lignes et de colonnes). Chaque case de la grille contient un nombre correspondant à sa valeur. A tour de rôle, chaque joueur choisi une case de la grille de façon alternative. Chaque case choisie rapporte au joueur autant de points que la valeur de cette case et elle ne pourra plus être choisi dans le reste de la partie. La particularité de Think Ahead est que l'on ne peut pas choisir une case n'importe où dans la grille, ce qui serait trop facile et enlèverait tout intérêt au jeu. L'originalité du jeu est que la case choisie ne peut être que dans une ligne ou une colonne particulière, la méthode permettant de savoir quelle ligne ou colonne est la suivante :

Pour commencer une ligne (ou colonne) est choisi aléatoirement (valeur¹ entre 1 et m). Le premier joueur joue (choisi une case et donc sa valeur) dans cette ligne (ou colonne). Le joueur suivant doit jouer dans la colonne (respectivement ligne) contenant la case précédemment choisi. Les joueurs jouent donc alternativement dans une ligne, puis dans une colonne, la case choisie servant de pivot. Par exemple dans la grille ci-dessous, si un joueur ($j1$) joue dans la ligne n°3 et choisi la case contenant la valeur 20, alors le joueur suivant ($j2$) devra jouer dans la colonne n°1.

1	5	9	3	20
10	9	4	8	8
20	0	8	3	6
4	7	5	0	15
2	9	15	10	6
1	0	0	2	7

Et si $j2$ choisi la case contenant 10, $j1$ devra maintenant choisir un nombre dans la ligne n°2.

1	5	9	3	20
10	9	4	8	8
	0	8	3	6
4	7	5	0	15
2	9	15	10	6
1	0	0	2	7

A ce niveau les joueurs $j1$ et $j2$ ont respectivement 20 et 10 points.

1	5	9	3	20
	9	4	8	8
	0	8	3	6
4	7	5	0	15
2	9	15	10	6
1	0	0	2	7

On comprend maintenant le nom du jeu, car le joueur doit choisir une case lui apportant un maximum de points tout en obligeant l'autre joueur à devoir choisir des cases lui apportant peu de points ; il faut donc anticiper les coups (penser en avant ou think ahead).

¹ Elles sont numérotées de 1 à m pour le joueur, mais pour le programme l'indice est compris entre 0 et $m-1$.

Point particulier de règle, si un joueur sélectionne la dernière case de la ligne (colonne) dans laquelle l'autre joueur doit jouer, alors le joueur suivant ne peut pas jouer et la partie est finie, même s'il reste encore des cases non sélectionnées.

Contrainte de modélisation

Le programme Java devra être décomposé en 3 packages : *modele*, *modeTexte* et *modeGraphique*. Le package *modele* correspondra au modèle mémoire et contiendra toutes les classes permettant la gestion des règles et des données du jeu. Le package *modeTexte* permettra de tester rapidement en mode texte le bon fonctionnement du modèle mémoire. Le package *modeGraphique* utilisera *Jtuto* (ou *Swing*, mais *Jtuto* est fortement conseillé par mesure de simplicité) pour permettre de jouer à Think Ahead en mode graphique.

Dans un premier temps, vous devrez créer le modèle mémoire et mettre en évidence son bon fonctionnement à partir des classes du mode texte. Lorsque cette première phase sera validée, vous pourrez passer à la réalisation du mode graphique.

Le modèle mémoire :

Cette partie devra être traitée avec grande intention, car elle concentre les notions essentielles relatives au module Algo. Prog. de ce semestre, aussi bien sur les concepts de la POO que sur l'utilisation de notions algorithmiques (en particulier l'exploitation des listes : *LinkedList*). Les règles du jeu ThinkAhead devront être scrupuleusement respectées par ce modèle mémoire.

Le diagramme de classe en fin de document devra être respecté. Ce diagramme n'est pas complet, en particulier au niveau de l'ensemble des méthodes nécessaires, mais il contient les éléments principaux. Vous devez obligatoirement définir un modèle mémoire contenant les classes : *Partie*, *Joueur* (*abstraite*), *JoueurHumain*, *JoueurOrdinateur*, *Grille*, *Case*, *Position*, *Orientation*, *AlignementCases* et *Coup*. Une partie est composée d'une grille, de deux joueurs, de la liste des coups joués et des scores des deux joueurs. La grille contient les cases et elle est de dimension $n*n$. Un coup permet de définir quel joueur a joué ce coup. En plus du joueur, les instances de cette classe contiennent la case sélectionnée (*positionCase*), ainsi que l'orientation, afin de savoir si le coup joué était dans une ligne ou une colonne. Grâce à la liste des coups, il sera possible d'annuler des coups joués afin de revenir en arrière et tester un autre choix.

La classe *AlignementCases* contient les cases correspondant à une ligne ou une colonne de la grille. La classe grille devra donc contenir une méthode (non mentionnée dans le diagramme UML) correspondant à l'extraction d'un alignement à partir des cases de la grille. Elle devra être utilisée pour renseigner la ligne ou colonne active de la grille pendant le déroulement d'une partie. Cette classe servira également à vérifier qu'un joueur humain sélectionne une case valide et à faciliter l'écriture d'algorithmes permettant à un joueur géré par l'ordinateur de suivre une stratégie particulière.

Les stratégies de l'ordinateur devront être les suivantes :

- 1) Choisir une case au hasard parmi celles qui sont libres.
- 2) Choisir la case libre ayant la valeur maximale.
- 3) Choisir la case permettant d'avoir un écart maximal entre la valeur de cette case et celle que choisira son adversaire au prochain coup. Vous supposerez que l'adversaire utilise la stratégie 2.

- 4) Utiliser la stratégie 2, mais avec un niveau de profondeur plus important, le niveau de profondeur étant paramétrable (voir en TP avec l'enseignant pour plus d'explication sur cette dernière stratégie).

Pour l'initialisation des grilles. Il devra être possible de choisir différentes configuration :

- 1) Une case à une valeur aléatoire entre 0 et 9.
- 2) Un quart des cases à la valeur 1, le quart restant à la valeur 2, et ainsi de suite pour les valeurs suivantes. Le nombre de cases concernées sera toujours arrondi au supérieur. Exemple pour une grille de 8*8, il y aura : 16 valeurs 1, 12 valeurs 2, 9 valeurs 3, 7 valeurs 4, 5 valeurs 5, 4 valeurs 6, 3 valeurs 7, 2 valeurs 8, 2 valeurs 9, 1 valeur 10, 1 valeur 11, 1 valeur 12 et 1 valeur 13.
- 3) Un tableau T de la forme $\{\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_n, y_n\}\}$ précise que la valeur x_i doit être présente y_i fois dans le tableau. A la fin de la procédure d'affectation des valeurs x_i dans la grille à partir de T. Les cases non affectées on la valeur zéro.
- 4) Variante de la méthode 3, qui utilise la méthode 2 pour choisir les valeurs des cases non affectées après utilisation du tableau T.

Remarque, vous pouvez ajouter des attributs supplémentaires à une classe du modèle mémoire à condition de pouvoir en justifier les raisons.

Le mode texte :

La grille sera affichée en mode texte est devra être similaire au résultat ci-dessous.

```

      0  1  2  3
    +---+---+---+
0 | 6 | 6 | 9 | 9 |
    +---+---+---+
1 | 1 | 0 | 3 | 5 |
    +---+---+---+
2 | 7 | 9 | 7 | 4 |
    +---+---+---+
3 | 7 | 4 | 9 | 2 |
    +---+---+---+

```

La grille produite devra permettre de savoir quel joueur a sélectionné une case. Les scores des joueurs devront être affichés. Le numéro de la ligne ou de la colonne dans laquelle doit jouer un joueur devra également être affiché. Exemple d'affichage en mode console :

```

coup n°1 (Alain)
Vous devez jouer dans la colonne n°2
Les positions possibles sont : [0, 1, 3]
Saisissez la position où vous jouez :

```

Vous testerez les différentes méthodes d'initialisation des valeurs de la grille, ainsi que les différentes stratégies des joueurs ordinateurs en les faisant jouer les un contre les autres.

Le mode graphique :

Pour le mode graphique, jtuto est fortement conseillé. Aucune consigne particulière sur cette partie, sinon le fait qu'il doit être possible de faire une partie entre deux joueurs humains ou entre un joueur humain et un ordinateur (avec choix du type de stratégie) en respectant les règles du jeu. L'annulation des coups en mode graphique doit également être possible.

Diagramme de Classe du modèle mémoire de ThinkAhead

