
Fiche 4 : Représentation d'un plateau de jeu à 2 joueurs

Durée 60 min - Travail individuel avec possibilité d'interactions collectives via le WIKI

Objectif : Savoir représenter des connaissances dans un formalisme adapté à un moteur de résolution pour un jeu à 2 joueurs.

Module de représentation du jeu

Dans cette première étape de la réalisation de votre programme, vous allez réaliser la supervision et l'affichage d'une partie de jeu entre deux joueurs humains. Votre programme affichera la grille de jeu, saisira les coups, vérifiera qu'ils sont corrects et modifiera la grille de jeu en conséquence. A la fin de la partie, il affichera le score et le vainqueur.

Remarque : Dans questions qui suivent, la description des prédicats n'est pas forcément complète. Dans certains cas, il vous faudra écrire des prédicats intermédiaires afin de réaliser des prédicats demandés.

Si besoin, vous pouvez vous servir du WIKI pour partager des informations ou rajouter des informations sur le contenu des modules ou des prédicats (sans pour autant donner la solution !).

REPRÉSENTATION DE LA GRILLE DE JEU

Nous allons représenter la grille de jeu à l'aide de listes Prolog.

Une ligne de la grille de jeu sera représentée par une liste à N éléments (N correspondant à la largeur du plateau de jeu) représentant chacun une case de la ligne. Une case peut être vide (représentée par '-') ou occupée par un pion noir ('x') ou par un pion blanc ('o').

La grille de jeu complète sera représentée par une liste de N listes, chacune de ces listes représentant une ligne de la grille, de la ligne supérieure à la ligne inférieure.

1. Ecrivez la clause fait **grilleDeDepart(+Grille)** qui est satisfait si la **Grille** est la grille correspondant à la position de départ du jeu.

Dans tout le texte qui suit, nous parlerons de *grille* pour faire référence à la liste contenant toute la grille de jeu et nous parlerons de *ligne* pour faire référence à une liste représentant une ligne quelconque.

Une ligne sera repérée par un nombre entier de 1 à N, son numéro depuis le haut de la grille. Une colonne sera repérée par une lettre de *a* à *h*, de la gauche vers la droite. Ainsi la case en haut à gauche de la grille sera la case *a1* et celle en bas à droite la case *h8*.

VISUALISATION DE LA GRILLE DE JEU

Vous allez maintenant écrire les prédicats utiles à l'affichage de votre grille de jeu :

2. Ecrivez un ensemble de clauses faits `succNum(?Val1,?Val2)` pour définir le successeur de chaque entier de 1 à 8. De la même manière, écrivez un ensemble de clauses faits `succAlpha(?Alpha1,?Alpha2)` pour définir le successeur de chaque lettre de l'alphabet entre *a* et *h*. Ces prédicats nous permettront de passer d'une case à l'autre ou d'une ligne à l'autre.
3. Ecrivez le prédicat `afficheLigne/1` qui écrit une ligne d'une grille. Ce prédicat prend en paramètre la liste à afficher. Vous pouvez utiliser le prédicat prédéfini `tab/1` qui permet d'écrire un certain nombre d'espaces. Par exemple, `tab(3)` écrit 3 caractères 'espace'.
4. Ecrivez le prédicat `afficheGrille/1` qui écrit la grille de jeu à l'écran. Ce prédicat prendra en paramètre une grille de jeu (une liste de listes). On souhaiterait aussi avoir les indications relatives aux coordonnées de la grille (affichage en première ligne de la liste des coordonnées des colonnes, et au début de chaque ligne, le numéro de la ligne correspondante).
5. Ecrivez le prédicat récursif `ligneDansGrille(+NumLigne, +Grille, ?Ligne)` qui est satisfait si `Ligne` est la ligne numéro `NumLigne` dans la `Grille`. De la même façon, écrivez le prédicat récursif `caseDansLigne(+Col, +Liste, ?Valeur)` qui est satisfait si `Valeur` est dans la `Liste` en position `Col`.
6. Ecrivez le prédicat `donneValeurDeCase(+Grille, +NumColonne, +NumLigne, ?Valeur)` qui est satisfait si il y a une valeur `Valeur` dans la case de coordonnées [`NumColonne`, `NumLigne`] de la `Grille`.
7. Ecrivez le prédicat `caseVide(+Grille, +NumColonne, +NumLigne)` qui est satisfait si la case de la grille correspondante est vide.