RAPPORT IA02 – Projet Jeu de Khan

I – Principaux prédicats du jeu

I.1 – Les faits statiques

I.1.1 – Liste des faits statiques

Faits servant à définir les numéros des joueurs

* player1(1).
* player2(2).

Liste des pions

* pawnList(['S1', 'S2', 'S3', 'S4', 'S5', 'K'])

Plateau de jeu

* chooseBoardDisplay(Clé, PlateauVouluSelonClé)
* Fait statique qui permet au joueur de choisir la disposition du plateau qu’il veut

I.1.2 – Utilisation des faits statiques

Avoir la valeur d’une case ayant pour coordonnées X et Y

* get\_cell\_value(X, Y, CellValue).

Récupérer le code de l’autre joueur

* get\_other\_player(ActualPlayer, Player\_2)
* get\_other\_player(ActualPlayer, Player\_1)

Vérifier qu’une case de coordonnées X et Y soit sur le plateau

* cell\_in\_board(X,Y)

I.2 – Les faits dynamiques

Le plateau de jeu « actif »

* :-dynamic(activeBoard/1).
* Permet de sauvegarder et récupérer la disposition du plateau choisie par le joueur au début du jeu

Indices I et J qui servent au déplacement sur le plateau de jeu au cours de la partie

* :-dynamic(i/1).
* :-dynamic(j/1).

Prédicat pawn/4 qui sert à gérer les pions des joueurs sur le plateau

* :-dynamic(pawn/4).
* S’utilise de la façon suivante : pawn(Indice\_ligne, Indice\_colonne, Pion, Joueur)
* Permet énormément de choses :
  + En ayant la ligne et la colonne, on peut savoir s’il y a un pion sur la case et si oui, quel est ce pion et à quel joueur il appartient
  + En ayant le pion et le joueur, on peut savoir si le pion du joueur est sur le plateau et si oui, à quels indices il se situe
  + En ayant la ligne, la colonne et le joueur, on peut savoir si la case est vide, possède un pion adverse ou bien un pion du joueur.
* C’est un fait dynamique qui nous sera utile tout au long du jeu pour les déplacements des pions et le calcul des coups possibles
  + Permet de calculer la liste de tous les pions d’un joueur actuellement sur le plateau, de même que la liste des pions d’un joueur sortis du plateau

Prédicat khan/2 permettant de gérer la position du Khan

* :-dynamic(khan/2).
* Les deux arguments sont les indices de ligne et de colonne
* Fait dynamique qui est utile pour :
  + Récupérer la valeur de la case actuelle du Khan
  + Vérifier qu’une case possède la même valeur que la case actuelle du Khan

III – L’affichage du plateau de jeu

Affichage du plateau de jeu grâce au prédicat *dynamic\_display\_board(Board)* qui affiche la variable *Board* en utilisant les fait dynamiques *I* et *J*.

Nous avons aussi mis en place un un affichage des pions des joueurs en couleur afin de bien les distinguer entre eux.

IV – Placement des pions au tout début du jeu

Prédicats de contrôle servant à vérifier les informations entrées par l’utilisateur :

* Prédicat *pawn\_in\_initial\_lines(X,Y,Player)*
  + Vérifie, selon que *Player* soit le joueur 1 ou 2, que les coordonnées X et Y correspondent à une case situées sur les deux premières lignes faisant face au joueur.
* Prédicat *initial\_pawn\_placement\_correct(Player, X, Y)*
  + Vérifie que le joueur *Player* ne possède pas déjà un pion sur la case de coordonnées X et Y.
  + Vérifie aussi que les coordonnées X et Y soit sur les deux première lignes faisant face au joueur grâce au prédicat *pawn\_in\_initial\_lines(X,Y,Player)*

Placement initial des pions :

* Demande à chaque joueur de placer ses pions (on part de la liste des pions non utilisés des joueurs
* Puis à chaque itération, on demande à un joueur de placer un pion jusqu’à que la liste des pions inutilisés soit vide
  + Pour chaque placement, on vérifie bien que les coordonnées entrées par l’utilisateur soit correctes

V – Mouvements des pions au cours du jeu