

Projet Prolog: Okiya

IA02 – Université de Technologie de Compiègne
Semestre de Printemps 2013

Présentation et règle du jeu

Okiya est un jeu de pion sur le thème de la concurrence entre les Okiya, les maisons de Geishas, dans le Japon traditionnel.

Okiya oppose deux joueurs plaçant leur Geishas sur un plateau de tuiles disposé aléatoirement. Un joueur gagne lorsqu'il a réussi à aligner 4 geishas en ligne, à grouper 4 geishas en carré ou en rendant l'action du joueur adverse impossible.

Matériel

Le jeu compte 16 tuiles *Jardin Impérial* différentes. Chaque tuile dispose d'un type de végétation parmi : érable, cerisier, pin et iris ; ainsi qu'un sujet parmi : soleil, oiseau, pluie, tanzaku (bandeau avec un poème).

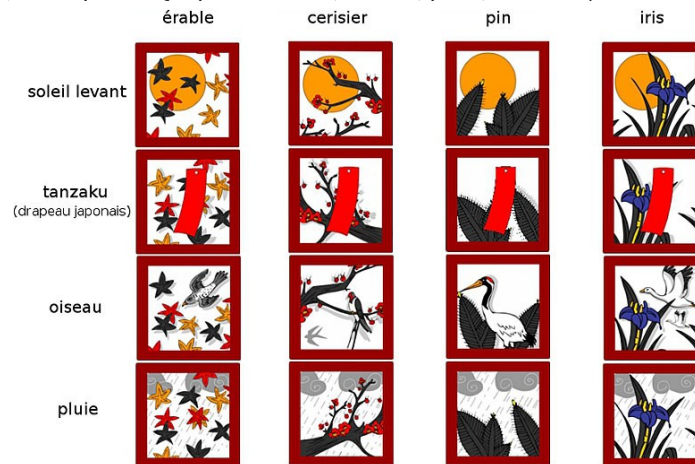


Figure : Les tuiles "Jardin imperial"

Chaque joueur dispose de 8 *Geishas* de sa couleur.



Figure : Deux types de geishas

Les tuiles sont disposées aléatoirement en début de partie sur une grille 4x4.

Déroulement de la partie

De manière générale, les joueurs positionnent chacun leur tour une geisha sur un emplacement de tuile, retirent cette dernière et la place, face visible, entre les 2 joueurs.

Au premier tour, le premier joueur est libre de remplacer n'importe quelle tuile par l'une de ses geishas.



Figure : Plateau en début de partie

Ensuite, les joueurs sont contraints à remplacer uniquement les tuiles ayant la même végétation ou le même sujet que la tuile retirée par le joueur précédent.



A son tour, le joueur noir peut choisir une tuile Iris (A1, A2 ou A3) ou une tuile avec un Tanzaku (B1, B2 ou B3).

Figure : Plateau à l'issue du premier coup de J1



Le joueur noir ayant décidé de jouer en B3, rouge va devoir jouer sur une tuile érable (C1, C2 ou C3) ou Tanzaku (D1 ou D2).

Figure : Plateau à l'issue du coup de J2

Projet

Objectif

Il est demandé d'implémenter en Prolog le jeu complet de façon à permettre une partie Humain vs. Humain, Humain vs. Machine ainsi que Machine vs. Machine. Le code sera testé afin de vérifier le bon respect des règles du jeu. Les règles à implémenter sont celles décrites dans le présent document.

Structure de données

Le plateau de jeux sera représenté par une liste de liste, ligne par ligne :

[[L1C1, L1C2 , L1C3, L1C4], [L2C1, L2C2 , L2C3, L2C4], [L4C1,L4C2 , L4C3, L4C4], [L4C1,L4C2, L4C3, L4C4]]

L2C3 représentant la case à la ligne 2 et à la colonne 3.

Les éléments de la liste peuvent prendre les valeurs suivantes :

- une paire [Vegetation, Sujet] dans laquelle Vegetation peut prendre les valeurs (erable, cerisier, pin, iris) et Sujet peut prendre les valeurs (soleil, oiseau, pluie, tanzaku) si la case est occupé par une tuile.

- j1 si la case correspond à une geisha du joueur 1
- j2 si la case correspond à une geisha du joueur 2;

Prenons l'exemple du plateau en figure 3, celui-ci se note :

[[[oiseau, erable],[tanzaku,iris],[soleil,erable],[pluie,cerisier]], [[tanzaku,pin],[oiseau,pin],[pluie,erable],[soleil,iris]], [pluie,iris],[pluie, pin],[soleil,cerisier],[oiseau,cerisier]], [[soleil,pin],[oiseau,iris],[tanzaku,cerisier],[tanzaku,erable]]]

On définit un coup (tour de jeu) en précisant le joueur (correspondant à une couleur de geisha) et la tuile d'arrivée (Joueur, Tuile).

Par exemple le premier coup du joueur 1 (cf figure 4) peut se noter ainsi:

(j1, [tanzaku,iris])

A l'issu de ce coup, le plateau devient (figure 4):

[[[oiseau, erable], j1 ,[soleil,erable],[pluie,cerisier]], [[tanzaku,pin],[oiseau,pin],[pluie,erable],[soleil,iris]], [pluie,iris],[pluie, pin],[soleil,cerisier],[oiseau,cerisier]], [[soleil,pin],[oiseau,iris],[tanzaku,cerisier],[tanzaku,erable]]]

Détails des tâches du projet

Le projet se décompose en 3 grandes étapes :

Affichage d'un plateau de jeu

Écrire le prédicat 'affiche_plateau' permettant l'affichage du plateau de jeu dans la console, et réfléchir à

l'interface de jeu.

affiche_plateau(+Plateau)

Jeu humain

Écrire le prédicat 'plateau_depart' permettant de générer le plateau initial du jeu.
`plateau_depart(?Plateau)`

Écrire le prédicat 'coup_possible' vérifiant qu'un coup est jouable pour un plateau donné, en suivant les règles du jeu.

`coup_possible(+Plateau, +Tuile, ?Coup)`

Écrire le prédicat 'jouer_coup' qui permet de déterminer l'état du jeu après l'application d'un coup. Il sera défini sous la forme :

`jouer_coup(+PlateauInitial, ?Coup, ?NouveauPlateau)`

Faire une boucle de jeu interactive, permettant à deux joueurs humains de s'affronter en respectant les règles du

jeu.

Faites en sorte que le programme soit robuste aux erreurs de saisie courantes.

Intelligence Artificielle

Écrire un prédicat qui permet de lister tous les coups possibles à partir de l'état de jeu courant

`coups_possibles(+Plateau,+Tuile,-ListeCoupsPossibles)`

Écrire le prédicat 'meilleur_coup'. Ce prédicat a pour objectif de déterminer, à partir du plateau courant, quel est le « meilleur » coup à jouer. Pour cela, vous devrez prendre en compte les différentes façons de gagner.

`meilleur_coup(+Plateau,+Tuile, -Coup)`

Intégrer l'intelligence artificielle dans la boucle de jeu, en donnant le choix au départ entre une partie « humain contre humain », « humain contre machine » ou « machine contre machine ».

Rendu

Le projet est à rendre pour la dernière séance de TP (commune aux groupes A et B), où une soutenance / démonstration de 10 minutes sera organisée pour chaque groupe.

En plus de la soutenance, il est demandé de fournir le code Prolog (commenté) du projet ainsi qu'un rapport papier contenant notamment les éléments suivants :

- Une présentation des prédicats principaux composant le jeu.
- La description de l'IA et des choix faits au niveau de l'implémentation et de l'évaluation des coups.
- Les difficultés rencontrées et améliorations possibles

Remarques :

- Le rapport ne doit pas être un simple listing de code !
- Pensez à fournir une version électronique au format PDF en même temps que vos codes sources.