

Projet IA02 : Gygès

Semestre de Printemps 2011

1 Présentation et règles du jeu

Le projet Prolog du printemps 2011 s'articule autour du jeu de plateau *Gygès*. Le site <http://www.gyges.com/> contient une explication complète des règles du jeu et offre la possibilité de jouer en ligne contre de vrais adversaires (rubrique « jouer en ligne ») ou contre une intelligence artificielle (rubrique « entraînement / Méthis »).

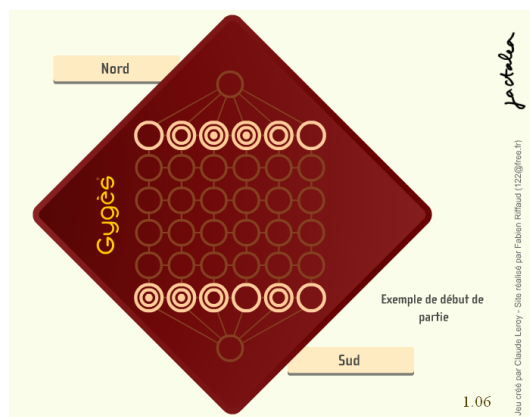


Fig. 1: Plateau de départ de Gygès (source : www.gyges.com)

Gygès se joue à deux (Nord et Sud) en déplaçant des pions (anneaux) sur un plateau composé d'une matrice de 6×6 cases reliées par des lignes horizontales et verticales, plus une case d'arrivée pour chaque joueur. Le joueur gagne dès qu'il place un anneau sur la case d'arrivée de son adversaire.

1.1 Pions

Le jeu compte 12 pions, de trois types différents : 4 anneaux simples (1 déplacement), 4 anneaux doubles (2 déplacements), 4 anneaux triples (3 déplacements).

Les pions sont tous de la même couleur et n'appartiennent donc pas à un joueur en particulier. Un joueur ne peut cependant déplacer que les pions se

trouvant sur la ligne non-vide la plus proche de lui.

Dans le cas où le joueur est dans l'impossibilité de jouer les pions situés sur la ligne la plus proche (aucun des pions de cette ligne ne peut se déplacer sans enfreindre les règles), il joue un des pions situés sur la ligne suivante.

1.2 Position de départ

Chaque joueur prend deux pions de chaque type (2 simples, 2 doubles et 2 triples). Le premier joueur (Sud) place ses pions comme il le souhaite sur sa première ligne. Le second joueur (Nord) fait ensuite de même. La partie commence alors : chaque joueur déplace un pion à tour de rôle.

1.3 Déplacement d'un pion

Un pion se déplace d'autant de cases (exactement, ni plus ni moins) qu'il a d'anneaux. Ainsi un anneau double se déplace de 2 cases.

Un pion peut se déplacer dans les quatre directions, en suivant les lignes tracées sur le plateau, mais ne peut pas passer par une case occupée. On ne peut pas se déplacer en diagonale (mais on peut changer de direction en cours de déplacement). On ne peut également pas passer plus d'une fois par la même ligne au cours d'un déplacement.

Un pion peut terminer son déplacement sur une case déjà occupée. Deux possibilités s'offrent alors au joueur : rebondir ou remplacer.

1.3.1 Rebondir

Le pion rebondit sur la pièce atteinte d'autant de cases que cette pièce a d'anneaux.

Un pion peut rebondir successivement sur plusieurs pions.

Attention : lorsqu'on rebondit sur un pion triple par exemple, il faut pouvoir effectuer au moins 2 déplacements avant de pouvoir rebondir à nouveau.

1.3.2 Remplacer

Le pion remplace la pièce présente sur sa case d'arrivée. Cette pièce est alors placée sur n'importe quelle case libre du plateau, sauf derrière la première ligne de l'adversaire.

Un pion peut commencer par rebondir sur une ou plusieurs pièces, puis finir son déplacement sur un autre pion et le remplacer. Le déplacement d'un pion peut donc être long et complexe, par rebonds successifs.

1.4 Déplacements interdits

Un pion, lors de son déplacement, ne peut passer qu'une seule fois par la même ligne (arc reliant deux cases). Cela n'empêche néanmoins pas de passer deux fois par la même case (en accédant à chaque fois à la case par des lignes différentes).

Le plateau doit être modifié après un tour de jeu (il est interdit de faire du « sur-place », un coup nul).

Un pion ne peut pas passer par une case d'arrivée. Il ne peut que s'y arrêter en tombant « pile » (ce qui entraîne par conséquent la victoire).

Un pion ne peut pas passer par dessus un autre pion.

Pour gagner, un pion doit finir son déplacement exactement sur la case d'arrivée.

Un pion remplacé ne peut pas être envoyé au delà de la première ligne de l'adversaire.

2 Projet

2.1 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 et sur le site <http://www.gyges.com/>.

2.2 Structure des données et notation des coups

Les joueurs seront représentés par les atomes *n* pour Nord et *s* pour Sud.

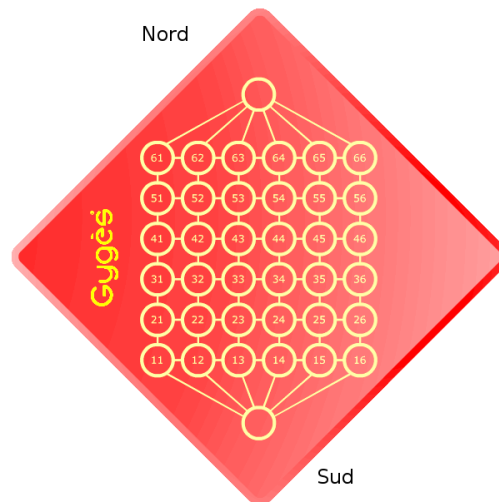


Fig. 2: Numérotation des cases

Afin de respecter au mieux les conventions de notation, les cases du plateau sont désignées par des numéros à 2 chiffres, le premier indiquant la ligne (1 pour sud, 6 pour nord) et le second le numéro d'ordre de la case de gauche à droite

du point de vue de Sud (Fig. 2). Les buts (cases d'arrivée) au Nord et au Sud seront respectivement notées n et s .

L'état du jeu courant est représenté par la liste Plateau= $[[P1a,P1b,P1c,P1d], [P2a,P2b,P2c,P2d], [P3a,P3b,P3c,P3d], J]$ où :

- $P1a,P1b,\dots$ représentent les positions des pions à 1 anneau
- $P2a,P2b,\dots$ représentent les positions des pions à 2 anneaux
- $P3a,P3b,\dots$ représentent les positions des pions à 3 anneaux
- J est le nom du joueur qui doit faire le prochain mouvement (n ou s)

Ainsi, si c'est au joueur Sud de commencer, le plateau de la figure 1 est représenté par la liste

$[[14, 16, 61, 66], [13, 15, 62, 65], [11, 12, 63, 64], s]$

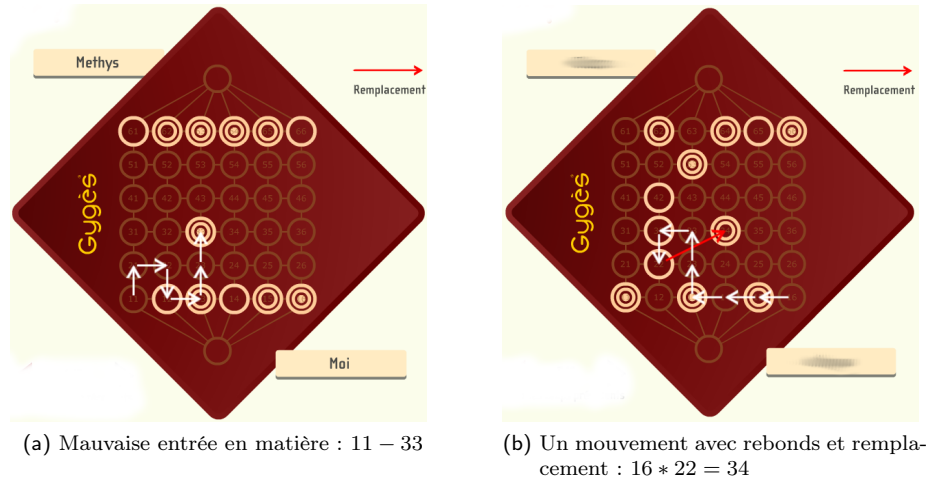


Fig. 3: Exemples de déplacements

Un coup (tour de jeu) se note de la façon suivante :

- Pour un simple déplacement (avec éventuellement des rebonds intermédiaires), on utilisera l'expression *CaseDépart*–*CaseArrivée*. Par exemple : 11 – 33, le pion de la case 11 se déplace en case 33 (Fig. 3a).
- Pour un déplacement avec remplacement, on utilise l'expression *CaseDépart***CaseArrivée* = *NouvelCaseDuPionDélogé*. Exemple : 16 * 22 = 34, le pion de la case 16 se déplace en case 22. Le pion qui se trouvait auparavant sur la case 22 est déplacé en 34 (Fig. 3b).

2.3 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, en demandant aux joueurs la position initiale de leurs pions sur leur ligne.

```
plateau_depart(?Plateau)
```

Écrire le prédicat 'coup_possible' vérifiant qu'un coup est jouable pour un plateau donné, en suivant les règles du jeu. Proposer une variante qui permet également de retrouver le chemin parcouru par le pion.

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

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.

Intelligence Artificielle

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

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

Réfléchir à une façon d'évaluer si une situation est favorable au joueur Nord ou au joueur Sud. Écrire un prédicat qui évalue un plateau de jeu, l'évaluation étant négative si le jeu est favorable au joueur Sud, et positive si le jeu est favorable au joueur Nord.

```
evaluation(+Plateau, -Valeur)
```

Écrire le prédicat 'meilleur_coup'. Ce prédicat a pour objectif de déterminer, à partir du plateau courant, quel est le « meilleur » coup à jouer.

```
meilleur_coup(+Plateau, -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 ».

2.4 Remarques

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

La possibilité de « remplacer » peut mener à beaucoup de choix possibles à chaque tour de jeu. Tout en n'entravant pas les possibilités de jeu de l'être humain, il sera peut-être nécessaire de limiter par une heuristique la liste des coups étudiés par l'ordinateur s'il prévoit plusieurs coups à l'avance, afin d'éviter l'explosion combinatoire.

2.5 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 :

- Un tableau présentant la répartition des tâches au sein du groupe.
- 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.