

Projet de réseau

jeu du morpion aveugle

L'objectif du projet est de réaliser en binômes un jeu de morpion aveugle en réseau.

1 Sujet

1.1 Description générale

1.1.1 Jeu du morpion

Le jeu de *morpion* est un jeu à 2 joueurs qui se déroule sur une grille 3×3 ; chaque joueur à tour de rôle marque une *case* dans la grille (le premier joueur utilise la marque *X* et le second la marque *O*). Le gagnant est le premier joueur qui arrive à aligner 3 de ses marques (voir Figure 1). Si la grille est complètement remplie sans qu'il y ait 3 marques identiques alignées, il y a match nul. Dans la suite le terme *grille* désigne un état du jeu. Le jeu commence par la grille vide. La Figure 1 représente une séquence de 8 grilles correspondant à une partie.

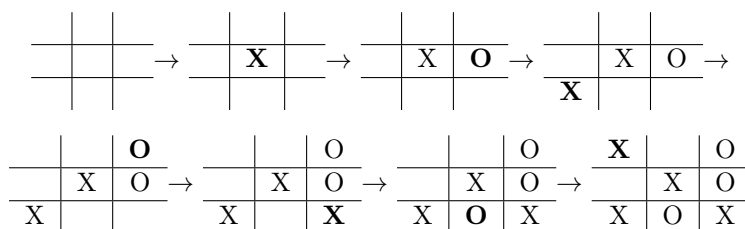


FIGURE 1 – Exemple de partie de morpion. Le premier joueur commence par marquer la case du centre, puis le second joueur joue au milieu à droite, ainsi de suite. Ici la partie se termine sur une victoire du premier joueur.

Par convention nous allons numéroté les cases comme indiqué dans la Figure 2.

0	1	2
3	4	5
6	7	8

FIGURE 2 – Numérotation des cases de la grille.

Avec cette convention la partie de la Figure 1 correspond aux coups 4,5,6,2,8,7,0 (les coups impairs étant joués par le joueur 1 et les coups pairs par le joueur 2).

1.1.2 Jeu de morpion aveugle

Le *morpion aveugle* est une variante de ce jeu dans laquelle les joueurs ne voient pas les coups joués par l'adversaire. Si un joueur essaie de marquer une case déjà marquée par l'adversaire, le joueur est informé que cette case est prise et il doit marquer une autre case. La Figure 3 représente un exemple de partie. Comme à la bataille navale, chaque joueur n'a qu'une vision partielle de l'état du jeu. Essayer de marquer une case déjà prise est un coup intéressant car il permet de savoir où l'adversaire a joué.

Vision/coups du joueur 1	Etat réel du jeu	Vision/coups du joueur 2																											
case 4																													
<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>					X					<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>					X					<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>									
	X																												
	X																												
		case 2																											
<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>					X					<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>			O		X					<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>			O						
	X																												
		O																											
	X																												
		O																											
case 6																													
<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>					X		X			<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>			O		X		X			<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>			O						
	X																												
X																													
		O																											
	X																												
X																													
		O																											
		case 4																											
<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>					X		X			<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>			O		X		X			<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>			O		X				
	X																												
X																													
		O																											
	X																												
X																													
		O																											
	X																												
		case 8																											
<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>					X		X			<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td>O</td></tr></table>			O		X		X		O	<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td>O</td></tr></table>			O		X				O
	X																												
X																													
		O																											
	X																												
X		O																											
		O																											
	X																												
		O																											
case 2																													
<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>			O		X		X			<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td>O</td></tr></table>			O		X		X		O	<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td>O</td></tr></table>			O		X				O
		O																											
	X																												
X																													
		O																											
	X																												
X		O																											
		O																											
	X																												
		O																											
case 0																													
<table><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>	X		O		X		X			<table><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td>O</td></tr></table>	X		O		X		X		O	<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td>O</td></tr></table>			O		X				O
X		O																											
	X																												
X																													
X		O																											
	X																												
X		O																											
		O																											
	X																												
		O																											
		case 5																											
<table><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>X</td><td></td></tr><tr><td>X</td><td></td><td></td></tr></table>	X		O		X		X			<table><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>X</td><td>O</td></tr><tr><td>X</td><td></td><td>O</td></tr></table>	X		O		X	O	X		O	<table><tr><td></td><td></td><td>O</td></tr><tr><td></td><td>X</td><td>O</td></tr><tr><td></td><td></td><td>O</td></tr></table>			O		X	O			O
X		O																											
	X																												
X																													
X		O																											
	X	O																											
X		O																											
		O																											
	X	O																											
		O																											

FIGURE 3 – Exemple de partie du morpion Aveugle. Le joueur 2 finit par gagner.

1.2 Information relative au code source

Le projet est packagé de la manière suivante et disponible à <http://dept-info.labri.fr/~thibault/Reseau/morpion.tgz> :

```
grid.py
main.py
```

Ce code vous permet de jouer à un joueur contre la machine (qui joue simplement des coups aléatoires).

2 Passage en réseau

Il s'agit maintenant de jouer à 2 joueurs se connectant sur un même serveur...

Dans un premier temps, faites une version simple avec un support pour seulement 2 joueurs (clients) à la fois. Sur le serveur, on lance le programme sans argument, et le programme doit alors ouvrir une socket TCP en écoute. Sur les clients, on lance le programme avec le nom du serveur en paramètre, le programme doit, dans ce cas, se connecter en TCP à la machine passée en paramètre.

Maintenant que l'on a une socket établie entre chaque client et le serveur, il s'agit de transmettre les informations sur les coups joués et l'état de la grille du joueur (ce sont les clients qui feront le rendu du

jeu chacun de leur côté). Il faut donc établir un protocole pour les exprimer. Il est recommandé d'établir un protocole textuel ascii, qu'il sera ainsi facile de déboguer à la main. Réfléchissez et implémentez dans un premier temps votre propre protocole. Puisque le morpion se joue chacun son tour, le serveur pourra simplement réceptionner tour à tour depuis chaque joueur.

3 Extensions

Plusieurs extensions sont possibles, à vous d'essayer d'en implémenter !

- On peut vouloir jouer plusieurs parties, en comptant les scores.
- Un des joueurs peut être un robot qui joue aléatoirement des coups.
- Si un troisième client se connecte, c'est un observateur : il voit tous les coups joués (il n'est donc pas aveugle).
- Lorsque la partie est finie, le serveur demande qui veut jouer la partie suivante parmi les 3 clients, et le 3e se retrouve observateur.
- Il peut y avoir bien plus que 3 clients, les autres étant aussi observateurs.
- Il se peut que les connections entre le serveur et les clients soient interrompues. On ne veut pour autant pas perdre la partie en cours. Du côté du serveur, on doit donc simplement mettre en pause la partie en attendant que le client perdu se reconnecte. Du côté du client, on essaie simplement de se reconnecter. Il s'agit en fait d'implémenter la couche Session du modèle ISO. Pour tester, vous pouvez simplement ajouter un `s.close()` qui survient au bout d'un certain temps (utiliser `time.time()` pour obtenir l'heure en secondes).
- Il se peut qu'un client soit tué (crash de la machine, control-C inattendu, ...), il a donc perdu l'état du jeu. Lorsqu'on le relance, il faudrait que le serveur lui redonne l'état du jeu pour pouvoir reprendre la partie.
- et tout autre idée intéressante, discutez-en avec votre chargé de TD !

4 Organisation

Le projet est travaillé et étudié en binôme mais la notation est individuelle. Le projet *doit* être réalisé avec un outil de gestion de révision (svn, git, ...) : directement dans votre *home* ou sur la savanne du CREMI (<https://services.emi.u-bordeaux1.fr/projet/savane/>). Les enseignants évalueront la contribution de chacun des éléments du binôme au travail commun. Les enseignants se réserveront la possibilité de modifier la composition de chacun des binômes. Afin de permettre un travail profitable, il est conseillé de ne pas créer de groupes avec des niveaux trop différents.

5 Rapport

Il s'agit de mettre en valeur la qualité de votre travail à l'aide d'un petit rapport. Pour cela le rapport doit explicitement faire le point sur les fonctionnalités du logiciel (lister les objectifs atteints, lister ce qui ne fonctionne pas et expliquer - autant que possible - pourquoi). Attention bien sûr au style, la grammaire, l'orthographe et la présentation.