Cet article est rédigé par des élèves. Il peut comporter des oublis et imperfections,
autant que possible signalés par nos relecteurs dans les notes d'édition.

Quoridor

Année 2022 – 2023

Simon Machado, Bastien Letteron, Titouan Weiller, Thomas Schvertz—Godon,

Steven Randriamanantsoa, Maxime Puissant, élèves de seconde

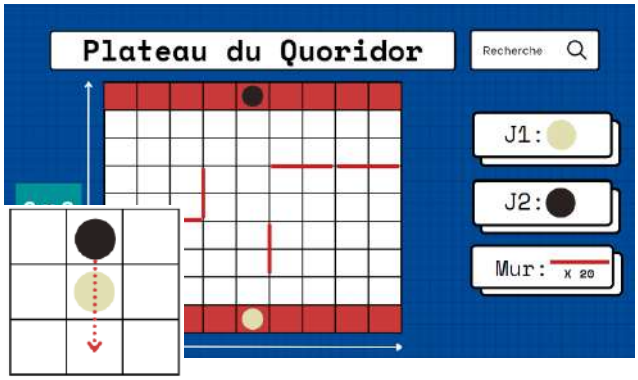
Etablissement : Lycée Marguerite de Navarre à Bourges (18)
Encadrés par : Frédéric Brinas, Olivier Créchet, Nathalie Herminer, Amélie Roche-Hernandez
Chercheurs : Benjamin Nguyen et Xavier Bultel , INSA Centre Val de Loire

Table des matières

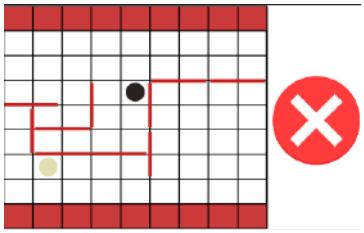
Introduction

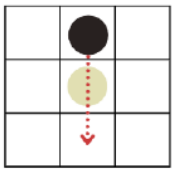
Quoridor est un jeu de stratégie combinatoire abstrait, se jouant à 2 ou 4 joueurs conçu en 1997 par Mirko Marchesi et édité par les jeux Gigamic. Ce jeu remporta la récompense Mind Game de Mensa au moment de sa sortie, en 1997, et a été élue meilleur jeu de l'année en France, aux Etats-Unis et au Canada.

Règle du jeu



Avant de commencer nos différentes recherches nous nous sommes intéressés aux règles de ce jeu et à son fonctionnement. Ce jeu composé d'un plateau de 81 tuiles carrées (9×9) ainsi que de 2 ou 4 pions, ressemble à un damier d'échecs ou de dames. Néanmoins dans Quoridor on peut placer des murs entre les cases, on en compte 20 au total, ce qui est départagé équitablement en $10 - 10$. Pour gagner, chaque joueur doit atteindre la ligne opposée de leur point de départ. L'objectif est de traverser le terrain en premier et donc le premier joueur à atteindre cette ligne gagne. Pour traverser le plateau, on peut avancer d'une case soit horizontalement ou verticalement mais pas en diagonale. La seconde option est de placer un mur, de longueur 2 cases, horizontalement ou verticalement, pour bloquer l'adversaire et donc le ralentir. Malgré tout, les joueurs ont interdiction de bloquer complètement leur adversaire. Le joueur doit toujours avoir la possibilité d'atteindre le côté opposé même si ce n'est pas en ligne droite. Il existe également une autre exception, si les deux joueurs se retrouvent collés entre deux murs, ou si l'un d'eux à un mur derrière lui ou encore dans d'autres cas de figure, les joueurs peuvent sauter par-dessus l'autre en diagonale ou de façon verticale ou horizontale.





Début des recherches

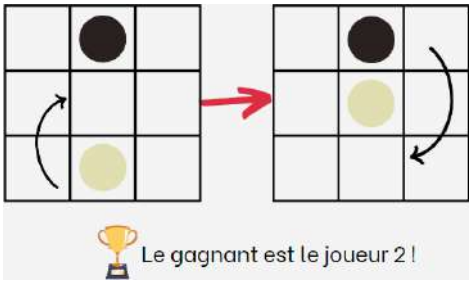
Pour comprendre un maximum comment jouer au jeu, nous avons d'abord effectué de nombreuses parties. Etant donné que le plateau est constitué de 81 cases et de 20 murs, nous avons remarqué que les issues possibles étaient trop importantes donc notre objectif était dans un premier temps de se concentrer sur un plateau de 3×3 pour que les calculs soient réalisables.

Nous avons suivi les instructions données par les chercheurs pour mener à bien nos recherches. Nous avons commencé à jouer et tester toutes les parties possibles car elles étaient beaucoup plus abordables sur cette taille de plateau. Nous avons aussi débuté la création d'un algorithme pour représenter le jeu en format 3×3. Nous sommes partis du principe que les deux joueurs jouent les meilleurs coups et ne font que très peu d'erreurs.

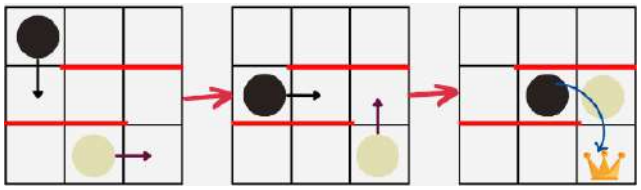
Élaboration des stratégies en 3×3

Après avoir testé toutes les parties possibles sur ce plateau, nous avons pu constater quelques stratégies pour gagner.

Tout d'abord, le joueur 1 ne doit jamais se déplacer sur la case du milieu, qui est devant lui, sinon le joueur n° 2 peut sauter au-dessus de son pion et va finalement gagner par la même occasion. De plus, le joueur 1 ne doit pas aussi se diriger vers sa gauche ou sa droite car il perdrait dans tous les cas car le joueur 2 gagnerait un tour d'avance et aurait seulement besoin d'avancer jusqu'à la ligne adverse.



Il faut donc « temporiser » avec des murs, pour le joueur 1 car si vous posez un mur, vous ne pouvez pas avancer et cela permet donc de gagner du temps. On saute donc votre tour pour éviter d'aller sur la case du milieu. Cependant, le nombre de murs ayant la possibilité d'être posés en même temps, sur ce petit plateau, est très faible. On peut en poser deux et trois au maximum dans un cas précis selon le positionnement des joueurs.



Nous avons donc constaté qu'il faut être le second joueur à commencer. Si les deux joueurs jouent les meilleurs coups, le gagnant sera forcément le second joueur. De même pour un cas avec des murs, le gagnant sera le joueur 2 car lorsque les deux joueurs se rencontreront le joueur 2 sautera au-dessus du n°1 et gagnera forcément.

Elaboration des stratégies en $3 \times X$

Nous avons ensuite voulu savoir si le second joueur était toujours le gagnant et s'il gardait encore son avantage en changeant la taille du plateau. Nous avons décidé de garder une largeur de 3 cases et de changer la longueur, que l'on a nommé X. Nous avons essayé de découvrir le mécanisme. Les stratégies restaient les mêmes que cela soit pour les plateaux de 3×3 ou de 3×7.

Mais contre tout attente, pour le plateau de largeur 3 et de longueur de 7. Le joueur 1 est grandement avantagé et si les deux joueurs jouent les meilleurs coups, le joueur 1 est sûr de gagner.

Nous avons donc voulu savoir s'il y avait une règle qui permettait de connaitre le gagnant de la partie pour tous les plateaux de largeur 3 et de longueur indéfinie. Après de nombreuses recherches, nous avons trouvé cette règle :

Nombre de Mur posable / Longueurs du plateau	Pair	Impair
Pair	1er Joueur Gagnant	2ème Joueur Gagnant
Impair	2ème Joueur Gagnant	1er Joueur Gagnant

Pour que ce tableau soit exploitable, il nous faut différentes données :

Tout d'abord, connaître le nombre de murs maximum posables sur le plateau mais sans bloquer complètement un joueur.

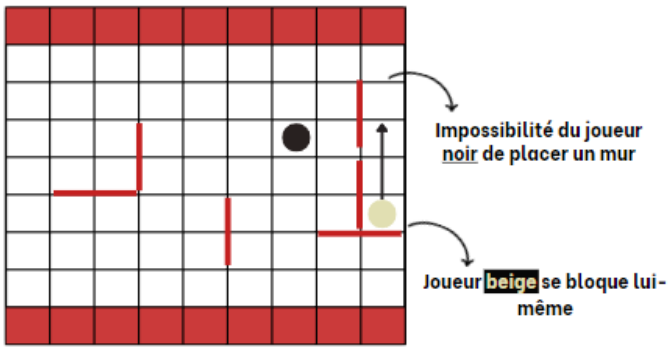
Puis la longueur X du plateau, avec largeur 3.

Début des stratégies en 9×9

À la suite de stratégies trouvées pour le 3×3 et $3 \times X$, nous avons essayé de trouver un fonctionnement commun avec nos recherches et tenté de trouver de nouvelles pistes de stratégies pour le plateau original de Quoridor (9×9) :

- Dans un premier temps, on peut essayer de se créer son propre chemin avec ses propres murs sans que l'adversaire puisse nous bloquer.

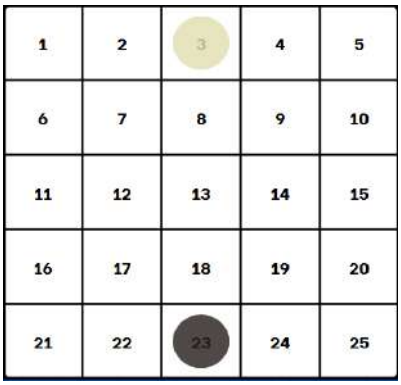
- On peut aussi faire en sorte d'attirer son adversaire sur un coté du plateau et bloquer cette partie dans le but de lui créer un passage plus long pour arriver à l'opposé.



Étant donné le manque de temps, nous avons achevé nos recherches en 3×3 sans trouver de stratégie concrète en 9×9.

Programme Python

En parallèle des recherches sur le jeu et les stratégies, nous avons commencé à programmer un algorithme en python pour représenter le jeu. Nous avons donc représenté le support sous forme de tableau et de listes. La représentation graphique de notre algorithme ressemble à ce schéma :



⚠ | n = taille du plateau (la taille sera obligatoirement un nombre impair)

La position des joueurs

La position **initiale** de chaque joueur est calculée de manière différente pour les deux joueurs :

- Pour le premier joueur, on effectue le calcul : $\frac{1+n}{2}$
- Pour le second joueur, on effectue le calcul : $n^2 - \frac{1+n}{2} + 1 = \frac{2 \times n^2 - n + 1}{2}$

Pour un plateau de côté 3, on aura :

- Le joueur 1 en position 2 car $\frac{1+3}{2} = 2$
- Le joueur 2 en position 8 car $\frac{2 \times 3^2 - 3 + 1}{2} = 8$

Remarque : Dans notre programme, le plateau sera forcément carré avec des côtés impairs pour que les joueurs soient positionnés au milieu.

Les mouvements possibles

Lorsque le plateau est mis en place et que les joueurs ont une place attribuée, le programme va choisir un mouvement aléatoire pour chacun des joueurs.

Les mouvements possibles pour chaque joueur sont :

- Avancer
- Reculer
- Aller à gauche
- Aller à droite

Ces mouvements sont enregistrés dans une liste.

Pour avancer ou reculer, *en fonction du joueur*,

On ajoute ou on retire n : $p[i]+n$ ou $p[i]-n$

Pour aller à gauche ou à droite, *en fonction du joueur*,

On retire ou on ajoute 1 : $p[i]-1$ ou $p[i]+1$

} $p[i]$ correspond à la position actuelle du joueur

Représentation de la liste : pos_possible = [p[i]-1, p[i]+1, p[i]-n, p[i]+n]

Les limites de ce programme

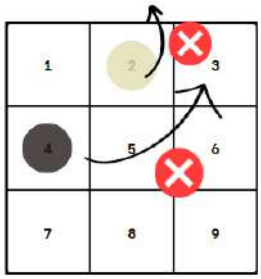
Ce programme à ce stade, a malheureusement quelques limites à la suite des règles du jeu.

1- Les bords du plateau :
Tout d'abord le joueur ne peut pas sortir du plateau, par exemple il n'est pas possible d'aller à droite lorsque le joueur se trouve sur le bord droit du plateau.

2- Le saut des joueurs
Ensuite le joueur a le droit d'effectuer un saut au-dessus du joueur adversaire dans certains cas de façon horizontale, verticale ou diagonale.

Nous avons donc pris en compte ces deux contraintes par la suite.

Les bords du plateau



En fonction de la position d'un joueur, certains mouvements sont interdits.
Si le joueur est au bord on remplace dans la liste le mouvement qui est interdit par un 0.
Pour savoir si le joueur est sur le bord, nous avons mis en place des tests pour chacun des côtés.

- Si le joueur est sur le bord gauche, on effectue le calcul : p_{joueur}/n et on vérifie si le reste est égal à 1. Si le reste est égal à 1 alors on remplace, dans la liste, le mouvement gauche, correspondant à $p[i]-1$, par 0
- Si le joueur est sur le côté droit, on effectue le calcul : p_joueur/n et on vérifie si le reste est égal à 0. Si cette condition est valide alors on remplace le mouvement par 0 dans la liste.
- Si le joueur est situé en haut du plateau, alors on vérifie si la valeur de sa position est inférieure ou égale à celle de la taille du plateau. Si cela est le cas, alors on remplace la valeur de l'action « avancer » ou « reculer » *en fonction du joueur* par $0 \longrightarrow p_{joueur} \leq n$.
- Enfin, si le joueur est en bas, la valeur de sa position sera inférieure ou égale à celle du plateau au carré **et** supérieur à celle de la différence entre la taille du plateau au carré et sa taille. Si le joueur est positionné en bas alors, on remplace la valeur par $0 \longrightarrow p_{joueur} \leq n^2$ **ET** $p_{joueur} > n^2 - n$.

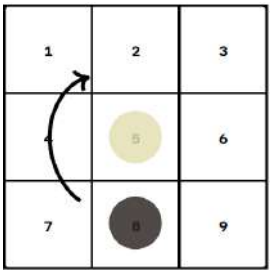
Lorsque l'on remplace la valeur par 0, le joueur sera dans l'incapacité à faire ce mouvement et choisira les autres options qui lui sont possibles.

Le saut des joueurs

Les joueurs sont parfois amenés à se rencontrer. Dans ces moments-là, ils peuvent sauter au-dessus de l'autre et donc sauter 1 case. Nous devons donc incorporer cette possibilité dans notre programme. Tout d'abord, nous recherchons où sont les joueurs par rapport à l'autre. Si les deux joueurs sont côte à côte, alors nous obtenons l'index du joueur opposé, pour savoir de quel côté il est par rapport à l'autre. Lorsque l'on connait son emplacement, nous allons remplacer la valeur de la liste des positions possibles par une autre valeur pour sauter au-dessus. Nous avons créé une liste au préalable avec les déplacements possibles comprenant les sauts :

pos_possible1 = [p[i]-2, p[i]+2, p[i]-n x 2, p[i]+n x 2]

Par exemple, si le joueur est devant lui, comme ci-dessous :



Ici on observe que le joueur adverse est devant, il a donc la possibilité de sauter par-dessus. Il peut donc aller directement en case 2. On connait maintenant la position du joueur adversaire, donc on remplace la 4ème valeur (« avancer ») de la liste des mouvements possibles par la valeur correspondante de la liste des sauts possibles. Cela donne :

pos_possible[3] = p[i] + n ⟶ pos_possible[3] = p[i] + n x 2

Nos futurs objectifs

Dans un premier temps, nous envisageons d'ajouter la possibilité de poser des murs et d'automatiser le programme pour trouver des stratégies différentes.
Nous aimerions aussi découvrir de nouvelles perspectives de jeux, qui seraient accompagnées de nouvelles stratégies avec un damier plus grand.