



## Tapatan

*Tu sais que ce que tu peux espérer de mieux est d'éviter le pire*  
(Italo Calvino, Si par une nuit d'hiver un voyageur, 1979)

Le jeu dont il est question dans ce chapitre m'accompagne depuis mon enfance. Il était arrivé un jour à la maison apporté par mon père qui, dans sa profession de commercial, était amené à recevoir ou donner de petits cadeaux tels que de petits jeux, des agendas, etc. Il consiste en une plaque de plexiglas dans laquelle sont aménagés neuf creux aux sommets, milieux et centre d'un carré et des boules de plexiglas, trois oranges et trois rouges. Les neuf creux de la plaque y sont reliés entre eux par huit segments : les côtés, médianes et diagonales du carré.



FIGURE 3.1 – Le jeu de morpion de mon enfance, dans sa boîte d'origine avec la règle qui étend le jeu au delà de la phase statique du jeu.

Au départ, il existait une boîte pour ce jeu mais celle-ci, endommagée, avait été remplacée par une vieille boîte à gâteaux métallique. La règle du « jeu de morpion » avait survécu. Pour ce jeu, appelé aussi *tic tac toe* dans sa version internationale, deux joueurs posent alternativement leurs boules sur le plateau en cherchant à réaliser un alignement. Si aucun des joueurs n'a réussi à aligner ses trois pions, le jeu continue. Chacun déplace à tour de rôle, lors de cette deuxième phase du jeu, un de ses pions le long d'un des segments du plateau vers un creux libre voisin, le but étant toujours de rechercher un alignement.

Pour de nombreuses personnes le jeu de morpion s'arrête à la fin de la première phase, aucun déplacement n'étant permis. Mais nous avons pris l'habitude d'y jouer selon cette règle qui autorise les déplacements, sans le nommer puisque, pour nous, le morpion était un autre jeu. C'était le jeu que l'on joue à l'école sur le quadrillage d'une page de cahier ; faisant des croix contre son voisin qui, lui, trace des ronds jusqu'à un alignement de trois ou de quatre. Pour nous, c'était le jeu de la boîte en fer qu'on oubliait régulièrement avec les bribes de stratégie qu'on ne manquait pas de réinventer, des mois ou des années plus tard, lorsque l'un d'entre nous le ressortait de sa boîte. J'aimais bien ce jeu qui n'était pas aussi facile que le populaire morpion des écoliers et, même après de longues années de pratique, je me fais encore avoir par mon adversaire, par un défaut de mon attention qui se centre souvent sur mes intentions, oubliant celles de mon adversaire.

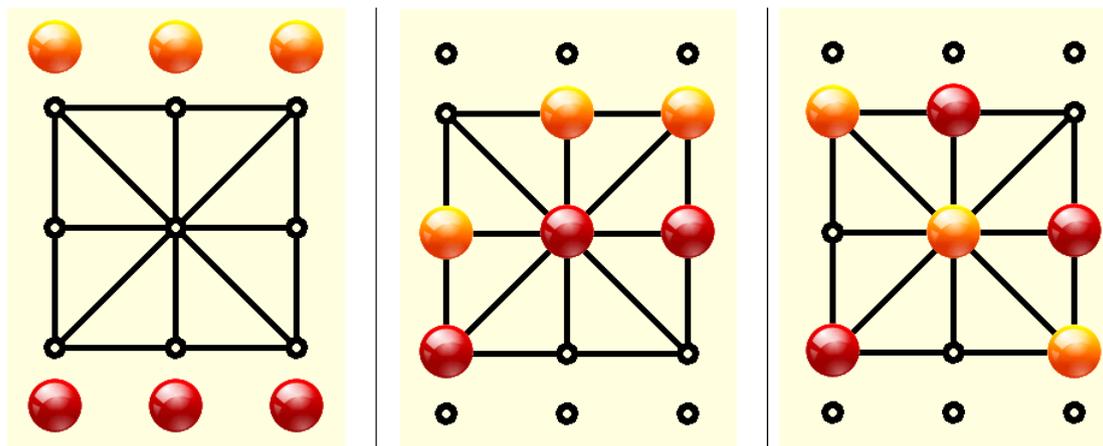


FIGURE 3.2 – Le plateau du jeu est constitué de neuf emplacements (creux, trous, dessins) reliés par huit traits. Dans la position initiale, trois pions sont donnés aux joueurs. Ceux-ci essaient de les aligner dès la phase statique (pose des pions) ou au-delà, dans une phase dynamique (déplacement vers un emplacement autorisé voisin). Le joueur du haut qui a les pions jaunes gagne, à droite, en alignant ses pions sur une des diagonales.

Devenu enseignant en mathématiques puis en NSI (Numérique et Science Informatique), me familiarisant toujours davantage avec la programmation, je m’amusais à essayer de simuler ce jeu, en Java tout d’abord, puis en Python. Je découvrais à cette occasion que ce jeu est appelé Tapatan aux Philippines et qu’il a encore bien d’autres noms comme Achi au Ghana, Three men’s morris en Grande-Bretagne, Kurna en Égypte, Shisima au Kenya, Hema en Cote d’Ivoire, Tsoro-Yemutatu au Zimbabwe, Fanoron-telo à Madagascar, Nira en Inde, Luk tsut k’i (« Jeu des six hommes ») dans le sud de la Chine, etc. Ce jeu était apparemment déjà connu des Amérindiens qui l’appelaient Pitarilla ou des romains qui l’appelaient Rota.

Extrêmement répandu sur la planète, sans doute à cause de la simplicité toute naturelle de ses règles et de la difficulté assez marquée à dégager la bonne façon de gagner, ce jeu n’a pas une définition unique. De nombreuses variantes existent qui les distinguent autrement que par le nom ou l’origine : se limiter à la phase statique est une différence majeure et le plus connu va jusqu’à remplir complètement la grille. En ajoutant la phase dynamique, les lignes tracées ne sont pas toujours les huit mentionnées, les diagonales sont parfois absentes, le carré est parfois remplacé par un cercle. Certains jouent avec quatre pions, d’autres interdisent le centre aux deux premiers coups...

Dans la suite, je resterai fidèle à la version de mon enfance dont je précise ici la règle :

- ♦ Commencement : le plateau est vide au départ. Deux joueurs reçoivent chacun trois pions d’une même couleur et laissent le hasard décider lequel commence. Après une première partie, si il y a eu un vainqueur le vaincu commence, sinon c’est au hasard de décider qui commence.
- ♦ Phase statique : chaque joueur dépose à tour de rôle un de ses pions dans un des neuf creux, n’importe lequel pourvu qu’il soit libre. Si un joueur réalise un des huit alignements possibles avec ses pions, il est déclaré vainqueur et la partie s’arrête.
- ♦ Phase dynamique : les six pions ayant été posés, un déplacement est effectué à tour de rôle par chacun des joueurs. Pour cela, un de ses pions est déplacé vers un creux vacant adjacent (relié au creux occupé par une des lignes du plateau). La partie s’arrête lorsqu’un des joueurs réalise un alignement.
- ♦ Partie nulle : pour éviter les parties infinies, on peut déclarer une partie nulle si les joueurs conviennent qu’ils sont à égalité. Dans la version informatique, une partie est déclarée nulle lorsqu’une même position est atteinte trois fois. La deuxième fois, le message que cette position a déjà été rencontrée est donné aux joueurs ; il convient de l’éviter si on espère un gain.

L’objectif de ce chapitre est l’étude de ce jeu pour en réaliser une version informatisée où un joueur humain se mesure à une intelligence artificielle. Le jeu choisi est suffisamment simple pour que cette IA soit la plus basique possible tout en restant efficace. L’algorithme à réaliser doit être compréhensible par tout un chacun et programmable avec un langage quelconque en n’utilisant que des instructions élémentaires. En ce qui me concerne, ce sera Python pour une utilisation sur ordinateur et Javascript pour une utilisation en ligne via mon site internet.

# 1. La phase statique

Cette première étape du jeu est plutôt bien connue ; de nombreuses études y sont attachées dont celle de Yves Dutrieux et Hervé Gianella, parue dans leur livre « Jeux, casse-tête et mathématiques » (pages 119-122) paru en 2021 chez Cassini. Ils étudient cependant une version différente du jeu où il est possible de remplir le plateau de pions. Une autre référence, plus ancienne et très synthétique est celle de Elwyn R. Berlekamp, John H. Conway et Richard K. Guy qui, dans « Winnig ways for your mathematical plays » (volume 3, pages 733-735) paru en 2003 chez A K Peters, procède à l'analyse complète du même jeu à quatre ou cinq pions. Cette étude identifie à chaque fois le ou les meilleurs coups pour chacun des joueurs (voir en annexe de ce chapitre l'image de cette analyse). Je vais essayer de reprendre en les adaptant les conclusions de Dutrieux et Gianella (dénommés dans la suite le duo) ainsi que celles de Berlekamp, Conway et Guy (dénommés le trio) pour construire l'algorithme de réponse de l'ordinateur pendant cette phase de jeu. Aux symétries et rotations près, il n'y a que trois commencements possibles : le centre, un coin ou un côté.

## a. Ouverture au centre

Dans le cas d'un commencement au centre, le 2<sup>e</sup> joueur peut répliquer dans un coin ou sur un côté mais la réponse sur le côté amène le 1<sup>er</sup> joueur à gagner en répliquant dans un coin du côté opposé, ce qui force le 2<sup>e</sup> joueur à bloquer la menace d'alignement et permet au 1<sup>er</sup> une double attaque imparable. En effet, quelle que soit la réponse du 2<sup>e</sup> joueur, dès le début de la phase dynamique, le 1<sup>er</sup> joueur peut l'emporter.

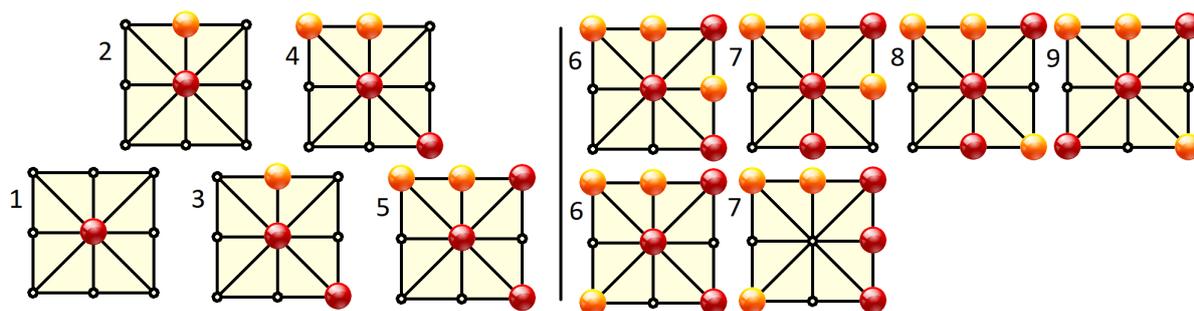


FIGURE 3.3 – Lors d'une ouverture au centre, si la réponse du joueur 2 est un placement sur le côté, elle conduit inévitablement à une double attaque au 5<sup>e</sup> coup puis à la victoire du joueur 1 au 7<sup>e</sup> ou au 9<sup>e</sup> coup.

L'analyse du trio propose également comme coup gagnant, pour celui qui commence au centre avec une réplique sur un côté, un des côtés adjacents au côté choisi par le 2<sup>e</sup> joueur, mais avec la règle du Tapatán ce début de partie conduit à une égalité.

Si, dans ce cas, il joue sur le côté opposé, l'erreur du 2<sup>e</sup> joueur au 4<sup>e</sup> coup (jouer sur un des autres côtés, voir en annexe le chemin 51932 et l'illustration ci-dessous) ne conduit pas systématiquement au gain du joueur 1 comme indiqué par le trio : avec la règle du Tapatán, celui-ci ne gagne que si le 2<sup>e</sup> joueur fait une erreur grossière au 6<sup>e</sup> coup (ligne inférieure) mais, normalement, c'est alors le 2<sup>e</sup> joueur qui gagne lors de la phase dynamique (ligne supérieure).

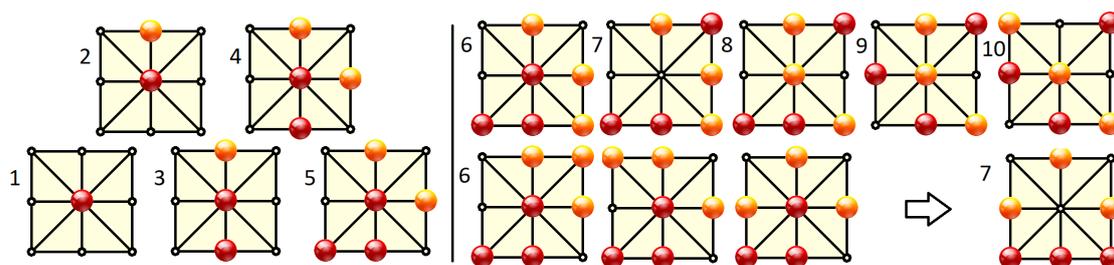


FIGURE 3.4 – La mauvaise réponse du joueur 1 au 3<sup>e</sup> coup le conduit à perdre lors de la phase dynamique si le joueur 2 joue correctement, contrairement à ce qu'annonce le trio qui se fonde sur une règle différente.

Ces derniers exemples me montrent à quel point la règle adoptée modifie les conclusions des deux groupes d'auteurs cités. Si je continue à vouloir fonder mon analyse sur celles de ces auteurs, je risque de me perdre complètement. Pour mieux mesurer l'écart entre les conclusions du trio et celles dont on a besoin avec les règles choisies, je vais adapter le début de leur tableau aux règles du Tapatán.

Le travail est fastidieux car il faut rejouer chaque partie avec, toujours, le risque de commettre une erreur, de ne pas avoir détecté un meilleur coup. Cette étape est effectuée à la main, sur un jeu réel, par un humain faillible (moi-même) et pour me compliquer encore la tâche, je me suis rajouté une difficulté supplémentaire en n'adoptant pas la même numérotation des cases que la numérotation choisie par le trio. Celle-ci est, certes, élégante et facilite l'identification d'un alignement (étant basée sur un carré magique de densité 15, il suffit que la somme des trois cases fasse 15 pour l'identifier) mais elle est moins intuitive et gêne la lecture des coups.

Voici donc le résultat de ce travail pour l'ouverture au centre. J'ai inscrit mon code des cases à droite et, pour comparer avec celui du trio (en annexe), j'ai donné la table des correspondances. J'ai aussi conservé l'ordre des différents coups et repris la typographie spécifique du trio, à savoir :

- ♦ en gras les coups forcés (pour éviter un alignement)
- ♦ ! un meilleur coup, ? un pire coup, ~ un coup aléatoire
- ♦ x le joueur ayant commencé gagne, 0 le 2<sup>e</sup> joueur gagne

Les nombreuses modifications que j'ai dû apporter aux conclusions du trio sont en bleu. J'ai eu la surprise de découvrir qu'en cas d'ouverture au centre, aucune réponse ne peut empêcher le 1<sup>er</sup> joueur de gagner s'il joue son 2<sup>e</sup> coup comme s'il plaçait son 2<sup>e</sup> pion à un saut de cavalier du pion du 2<sup>e</sup> joueur : suite à une réponse au milieu, il prend un des coins du bord opposé et suite à une réponse dans un coin, il prend un des côtes partant du sommet opposé.

Curieusement cette caractéristique ne m'était pas connue malgré les multiples parties jouées dans le cadre familial ou l'une des variantes numériques que j'ai eu l'occasion de tester... Je comprends mieux maintenant la règle additionnelle pratiquée dans une version française du Tapatán<sup>1</sup>, appelée marelle ou méréelle à trois qui interdit de jouer au centre pendant les deux premiers coups. Je réalise en même temps qu'il serait absurde de persévérer avec la règle initiale sinon l'ordinateur gagnerait à tous les coups et l'adversaire humain aussi, dès lors qu'il apprendrait cette ouverture.

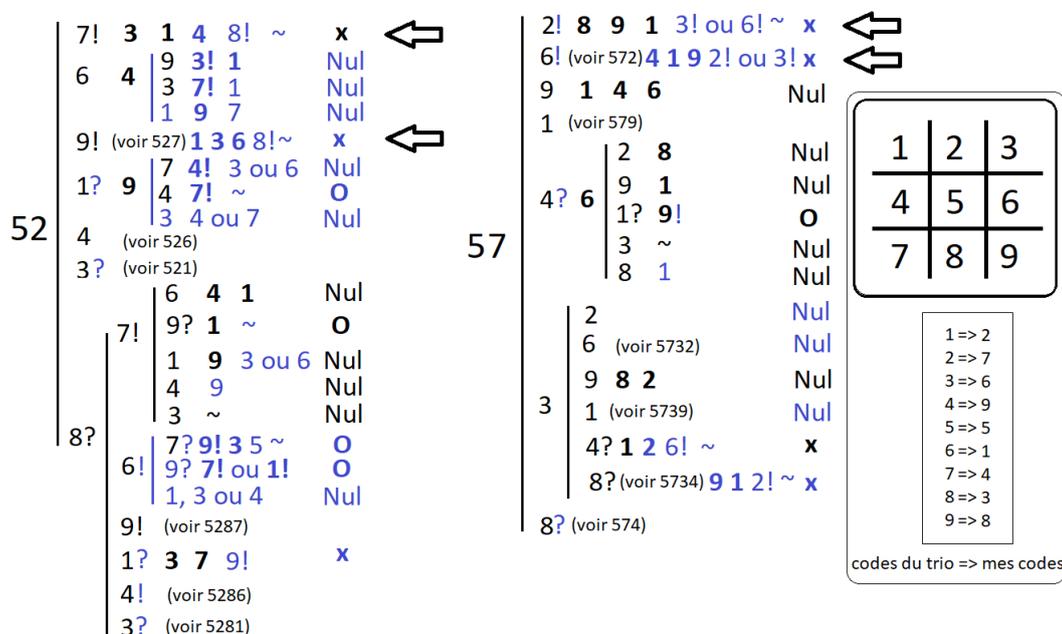


FIGURE 3.5 – Réécriture des conclusions du trio dans le cas de la règle initiale du Tapatán pour la phase statique du jeu (elle s'arrête lorsque les 3 pions de chaque joueur ont été posés). La flèche indique les coups gagnants imparables qui me forcent à revoir la règle du jeu.

1. Information lue sur le site de l'Elliott Avedon Museum & Archive Of Games de l'Université de Waterloo (Ontario, Canada) consultable à l'adresse <https://web.archive.org/web/20070208194807/http://www.gamesmuseum.uwaterloo.ca/VirtualExhibits/rowgames/tapatan.html>

La nouvelle règle doit-elle inclure les deux premiers coups dans l'interdiction du 1<sup>er</sup> coup au centre ? Je m'interroge car on pourrait seulement l'interdire au 1<sup>er</sup> joueur. En effet, dès le 2<sup>e</sup> coup, la configuration ayant changée, une stratégie imparable n'est plus possible. Je suis tenté de conserver la possibilité de répliquer au centre pour le 2<sup>e</sup> joueur, quitte à l'abandonner s'il s'avère qu'elle conduit à un gain inévitable de celui-ci ou à un trop grand avantage.

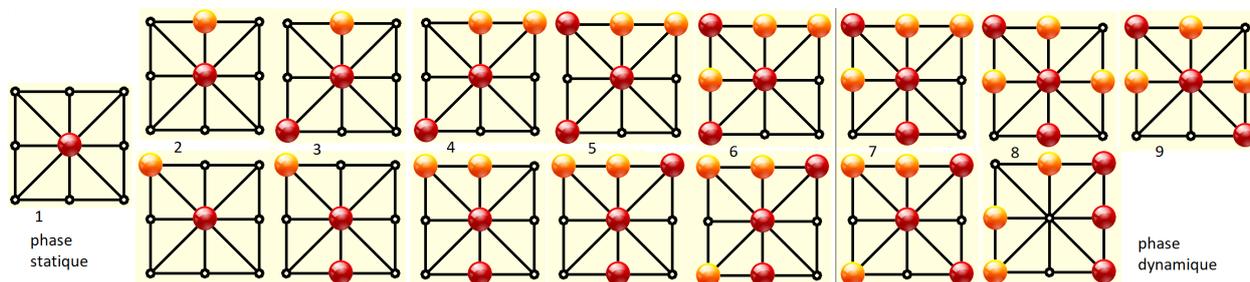


FIGURE 3.6 – Les deux réponses possibles à une ouverture au centre conduisent à la victoire du 1<sup>er</sup> joueur si celui-ci fait le « choix du cavalier » au 3<sup>e</sup> coup qui revient à forcer l'adversaire à jouer son 2<sup>e</sup> pion près du 1<sup>er</sup>.

### b. Réponse au centre

Je me demande si mon programme ne gagnerait pas à proposer les trois options :

- ♦ utilisation du centre autorisée à partir du 1<sup>er</sup> coup : traditionnel, pour joueurs débutants
- ♦ utilisation du centre autorisée à partir du 2<sup>e</sup> coup : intermédiaire, pour joueurs avertis
- ♦ utilisation du centre autorisée à partir du 3<sup>e</sup> coup : avancé, pour joueurs experts

Cette graduation des difficultés peut d'ailleurs être conjuguée avec la possibilité de contrôler ou pas les passages par des positions déjà empruntées. On l'a dit, dans la phase dynamique, cette répétition trois fois d'une même configuration de jeu conduit à un match nul. Mais pour des joueurs débutants cette option devrait pouvoir être désamorcée car difficile à gérer.

Pour tester tout de suite les avantages attachés à une réponse au centre, j'examine les deux ouvertures possibles qui la permettent, soit partout sauf au centre : un côté ou un coin. En supposant que le 2<sup>e</sup> joueur joue toujours un de ses meilleurs coups, cette réponse au centre lui assure le gain de la partie dans tous les cas sauf un : si l'ouverture est dans un coin et que le 2<sup>e</sup> coup du 1<sup>er</sup> joueur est dans un des deux coins adjacents (la partie 157463 ci-dessous), la phase statique s'achève sur une position neutre ; aucun des joueurs n'ayant un avantage décisif.



FIGURE 3.7 – Toute ouverture qui amène une réponse au centre conduit à la victoire du 2<sup>e</sup> joueur sauf si le 1<sup>er</sup> a ouvert dans un coin et qu'il place son 2<sup>e</sup> pion dans un des coins adjacents. Dans ce seul cas il peut espérer commencer la phase dynamique à égalité avec son adversaire.

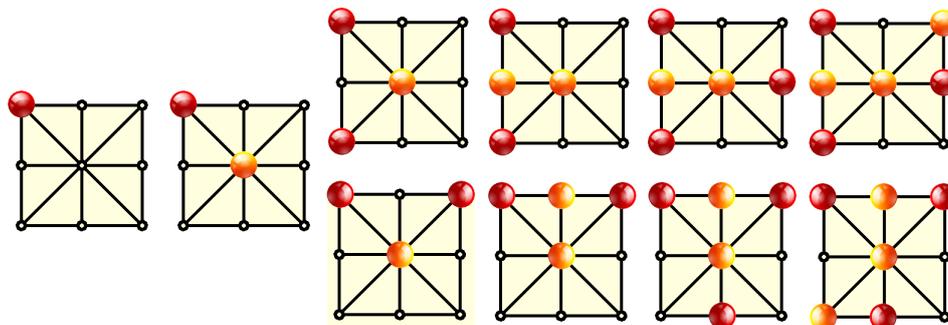


FIGURE 3.8 – La partie notée 157463 est la seule possibilité de commencer à égalité la phase dynamique lorsque le joueur 1 joue dans un coin (ici le coin supérieur, noté 1) et que le joueur 2 est autorisé à jouer au centre. La ligne inférieure montre une phase statique semblable, notée 153287.

La conclusion de cette analyse n'est pas aussi catégorique que lorsque le centre est employé dès le 1<sup>er</sup> coup, mais offrir ce privilège au 2<sup>e</sup> joueur restreint le 1<sup>er</sup> qui ne peut gagner que s'il joue deux coins contigus lors de ses deux premiers coups.

Bien sûr, si le 2<sup>e</sup> joueur fait des erreurs, un autre début peut également conduire le 1<sup>er</sup> à faire une partie nulle ou même à la gagner. Comme l'IA choisit toujours le moins pire des coups, elle commencera systématiquement de cette façon, deux coins contigus, jamais un côté. Sachant cela, en débutant dans un coin, le 2<sup>e</sup> joueur peut être amené à tester d'autres réponses que le centre, même si celui-ci est autorisé. Nous allons tout de suite envisager ce type de situation.

### c. Ouverture dans un coin ou sur un côté

Il me reste à envisager les deux ouvertures possibles qui n'utilisent pas la case centrale aux deux premiers coups. Aux translations et rotations près, il y a peu de début à envisager pour ces deux premiers coups : 12, 13, (14), 16, (17), (18) et 19 pour les ouvertures dans un coin. J'ai mis entre parenthèses les débuts de parties dont une version symétrique a été citée (14 symétrique de 12, etc.) qui n'ont donc pas à être étudiées pour elles-mêmes. Pour les ouvertures sur un côté, il reste à étudier 21, (23), 24, (26), 27, 28 et (29). Pour être complète, mon étude devrait donc considérer les huit débuts de partie différents : 12, 13, 16, 19, 21, 24, 27 et 28.

J'ai regardé avec attention le début 16. Comme le trio en son temps l'a fait, pour simplifier j'ai parfois été amené à regrouper des développements similaires. J'ai conservé leur symbolique : caractères en gras pour les coups forcés, ! et ? pour les meilleurs et les pires coups, et lorsque le détail des coups possibles n'est pas donné, j'ai supposé que l'adversaire jouait à chaque fois son meilleur coup. J'ai dû parfois envisager les débuts de la phase dynamique comme dans la 1<sup>re</sup> partie de cette série, notée 162395, à laquelle j'ai attribué une fin gagnante pour le joueur 1 ce qui n'est pas si évident. La figure ci-dessous montre les trois développements possibles de cette position, lors de la phase dynamique, chacun s'achevant sur une victoire du joueur 1.

Joueur 1	Joueur 2	Joueur 1	Joueur 2	Joueur 1	Joueur 2	Final	Joueur 1	Joueur 2	Joueur 1	Joueur 2	Joueur 1	Joueur 2	Final
1	6	2!	3	9	5	X	1	6	8	2	3	~	Nul
1	6	3	2	4	~	Nul	1	6	8	2	4	7	Nul
1	6	3	2	5	4!	O	1	6	8	2	5	9	Nul
1	6	3	2	5	7 ou 8	Nul	1	6	8	2	7?	9	O
1	6	3	2	5	9?	X	1	6	8	2	9!	7	X
1	6	3	2	7	5	Nul	1	6	8	3?	9	5	X
1	6	3	2	8	5?	X	1	6	8	4?	5	2	X
1	6	3	2	8	4, 7 ou 9	Nul	1	6	8	5!	4	7	O
1	6	3	2	9	5	Nul	1	6	8	7	2!	3	X
1	6	4?	7	2	5 ou 8!	O	1	6	8	7	3	~	Nul
1	6	4?	7	3 ou 5?	8!	O	1	6	8	7	4?	5	O
1	6	4?	7	8 ou 9?	5!	O	1	6	8	7	5!	9	X
1	6	5!	9	3	2	X	1	6	8	7	9?	5	O
1	6	7	4	5	2 ou 8!	Nul	1	6	8	9?	3	~	X
1	6	7	4	5	3 ou 9?	X	1	6	9?	5	4	7	O

FIGURE 3.9 – Les phases statiques commençant par 16, nombreuses et variées, sont toutes représentées ici et évaluées : gagnantes pour le joueur 1 (X), gagnantes pour le joueur 2 (O) ou indécises (NUL).

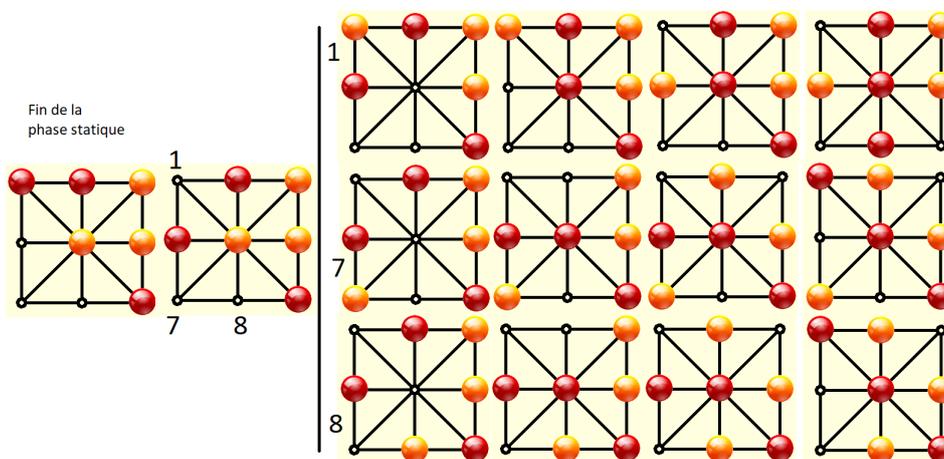


FIGURE 3.10 – La partie 162395 est notée gagnante en fin de la phase statique pour le joueur 1 d'après ces trois développements de la configuration qui le prouvent.

Au vu de ce tableau, on peut retenir l'ouverture 16 qui est gagnante pour le joueur 1 s'il joue son 2<sup>e</sup> pion en 2 ou 5 (les parties 162395 ou 165932) mais perdante s'il le joue en 8 (la partie 168547). Cette analyse est sans doute intéressante pour un joueur humain mais je ne dois pas perdre de vue que mon objectif est de faire gagner la machine. Ayant effectué ce travail épuisant et incertain, quel bénéfice puis-je espérer en tirer pour mon futur programme ? Je ne souhaite pas enregistrer telles qu'elles mes conclusions, potentiellement incomplètes ou partiellement erronées. J'ai une meilleure idée, bien qu'encore assez confuse.

#### d. Algorithme minimax

##### Le principe

J'ai l'idée d'adapter ici l'algorithme *minimax* qui s'applique en théorie des jeux, aux jeux entre deux joueurs à somme nulle (si l'un gagne l'autre perd) et à information complète (les règles et les positions de jeu sont connues à chaque instant). Cet algorithme consiste à maximiser le gain de l'ordinateur ou IA, en supposant que son adversaire humain, jouant le meilleur coup à sa disposition, cherche à le minimiser.

Pour la mise en pratique de cet algorithme, je dois considérer toutes les configurations de jeu et leur attribuer progressivement des codes numériques. L'ensemble de ces configurations forme ce qu'en mathématiques on appelle un graphe. Les configurations sont les sommets de ce graphe, reliés entre eux par les coups autorisés qui en sont les arêtes orientées. Je donne le code 5 aux configurations clairement indévisées (on y reviendra), le code  $-1$  à toutes les configurations impossibles (celles où les deux joueurs ont simultanément réalisé un alignement, celles où un des joueurs a aligné ses pions mais pour qui ce serait encore le tour de jouer). Aux situations de gain de l'ordinateur (il a aligné ses pions et c'est à l'humain de jouer) j'attribue le code 10 tandis qu'aux situations de gain de l'humain (il a aligné ses pions et c'est à l'ordinateur de jouer) j'attribue le code 0.

Le travail d'étiquetage des sommets de ce graphe des configurations peut se poursuivre à rebours, en remontant le fil des parties étiquetées gagnantes ou perdantes : pour chaque configuration étiquetée, je peux faire la liste de ses antécédents, les voisins immédiats qui n'ont pas été étiquetés, et leur attribuer à chacun une valeur, à priori choisie entre 24, 12 et 0 selon que c'est un gain, une indécision ou une perte pour l'ordinateur. La valeur d'un coup de l'ordinateur étant celle qui maximise son profit, si parmi les successeurs, les voisins immédiats qui ont été étiquetés, il s'en trouve un qui offre une victoire le maximum est alors 24 (gain futur ou immédiat) ; si tous offrent une défaite, le maximum est alors 0 (perte future ou immédiate) ; s'il y a un mélange de codes 0 et 12, le maximum est 12 (indécision). L'algorithme doit également étiqueter les configurations où c'est à l'humain de jouer, et pour cela minimiser les profits de l'ordinateur. Si parmi les successeurs d'un coup joué par l'ordinateur, il s'en trouve un qui annonce une perte à celui-ci, le minimum est alors 0 (perte future ou immédiate pour l'ordinateur mais gain pour l'humain) ; si tous annoncent un gain, le minimum est alors 24 (gain futur ou immédiat pour l'ordinateur mais perte pour l'humain) ; s'il y a un mélange de codes 24 et 12, le minimum est 12 (indécision).

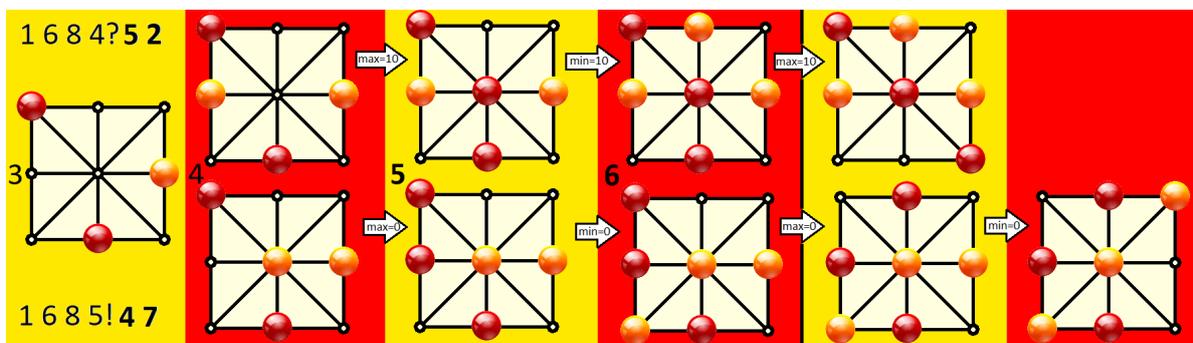


FIGURE 3.11 – Les parties notées 1684?52 et 1685!47 sont notées ainsi, avec les indications 4? et 5! car on peut attribuer rétrospectivement les notes 0 (perte pour l'ordinateur) et 24 (gain pour l'ordinateur) aux deux coups qui suivent l'ouverture 168. Les fonds rouges et jaunes indiquent le joueur dont c'est le tour de jouer.

Pour faire ces attributions, il est clair que je dois disposer de tous les codes des successeurs. Ce rétroplanning n'est donc pas si évident à exécuter car je ne sais pas vraiment identifier les parties indécisées. Les parties nulles sont représentées par le code 12 mais en n'effectuant que les étiquetages mentionnés à la fin de la partie statique, j'ai encore de nombreuses configurations perdantes ou gagnantes qui n'ont pas été identifiées et qui pourraient porter abusivement le code 12. Pour ne citer que les exemples que j'ai illustré, la partie 162395 de la figure 3.10 est gagnante en phase dynamique mais à la fin de la partie statique elle aurait été évaluée indécisée ; de même pour les parties 168452 et 168547 de la figure 3.11 qui sont pourtant, respectivement, gagnante et perdante.

Tout le travail d'analyse et d'évaluation qui a été effectué jusqu'ici est le fruit d'un labeur humain, intelligent certes, parcellaire et suspect à bien des égards. Mon ambition est d'obtenir un résultat plus exhaustif, systématique et irréprochable. Pour cela je vais devoir m'atteler à la programmation d'un algorithme simple, complet et efficace. Je veux qu'il reste compréhensible, proche de mon analyse manuelle, accessible par tout un chacun qui se donnerait la peine de suivre les explications de ce chapitre.

### Codage des configurations

Pour commencer, il me faut adopter un codage des différentes configurations. Pour rester dans la continuité de mon approche manuelle, je vais conserver le codage du plateau avec les chiffres de 1 à 9 disposés comme sur un clavier de smartphone. Pour les emplacements des pions, j'écris les codes des cases où sont placés ceux de l'ordinateur en premier, les codes des cases où sont placés ceux de l'humain et enfin un code indiquant le joueur dont c'est le tour de jouer : 0 si c'est à l'ordinateur de jouer et 1 si c'est à l'humain. Ces trois groupes de codes sont séparés par un tiret afin d'en faciliter la lecture. Petite précision : pour éviter les doublons qui pourraient passer pour des formes originales, je dois m'assurer que les groupes indiquant les emplacements des pions d'un joueur sont codés par le nombre le plus petit. Si, par exemple, les pions de l'ordinateur sont placés aux emplacements 1, 3 et 5, je dois écrire ce groupe 135 et non 153 ou 351. Un dernier détail : au début d'une partie, les pions n'étant pas tous en jeu, certains groupes d'emplacement peuvent n'avoir que deux, un ou même aucun code. Dans le cas d'une absence de code (deux premiers coups) j'ai utilisé le code 0, essentiellement pour des raisons typographiques.

La partie notée 162395 correspond ainsi aux successions de configurations suivantes :

- ♦ Si l'ordinateur a commencé : 0-0-0, 1-0-1, 1-6-0, 12-6-1, 12-36-0, 129-36-1, 129-356-0
- ♦ Si l'humain a commencé : 0-0-1, 0-1-0, 6-1-1, 6-12-0, 36-12-1, 36-129-0, 356-129-1

### Deux types d'enchaînement

La recherche des voisins immédiats, accessibles de façon légale en un coup, dépend du sens de l'enchaînement car notre graphe possède des arêtes orientées. Les successeurs d'une configuration ne sont pas identiques à ses antécédents pour la simple et bonne raison que ce n'est pas le même joueur qui a déplacé un pion. Si une configuration a été atteinte par le déplacement de l'humain, les antécédents de cette configurations sont toutes celles dont l'humain peut partir pour l'atteindre. Dans l'autre sens, les successeurs de cette configurations sont toutes celles que l'ordinateur peut atteindre à partir d'elle. Ces deux types d'enchaînement rappellent ceux qu'on rencontre en généalogie sauf que les antécédents immédiats en généalogie sont toujours au nombre de deux alors qu'ici les antécédents peuvent être bien plus nombreux.

Pour illustrer cela, j'ai construit les deux types d'enchaînement pour deux parties qui partagent le même début. C'est l'ordinateur qui a commencé : partant de 0-0-0 il a laissé le jeu dans la configuration 1-0-1. L'humain ayant joué, la configuration de jeu est passée à 1-6-0. L'ordinateur a alors répliqué en empruntant la configuration 18-6-1. La figure ci-dessous indique les chemins tracés dans le graphe des configurations pour ces deux parties qui sont celles de la figure 3.11 (168452 et 168547) ; à côté de ces chemins, j'ai indiqué les antécédents (lignes du haut) ou les successeurs (lignes du bas) possibles de chaque configuration. Cela donne une idée de la complexité du graphe qu'il nous reste à construire et à étiqueter.

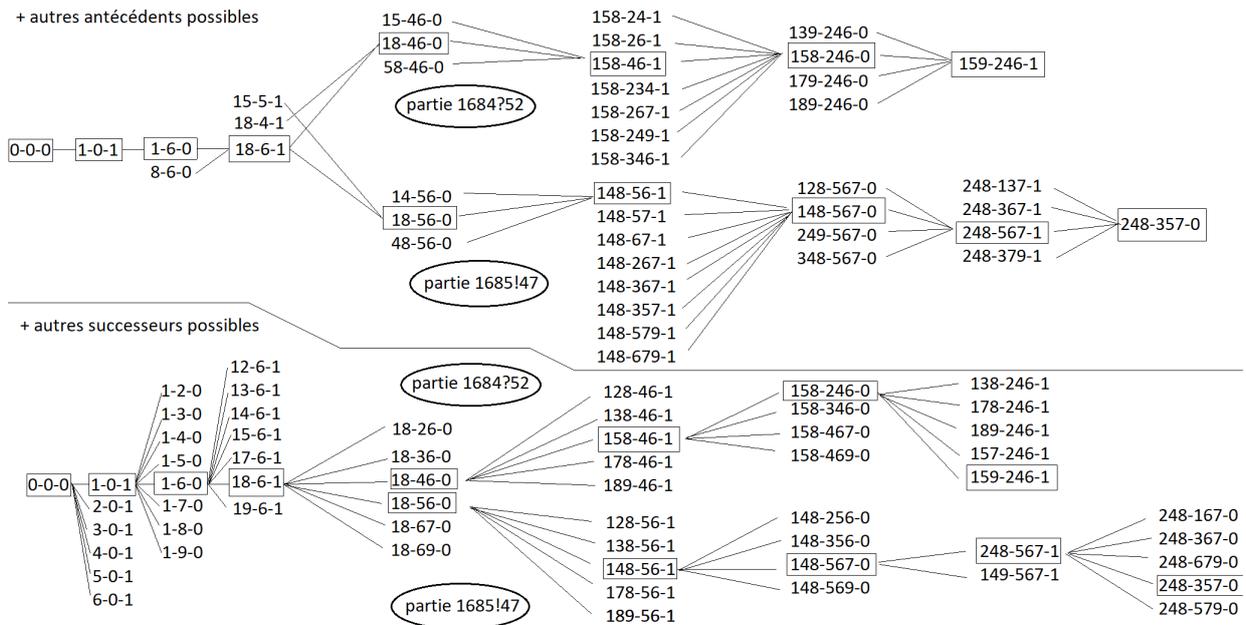


FIGURE 3.12 – Excerpts du graphe des configurations dans lesquels sont encadrés celles qui ont été empruntées par les joueurs pour réaliser le chemin gagnant pour l'ordinateur 168452 et le chemin perdant pour l'ordinateur 168547. Noter que c'est l'ordinateur qui a commencé ces parties (le code 0-0-0 indique cela).

## 2. La phase dynamique

Insensiblement, ma séparation initiale entre les phases statiques et dynamiques était en train de se dissoudre. Certes, le mode de jeu change à la frontière puisqu'avant on pose des pions et qu'après on les déplace mais l'évaluation des différentes configurations suit la même logique de part et d'autre de celle-ci. On devrait plutôt ajouter une frontière entre les 4<sup>e</sup> et 5<sup>e</sup> coup puisqu'à partir du 5<sup>e</sup> il se peut que la partie s'arrête, le joueur qui a commencé gagnant la partie. L'autre frontière naturelle se situe entre les 5<sup>e</sup> et 6<sup>e</sup> coup puisqu'à partir du 6<sup>e</sup> la totalité des pions étant sur le plateau, le reste de la partie reste circonscrite à l'intérieur de l'ensemble de ces configurations à six pions que l'on pourrait appeler *complètes*.

### a. Construction du graphe étiqueté

Pour commencer, je vais construire par programme le graphe non-étiqueté des configurations. Je disposerai ensuite progressivement des étiquettes pour les configurations dont le sort est identifiable. L'idée étant de limiter au maximum l'ensemble des configurations dont le sort est indécis.

#### Le graphe non-étiqueté

Pion par pion, je construis les différentes configurations de ce graphe. Chaque pose d'un pion nécessite de faire la liste, pour une des configurations déjà enregistrées, des emplacements vacants du plateau et d'enregistrer les nouvelles configurations qui s'en déduisent. J'obtiens finalement l'ensemble des configurations envisageables, chacune est dédoublée selon la très importante distinction du joueur dont c'est le tour de jouer. Voici le décompte obtenu à ce stade :

- ♦ 2 parties initiales (l'ordinateur débute 0-0-0 ou l'humain débute 0-0-1)
- ♦ 18 parties avec 1 pion posé (9 par joueur)
- ♦ 144 parties avec 2 pions posés (72 par joueur)
- ♦ 504 parties avec 3 pions posés (252 par joueur)
- ♦ 1512 parties avec 4 pions posés (756 par joueur)
- ♦ 2520 parties avec 5 pions posés (1260 par joueur)
- ♦ 3360 parties avec 6 pions posés (1680 par joueur)

## Premières étiquettes

Au total, les configurations envisageables sont donc au nombre de 8060. Parmi celles-ci, un certain nombre sont impossibles comme il a été dit plus haut (les deux joueurs ont simultanément réalisé un alignement ou bien un des joueurs a aligné ses pions mais c'est à lui de jouer) et un certain nombre sont clairement gagnantes pour l'un ou pour l'autre joueur. Comme il est très facile d'identifier ces deux cas, j'étiquète directement ces configurations :

- ♦ 320 parties sont impossibles
- ♦ 268 parties sont gagnées par l'ordinateur et 268 sont gagnées par l'humain

```

Jeu n°1:0-0-0      Jeu n°163:9-8-0      Jeu n°668:9-78-0      Jeu n°2181:12-345-0 configuration indéci
Jeu n°2:0-0-1      Jeu n°164:9-8-1      Jeu n°669:12-34-0     Jeu n°2182:12-346-0 configuration indéci
Jeu n°3:0-1-0      Jeu n°165:1-23-0     Jeu n°670:12-34-1     Jeu n°2183:12-347-0 configuration indéci
Jeu n°4:0-2-0      Jeu n°166:1-24-0     Jeu n°671:12-35-0     Jeu n°2184:12-348-0 configuration indéci
Jeu n°5:0-3-0      Jeu n°167:1-25-0     Jeu n°672:12-35-1     Jeu n°2185:12-349-0 configuration indéci
Jeu n°6:0-4-0      Jeu n°168:1-26-0     Jeu n°673:12-36-0     Jeu n°2186:12-350-0 configuration indéci
Jeu n°7:0-5-0      Jeu n°169:1-27-0     Jeu n°674:12-36-1     Jeu n°2187:12-351-0 l'humain a gagné
Jeu n°8:0-6-0      Jeu n°170:1-28-0     Jeu n°675:12-37-0     Jeu n°2188:12-352-0 configuration indéci
Jeu n°9:0-7-0      Jeu n°171:1-29-0     Jeu n°676:12-37-1     ****
Jeu n°10:0-8-0     Jeu n°172:1-34-0     Jeu n°677:12-38-0     Jeu n°3723:369-57-1 l'ordinateur a gagné
Jeu n°11:0-9-0     Jeu n°173:1-35-0     Jeu n°678:12-38-1     Jeu n°3724:369-58-1 l'ordinateur a gagné
Jeu n°12:1-0-1     Jeu n°174:1-36-0     Jeu n°679:12-39-0     Jeu n°3725:369-78-1 l'ordinateur a gagné
Jeu n°13:1-2-0     Jeu n°175:1-37-0     Jeu n°680:12-39-1     Jeu n°3726:37-124-0 configuration indéci
Jeu n°14:1-2-1     Jeu n°176:1-38-0     Jeu n°681:12-45-0     Jeu n°3727:37-125-0 configuration indéci
Jeu n°15:1-3-0     Jeu n°177:1-39-0     Jeu n°682:12-45-1     Jeu n°3728:37-126-0 configuration indéci
Jeu n°16:1-3-1     Jeu n°178:1-45-0     Jeu n°683:12-46-0     Jeu n°3729:37-128-0 configuration indéci
Jeu n°17:1-4-0     Jeu n°179:1-46-0     Jeu n°684:12-46-1     Jeu n°3730:37-129-0 configuration indéci
Jeu n°18:1-4-1     Jeu n°180:1-47-0     Jeu n°685:12-47-0     Jeu n°3731:37-145-0 configuration indéci
Jeu n°19:1-5-0     Jeu n°181:1-48-0     Jeu n°686:12-47-1     ****
Jeu n°20:1-5-1     Jeu n°182:1-49-0     ****
Jeu n°21:1-6-0     Jeu n°183:1-56-0     Jeu n°2169:89-45-0     Jeu n°3733:37-148-0 configuration indéci
Jeu n°22:1-6-1     Jeu n°184:1-57-0     Jeu n°2170:89-45-1     Jeu n°3734:37-149-0 configuration indéci
Jeu n°23:1-7-0     Jeu n°185:1-58-0     Jeu n°2171:89-46-0     Jeu n°3735:37-156-0 configuration indéci
Jeu n°24:1-7-1     Jeu n°186:1-59-0     Jeu n°2172:89-46-1     Jeu n°3736:37-158-0 configuration indéci
Jeu n°25:1-8-0     Jeu n°187:1-67-0     Jeu n°2173:89-46-1     l'humain a gagné
Jeu n°26:1-8-1     Jeu n°188:1-68-0     Jeu n°2173:89-47-0     Jeu n°3737:37-159-0 configuration indéci
Jeu n°27:1-9-0     Jeu n°189:1-69-0     Jeu n°2174:89-47-1     Jeu n°3738:37-169-0 configuration indéci
Jeu n°28:1-9-1     Jeu n°190:1-78-0     Jeu n°2175:89-56-0     Jeu n°3739:37-169-0 configuration indéci
Jeu n°29:2-0-1     Jeu n°191:1-79-0     Jeu n°2176:89-56-1     Jeu n°3740:37-189-0 configuration indéci
Jeu n°30:2-1-0     Jeu n°192:1-89-0     Jeu n°2177:89-57-0     Jeu n°3741:37-245-0 configuration indéci
Jeu n°31:2-1-1     Jeu n°193:12-3-1     Jeu n°2178:89-57-1     Jeu n°3742:37-246-0 configuration indéci
Jeu n°32:2-3-0     Jeu n°194:12-4-1     Jeu n°2179:89-67-0     Jeu n°3743:37-248-0 configuration indéci
Jeu n°33:2-3-1     Jeu n°195:12-5-1     Jeu n°2180:89-67-1     Jeu n°3744:37-249-0 configuration indéci
****              ****              ****              ****
Jeu n°6739:346-789-0 l'humain a gagné
Jeu n°6740:346-789-1 configuration impossible
Jeu n°6741:347-125-0 configuration indéci
Jeu n°6742:347-125-1 configuration indéci
Jeu n°6743:347-126-0 configuration indéci
Jeu n°6744:347-126-1 configuration indéci
Jeu n°6745:347-128-0 configuration indéci
Jeu n°6746:347-128-1 configuration indéci
Jeu n°6747:347-129-0 configuration indéci
Jeu n°6748:347-129-1 configuration indéci
Jeu n°6749:347-156-0 configuration indéci
Jeu n°6750:347-156-1 configuration indéci
Jeu n°6751:347-158-0 configuration indéci
Jeu n°6752:347-158-1 configuration indéci
Jeu n°6753:347-159-0 l'humain a gagné
Jeu n°6754:347-159-1 configuration impossible
Jeu n°6755:347-168-0 configuration indéci
Jeu n°6756:347-168-1 configuration indéci
Jeu n°6757:347-169-0 configuration indéci
Jeu n°6758:347-169-1 configuration indéci
Jeu n°6759:347-189-0 configuration indéci
Jeu n°6760:347-189-1 configuration indéci
Jeu n°6761:347-256-0 configuration indéci
Jeu n°6762:347-256-1 configuration indéci
Jeu n°6763:347-258-0 l'humain a gagné
****
Jeu n°8053:789-345-0 configuration impossible
Jeu n°8054:789-345-1 l'ordinateur a gagné
Jeu n°8055:789-346-0 configuration impossible
Jeu n°8056:789-346-1 l'ordinateur a gagné
Jeu n°8057:789-356-0 configuration impossible
Jeu n°8058:789-356-1 l'ordinateur a gagné
Jeu n°8059:789-456-0 configuration impossible
Jeu n°8060:789-456-1 configuration impossible

```

FIGURE 3.13 – Extraits du graphe des configurations qui comporte 8060 configurations au total dont 320 sont impossibles et 536 sont gagnantes pour l'un ou l'autre des joueurs (les étiquettes numériques ont été remplacées par des textes plus explicites).

## Secondes étiquettes

Il reste alors 7204 configurations à étiqueter. En passant celles-ci en revue, je construis pour chacune la liste des emplacements qui lui succède et si, parmi ces successeurs, il s'en trouve qui ont reçu une étiquette, je leur applique le principe de l'algorithme minimax, à savoir :

- ♦ si le joueur est l'ordinateur, je lui attribue l'étiquette 23 si un des successeurs a reçu l'étiquette 24 et sinon l'étiquette 1 si un des successeurs a reçu l'étiquette 0 et qu'aucun des successeurs n'est une configuration indéci.
- ♦ si le joueur est l'humain, je lui attribue l'étiquette 1 si un des successeurs a reçu l'étiquette 0 et sinon l'étiquette 23 si un des successeurs a reçu l'étiquette 24 et qu'aucun des successeurs n'est une configuration indéci.

Ce 2<sup>e</sup> étiquetage conduit à 1332 parties gagnantes au prochain coup pour l'ordinateur (configurations étiquetées 9) et autant pour l'humain (configurations étiquetées 1), si le joueur joue son meilleur coup disponible ce qui est l'hypothèse de base de l'algorithme. Ces configurations gagnantes au prochain coup se répartissent en 528 configurations complètes (contenant les six pions), 444 contenant cinq pions et 360 n'en contenant que quatre.

Pourquoi diminuer de 1 la notation d'un coup gagnant pour l'ordinateur ou augmenter de 1 celle d'un coup gagnant pour l'humain ? Ce n'est pas ce que j'avais prévu et annoncé plus haut mais de cette façon j'identifie mieux la proximité d'une victoire et surtout, je peux isoler les configurations qui viennent d'être étiquetées. Pour le traitement, cela simplifie mon algorithme et pour l'information apportée aux configuration, cela l'enrichit.

12	45	14	23	17	23	23	45	28	13	36	12	39	12	46	12	56	12	58	13	69	12
12	46	14	25	17	25	23	46	28	14	36	14	39	14	46	13	56	13	58	14	69	14
12	47	14	26	17	26	23	47	28	16	36	15	39	15	46	17	56	17	58	16	69	15
12	48	14	28	17	28	23	48	28	17	36	17	39	17	46	18	56	18	58	17	69	17
12	49	14	29	17	29	23	49	28	19	36	18	39	18	46	19	56	19	58	19	69	18
12	56	14	35	17	35	23	56	28	34	36	24	39	24	46	23	56	23	58	34	69	24
12	457	14	235	17	235	23	457	28	134	36	124	39	124	46	127	56	127	58	134	69	124
12	458	14	236	17	236	23	458	28	136	36	125	39	125	46	128	56	128	58	136	69	125
12	459	14	238	17	238	23	459	28	137	36	127	39	127	46	129	56	129	58	137	69	127
12	467	14	239	17	239	23	467	28	139	36	128	39	128	46	137	56	137	58	139	69	128
12	468	14	256	17	256	23	468	28	146	36	145	39	145	46	138	56	138	58	146	69	145
12	469	14	259	17	259	23	469	28	149	36	148	39	148	46	139	56	139	58	149	69	148
125	467	145	236	169	234	246	137	269	145	358	124	459	127	135	467	156	234	234	679	257	349
125	468	145	238	169	237	246	138	269	148	358	126	459	128	135	468	156	237	234	689	257	468
125	469	145	239	169	238	246	139	269	157	358	129	459	137	135	469	156	238	235	467	257	469
125	478	145	268	169	247	246	178	269	158	358	146	459	138	135	478	156	239	235	468	257	489
125	479	145	269	169	248	246	179	269	178	358	149	459	178	135	479	156	247	235	469	257	689
125	489	145	289	169	278	246	189	269	457	358	167	459	236	135	489	156	248	235	478	259	134

FIGURE 3.14 – 198 configurations parmi les 1332 configurations gagnantes en un coup pour l'ordinateur. Le premier groupe de chiffres sont les emplacements de l'ordinateur, le second groupe celui de l'humain et c'est à l'ordinateur de jouer.

### Étiquettes suivantes

Il reste encore 4540 configurations à étiqueter. De la même façon que pour mettre la 2<sup>e</sup> étiquette, on considère les configurations non encore étiquetées (code 11) et on dresse pour chacune la liste des configurations voisines qui lui succède. La meilleure étiquette si c'est l'ordinateur qui joue est alors la plus grande diminuée de 1 si la plus grande est présente (si on a l'étiquette 23 on choisit 22) et sinon la plus petite augmentée de 1 s'il aucun des successeurs n'est indécis et que la plus petite est présente. Si c'est l'humain qui joue il suffit d'inverser ces règles : la meilleure étiquette est la plus petite augmentée de 1 si la plus petite est présente (si on a l'étiquette 1 on choisit 2) et sinon la plus grande diminuée de 1 s'il aucun des successeurs n'est indécis et que la plus grande est présente. On a ainsi simulé le meilleur coup de chacun des joueurs. La même règle sera par la suite adoptée pour attribuer les étiquettes des niveaux supérieurs (les séries d'étiquettes de la 4<sup>e</sup> à la 12<sup>e</sup>).

125	478	156	238	158	46	257	148	269	178	358	129	457	689	567	149
125	479	156	239	158	47	257	149	269	478	358	146	459	127	567	189
126	478	156	247	158	467	257	169	278	145	358	149	459	128	567	28
126	479	156	278	158	469	257	189	289	356	358	169	459	178	567	29
128	457	156	289	158	479	257	346	345	12	358	269	459	27	567	238
129	356	156	37	158	67	257	46	345	18	358	46	459	28	567	239
135	467	156	38	158	679	257	49	345	19	358	49	459	237	567	289
135	468	156	347	169	235	257	469	345	126	358	467	459	238	567	389
135	469	156	378	179	458	257	489	345	127	358	469	459	278	567	489
135	478	156	389	189	457	259	134	345	128	358	479	459	37	567	89
135	479	157	236	234	679	259	136	345	129	358	69	459	38	569	178
135	679	157	238	234	689	259	146	345	169	358	679	459	367	569	478
135	689	157	239	235	679	259	34	345	178	359	124	459	368	578	124
137	245	157	268	235	689	259	36	345	189	359	127	459	378	578	134
139	256	157	289	237	145	259	37	345	28	359	128	459	678	579	124
145	236	157	389	238	569	259	346	345	29	359	178	459	78	579	134
145	239	157	689	246	135	259	347	345	269	359	248	467	589	579	136
146	235	158	234	247	389	259	367	345	278	359	278	468	579	579	146
148	236	158	237	247	689	259	368	345	289	359	478	469	578	579	236
148	239	158	247	248	157	259	378	346	125	367	589	489	136	579	246
149	578	158	34	257	14	259	46	347	125	368	124	489	236	579	346
156	23	158	37	257	16	259	467	356	124	368	127	567	18	589	136
156	27	158	346	257	19	259	67	356	127	378	569	567	19	589	236
156	28	158	347	257	134	259	678	358	16	379	568	567	128	678	124
156	234	158	367	257	136	268	359	358	19	457	389	567	129	678	134
156	237			257	146			358	126			567	148		

FIGURE 3.15 – Les 204 configurations configurations gagnantes en deux coups pour l'ordinateur. Ici, c'est à l'humain de jouer.

La 3<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à deux coups d'une victoire. Ces étiquettes sont attribuées à 204 configurations où c'est à l'humain de jouer et, bien sûr, autant pour lesquelles c'est le tour de l'ordinateur. Parmi les 204 gagnantes, 40 correspondent à des plateaux contenant cinq pions seulement, les 164 configurations restantes étant complètes. Il reste encore, à l'issue de ce 3<sup>e</sup> étiquetage, 4132 configurations à étiqueter.

La 4<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à trois coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 476 configurations où c'est à lui de jouer et il y en a autant pour les victoires de l'humain. Parmi ces 476 positions gagnantes, 40 correspondent à des plateaux contenant quatre pions seulement, 224 correspondent à des plateaux contenant cinq pions, les 212 configurations restantes étant complètes. Il reste encore, à l'issue de ce 4<sup>e</sup> étiquetage, 3180 configurations à étiqueter.

12	356	19	256	29	37	38	46	49	278	67	289	148	367	245	389	345	679	479	236
13	245	19	356	29	46	38	49	49	367	67	389	148	679	245	679	346	589	479	238
13	256	19	457	29	67	38	69	49	368	67	489	149	235	245	689	347	589	479	368
14	578	19	458	29	134	38	124	49	378	67	589	156	479	247	136	348	125	489	127
15	239	19	578	29	136	38	126	49	578	68	124	157	234	247	169	348	126	489	237
15	289	23	145	29	146	38	127	49	678	68	127	157	346	247	189	348	127	489	367
15	389	24	135	29	178	38	129	56	124	68	134	157	469	249	135	348	129	567	134
15	469	24	157	29	346	38	146	56	148	68	359	157	489	249	136	348	157	568	124
15	479	24	389	29	347	38	149	56	149	68	579	158	239	249	157	348	169	568	127
15	679	24	679	29	356	38	169	56	234	69	235	167	234	249	178	348	269	568	129
15	689	24	689	29	367	38	269	56	247	78	569	167	235	249	356	348	569	568	134
16	23	25	148	29	368	38	467	56	347	79	458	167	238	249	367	348	579	568	149
16	27	25	189	29	378	38	469	56	478	79	568	167	239	249	368	348	679	568	239
16	28	25	368	29	467	38	479	56	489	89	457	167	245	249	378	349	125	568	479
16	37	25	378	29	478	38	569	57	134	126	347	167	289	249	578	349	126	579	128
16	38	25	478	29	678	38	679	57	136	126	378	167	389	249	678	349	127	579	148
16	234	25	489	34	12	39	256	57	236	126	389	167	458	256	134	349	128	579	238
16	235	25	678	34	18	39	568	57	238	127	346	167	489	256	178	349	178	579	368
16	237	25	689	34	19	45	126	57	239	127	389	167	589	256	347	349	256	678	129
16	238	26	135	34	28	45	169	57	346	127	469	168	234	256	378	349	278	678	149
16	239	26	178	34	29	45	236	57	389	127	489	168	235	256	389	349	568	678	239
16	247	26	359	34	125	45	269	58	124	127	689	168	237	256	478	349	578	679	124
16	278	26	478	34	126	45	367	58	126	128	356	168	239	256	479	349	678	679	128
16	289	26	479	34	127	45	368	58	129	129	457	168	247	257	389	358	127	679	134
16	347	27	14	34	128	45	678	58	234	134	269	168	347	259	178	359	126	679	148
16	378	27	16	34	129	45	689	58	236	134	278	168	359	267	134	359	146	679	238
16	389	27	19	34	169	46	125	58	237	134	289	168	457	267	135	359	467		
16	478	27	46	34	178	46	135	58	247	134	679	168	479	267	145	359	678		
16	479	27	49	34	189	46	235	58	269	134	689	168	579	267	148	367	125		
17	245	27	134	34	269	46	578	59	124	135	247	169	578	267	149	368	129		
17	458	27	136	34	278	46	579	59	127	135	269	178	234	267	189	368	149		
18	34	27	145	34	289	46	589	59	128	135	278	178	236	267	359	368	479		
18	37	27	146	34	679	47	125	59	134	135	289	178	239	267	389	378	145		
18	46	27	148	34	689	48	136	59	136	136	247	178	346	267	489	389	124		
18	47	27	149	35	127	48	157	59	146	136	278	178	469	267	589	389	126		
18	67	27	169	35	178	48	236	59	178	136	289	189	356	269	134	389	127		
18	234	27	189	35	278	48	239	67	18	136	478	234	169	269	347	389	146		
18	236	27	346	35	467	48	579	67	19	136	479	234	178	269	378	389	467		
18	237	27	389	35	478	49	27	67	28	138	245	234	189	278	569	458	127		
18	239	27	469	35	479	49	28	67	29	138	247	237	569	279	134	458	136		
18	247	27	489	35	679	49	37	67	89	138	256	238	145	279	136	458	236		
18	346	27	689	36	589	49	38	67	124	138	269	239	146	279	145	458	237		
18	347	28	145	37	125	49	78	67	128	138	457	239	178	279	146	458	239		
18	367	28	157	37	145	49	127	67	129	138	467	239	467	279	148	458	367		
18	457	28	356	37	245	49	128	67	134	138	469	239	478	279	346	458	679		
18	467	28	359	37	568	49	136	67	148	138	479	239	678	279	356	459	136		
18	469	28	457	37	569	49	178	67	149	138	569	245	136	279	368	467	125		
18	479	28	569	37	589	49	236	67	189	138	679	245	169	279	458	469	235		
18	679	29	34	38	16	49	237	67	238	146	578	245	178	279	568	479	128		
19	235	29	36	38	19	49	238	67	239	148	237	245	189	289	457	479	136		

FIGURE 3.16 – Les 476 configurations gagnantes en trois coups pour l'ordinateur. Ici, c'est à l'ordinateur de jouer.

La 5<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à quatre coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 148 configurations où c'est

à l'humain de jouer et il y en a autant pour les victoires de l'humain. Parmi ces 148 positions gagnantes, 64 correspondent à des plateaux contenant quatre pions seulement, 48 correspondent à des plateaux contenant cinq pions, les 36 configurations restantes étant complètes. Il reste encore, à l'issue de ce 5<sup>e</sup> étiquetage, 2884 configurations à étiqueter.

15	23	18	47	59	24	49	78	169	78	148	235	579	14	489	356
16	23	29	36	35	49	59	78	167	89	378	14	138	457	269	578
27	14	19	47	57	28	57	89	267	14	158	236	249	356	267	589
19	23	19	48	57	29	67	89	249	36	167	235	579	36	348	569
15	28	57	14	59	27	148	23	279	14	168	235	168	457	349	578
25	19	57	16	59	28	149	23	234	69	267	145	156	478	245	689
19	26	25	49	25	67	126	47	237	69	345	69	279	356	257	689
34	12	45	29	58	34	129	47	279	36	348	69	567	124	345	689
35	12	56	18	56	37	157	23	249	78	279	145	568	124		
37	12	58	16	59	34	158	23	247	89	349	78	678	14		
37	14	56	19	58	37	135	47	259	78	347	89	458	236		
15	38	58	19	59	36	138	47	257	89	359	78	459	236		
35	18	35	46	19	78	167	23	269	78	348	125	138	569		
19	36	15	67	57	46	168	23	267	89	349	125	256	478		
15	46	45	37	35	69	156	47	348	12	358	124	259	478		
37	24	15	68	37	68	135	69	349	12	368	125	167	589		
15	47	35	48	59	46	138	69	358	12	459	36	234	569		
25	37	45	38	37	69	168	47	359	12	489	36	678	145		
35	28	56	27	38	69	189	36	367	12	567	14	249	578		
45	19	57	26	37	89	157	89	368	12	126	457	247	589		

FIGURE 3.17 – Les 148 configurations gagnantes en quatre coups pour l'ordinateur. Ici, c'est à l'humain de jouer.

La 6<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à cinq coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 240 configurations où c'est à l'ordinateur de jouer et il y en a autant pour les victoires de l'humain. Parmi ces 240 positions gagnantes, 92 correspondent à des plateaux contenant trois pions seulement, 56 correspondent à des plateaux contenant quatre pions, 32 configurations contenant cinq pions et les 60 configurations restantes complètes. Il reste encore, à l'issue de ce 6<sup>e</sup> étiquetage, 2404 configurations à étiqueter.

1	23	0	4	29	0	6	23	0	9	48	0	26	578	0	48	125	0	124	689	0	246	589	0
1	26	0	4	37	0	6	27	0	9	78	0	26	589	0	48	235	0	127	589	0	248	356	0
1	28	0	4	38	0	6	37	0	15	69	0	27	36	0	48	356	0	128	457	0	248	569	0
1	36	0	4	78	0	6	89	0	15	89	0	27	69	0	48	569	0	128	569	0	268	145	0
1	38	0	5	12	0	7	12	0	16	47	0	27	89	0	49	12	0	129	356	0	268	457	0
1	46	0	5	14	0	7	14	0	16	78	0	27	356	0	49	23	0	129	578	0	278	145	0
1	47	0	5	16	0	7	16	0	16	89	0	27	589	0	49	36	0	134	569	0	278	356	0
1	48	0	5	18	0	7	24	0	16	457	0	29	14	0	49	125	0	136	457	0	289	145	0
1	67	0	5	19	0	7	26	0	16	589	0	29	47	0	49	356	0	137	569	0	289	356	0
1	68	0	5	23	0	7	28	0	18	23	0	29	78	0	56	14	0	137	589	0	346	125	0
1	78	0	5	24	0	7	29	0	18	36	0	29	145	0	56	47	0	139	457	0	346	578	0
2	14	0	5	26	0	7	46	0	18	69	0	29	578	0	57	23	0	139	578	0	347	125	0
2	19	0	5	27	0	7	68	0	18	235	0	34	69	0	57	36	0	146	235	0	347	569	0
2	36	0	5	28	0	7	69	0	18	569	0	34	78	0	58	12	0	146	589	0	367	145	0
2	37	0	5	29	0	7	89	0	24	36	0	34	89	0	58	23	0	149	356	0	367	589	0
2	49	0	5	34	0	8	16	0	24	69	0	34	569	0	59	12	0	149	578	0	378	125	0
2	67	0	5	36	0	8	19	0	24	78	0	34	578	0	59	14	0	169	235	0	378	569	0
3	12	0	5	37	0	8	34	0	24	89	0	35	47	0	67	12	0	169	457	0	379	125	0
3	14	0	5	38	0	8	37	0	24	356	0	35	78	0	67	14	0	178	235	0	379	145	0
3	18	0	5	46	0	8	47	0	24	569	0	38	12	0	67	23	0	179	235	0	389	125	0
3	24	0	5	47	0	8	69	0	24	578	0	38	14	0	67	145	0	179	356	0	467	235	0
3	28	0	5	48	0	9	23	0	24	589	0	38	47	0	67	235	0	189	235	0	467	589	0
3	46	0	5	49	0	9	24	0	25	78	0	38	125	0	68	12	0	189	457	0	468	125	0
3	48	0	5	67	0	9	26	0	25	89	0	38	457	0	68	14	0	236	478	0	468	235	0
3	49	0	5	68	0	9	27	0	26	14	0	45	36	0	68	23	0	237	145	0	469	125	0
3	68	0	5	69	0	9	28	0	26	47	0	45	69	0	68	47	0	237	589	0	469	578	0
3	69	0	5	78	0	9	34	0	26	78	0	48	12	0	68	125	0	238	457	0	478	236	0
3	89	0	5	89	0	9	36	0	26	89	0	48	23	0	68	145	0	238	569	0	479	356	0
4	12	0	6	18	0	9	46	0	26	145	0	48	36	0	68	235	0	239	578	0	679	145	0
4	19	0	6	19	0	9	47	0	26	457	0	48	69	0	68	457	0	246	578	0	689	124	0

FIGURE 3.18 – Les 240 configurations gagnantes en cinq coups pour l'ordinateur. Ici, c'est à l'ordinateur de jouer.

La 7<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à six coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 80 configurations où c'est à l'humain de jouer et il y en a autant pour les victoires de l'humain. Parmi ces 80 positions gagnantes, 4 correspondent à des plateaux contenant deux pions seulement, 16 correspondent à des plateaux contenant trois pions, 8 correspondent à des plateaux contenant quatre pions, 40 configurations contenant cinq pions et les 12 configurations restantes complètes. Il reste encore, à l'issue de ce 7<sup>e</sup> étiquetage, 2244 configurations à étiqueter.

La 8<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à sept coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 72 configurations où c'est à l'ordinateur de jouer et il y en a autant pour les victoires de l'humain. Parmi ces 72 positions gagnantes, 4 correspondent à des plateaux contenant un seul pion, 24 correspondent à des plateaux en contenant deux, 8 correspondent à des plateaux contenant trois pions, 8 correspondent à des plateaux contenant quatre pions, 8 configurations contenant cinq pions et les 20 configurations restantes complètes. Il reste encore, à l'issue de ce 8<sup>e</sup> étiquetage, 2100 configurations à étiqueter.

La 9<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à huit coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 65 configurations où c'est à l'humain de jouer et il y en a autant pour les victoires de l'humain. Parmi ces 65 positions gagnantes, 1 correspond à un plateau contenant un seul pion, 8 correspondent à des plateaux en contenant trois, 12 correspondent à des plateaux contenant cinq pions et les 44 configurations restantes sont complètes. Il reste encore, à l'issue de ce 9<sup>e</sup> étiquetage, 1970 configurations à étiqueter.

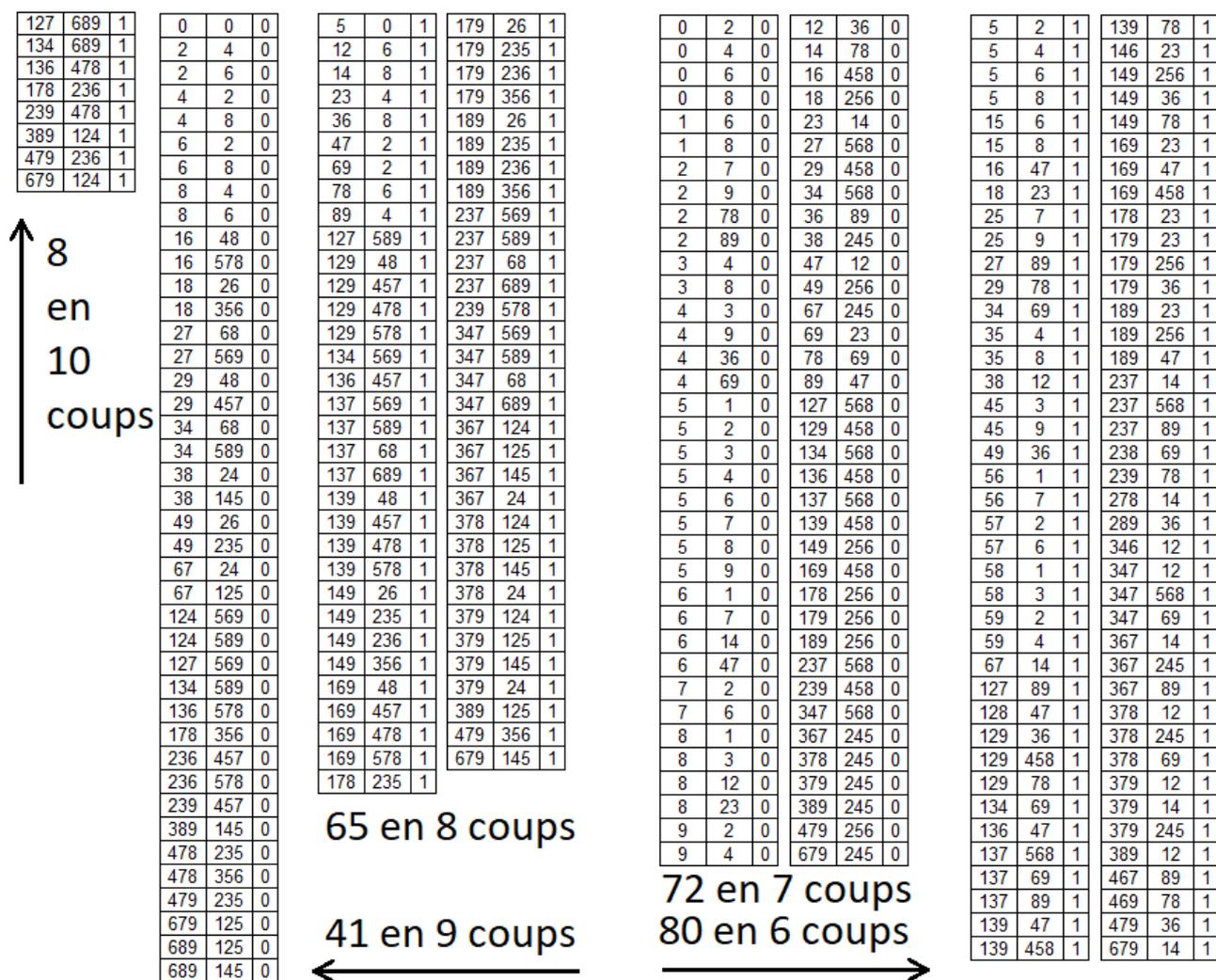


FIGURE 3.19 – Les 8, 41, 65, 72 et 80 configurations configurations gagnantes en 10, 9, 8, 7 et 6 coups pour l'ordinateur. La 3<sup>e</sup> colonne indique le joueur dont c'est le tour de jouer (0 : l'ordinateur ; 1 : l'humain).

La 10<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à neuf coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 41 configurations où c'est à l'ordinateur de jouer et il y en a autant pour les victoires de l'humain. Parmi ces 41 positions gagnantes, 1 correspond au plateau vide, 8 correspondent à des plateaux contenant deux pions seulement, 8 correspondent à des plateaux contenant quatre pions, 8 correspondent à des plateaux contenant cinq pions et les 16 configurations restantes sont complètes. Il reste encore, à l'issue de ce 10<sup>e</sup> étiquetage, 1888 configurations à étiqueter.

La 11<sup>e</sup> série d'étiquettes gagnantes correspond à des configurations qui sont à dix coups d'une victoire. Pour les victoires de l'ordinateur, ces étiquettes sont attribuées à 8 configurations où c'est à l'humain de jouer et il y en a autant pour les victoires de l'humain, toutes concernent des configurations complètes. Il s'agit de la dernière série d'étiquettes attribuables car l'algorithme, lancé sur l'analyse de la 12<sup>e</sup> série ne trouve aucune configuration gagnante. Il reste finalement, à l'issue de ce 11<sup>e</sup> et dernier étiquetage, 1872 configurations sans étiquette.

Lorsque l'adversaire humain de l'ordinateur laisse à celui-ci une configuration quelconque, il sera désormais facile de choisir le meilleur coup à jouer : ce sera toujours, parmi les différents successeurs possibles, celui qui maximise la valeur de l'étiquette. Pour cette raison l'étiquette d'une configuration indécise doit obligatoirement se situer entre la valeur la plus haute de celles des configurations gagnantes pour l'humain et la valeur la plus basse de celles des configurations gagnantes pour l'ordinateur. Comme il y a onze valeurs gagnantes pour l'humain (de 0 à 10) et onze gagnantes pour l'ordinateur (de 14 à 24), les valeurs 11 et 13 étant techniquement réservées aux configurations gagnantes en onze coups, il est logique d'attribuer *in fine* la valeur 12 aux 1872 étiquettes d'indécision. Ces configurations indécises – on pourrait dire équilibrées – il s'en trouve 272 par joueur à comporter les six pions, soit 544 en tout ce qui est assez peu finalement (16% des configurations complètes). Ce sont les configurations à surveiller lors de la phase dynamique pour le cas où les joueurs auraient décidé d'empêcher les parties d'être nulles en interdisant le retour d'une même configuration.

### Bilan sur le graphe étiqueté

Avant d'aller plus loin dans la construction de mon algorithme de jeu, je veux revenir sur les résultats de cette laborieuse étape d'étiquetage. Dans ce but, j'ai dressé un tableau de répartition en croisant la distance à un gain et le nombre de pions en jeu. On peut retrouver dans ce tableau bien des aspects évoqués précédemment, comme par exemple l'ouverture au centre qui assure de la victoire en neuf coups (figure 3.6) ou bien encore la réponse au centre lorsque l'adversaire à ouvert sur un côté qui assure de la victoire en sept coups (figure 3.7).

	0 pion	1 pion	2 pions	3 pions	4 pions	5 pions	6 pions	total	×2
impossibles	0	0	0	0	0	0	160	160	320
0 coup	0	0	0	0	0	120	148	268	536
1 coup	0	0	0	0	360	444	528	1332	2664
2 coups	0	0	0	0	0	40	164	204	408
3 coups	0	0	0	0	40	224	212	476	952
4 coups	0	0	0	0	64	48	36	148	296
5 coups	0	0	0	92	56	32	60	240	480
6 coups	0	0	4	16	8	40	12	80	160
7 coups	0	4	24	8	8	8	20	72	144
8 coups	0	1	0	8	0	12	44	65	130
9 coups	1	0	8	0	8	8	16	41	82
10 coups	0	0	0	0	0	0	8	8	16
indécises	0	4	36	128	212	284	272	936	1872
total	1	9	72	252	756	1260	1680	4030	8060
×2	2	18	144	504	1512	2520	3360	8060	

FIGURE 3.20 – Distribution des étiquettes aux différentes configurations du graphe de jeu selon deux paramètres : la distance qui la sépare d'une victoire (si les deux joueurs jouent leurs meilleurs coups) et le nombre de pions en jeu sur le plateau. Les nombres donnés correspondent au gain d'un des joueurs, les cases grisées indiquant celles où c'est son tour de jouer. Les totaux sont doublés pour tenir compte des deux joueurs.

## b. Une première application

La réalisation du jeu avec son intelligence artificielle constitue mon objectif final mais je veux d'abord tester mes résultats sur une application plus simple : je veux pouvoir choisir une position quelconque de départ et déterminer pour celle-ci une partie qui se déduit directement des étiquettes attribuées aux configurations du graphe. Mon premier choix se porte sur une des huit configurations complète gagnante en dix coups. Son existence m'est signalée par le tableau. La configuration 389.124.1.14 en est une dont le tour est au joueur 1 (l'humain) ; ses pions sont aux emplacements 1, 2 et 4 ; ceux de l'ordinateur sont aux emplacements 3, 8 et 9 et l'étiquette 14 indique que l'ordinateur gagne dans 10 coups.

Je construis un algorithme qui détermine le successeur de la configuration choisie : son code ressemble à 389.xxx.0.15 puisque l'humain aura joué un de ses pions. Il y a plusieurs configurations possibles pour les pions de ce joueur (125, 127, 145 ou 245) mais mon algorithme en choisira un, au hasard, parmi ceux d'étiquette 15. En fait il n'y en a qu'un ici (389.345.0.15), les autres assurant une victoire de l'ordinateur encore plus rapide (en 3 coups pour 127, 5 pour 125 et 7 pour 245). Je passe sur les détails technique de la réalisation mais il m'est facile d'effectuer ce choix à chaque étape de la partie en effectuant les recherches ciblées *ad hoc* dans mon graphe étiqueté.

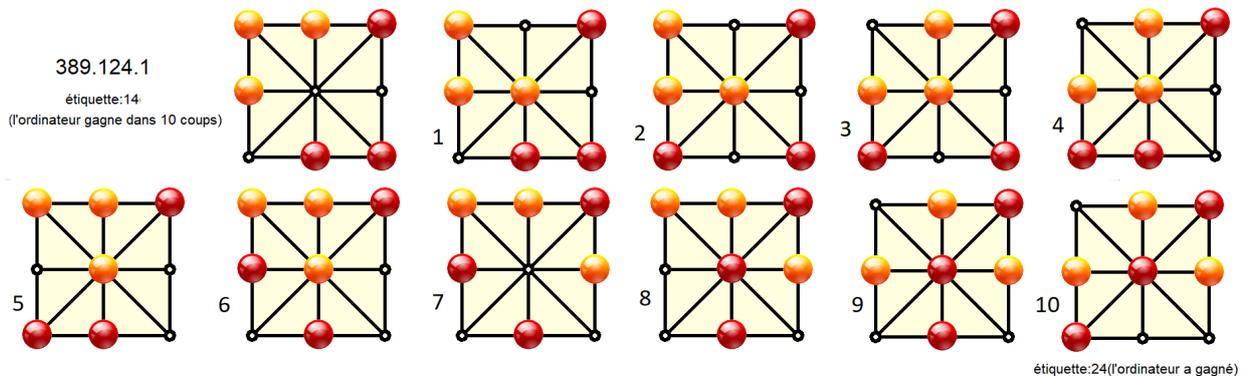


FIGURE 3.21 – Une partie qui se gagne en dix coups et qui ne concerne qu'une seule configuration complète. Il en existe huit variantes, celles-ci est gagnante pour l'ordinateur (en rouge) mais c'est à l'humain de jouer.

Le résultat de cette programmation intermédiaire est très satisfaisant car je peux ainsi découvrir des techniques pour mieux jouer à ce jeu. Je m'aperçois ici que c'est exactement la même partie qui se décline par rotation et symétrie de huit façons différentes pour un gain de l'ordinateur et d'autant de façons pour un gain de l'humain. En procédant de même, on peut identifier d'autres techniques permettant de gagner de façon plus efficace : explorant mon tableau synthétique, je m'aperçois qu'on peut gagner en neuf coups à partir d'une position à deux pions, 4.8.0.15 est une de celles-là. J'utilise alors mon application et visualise la succession des coups, espérant pouvoir m'en inspirer lors d'une prochaine partie en famille.

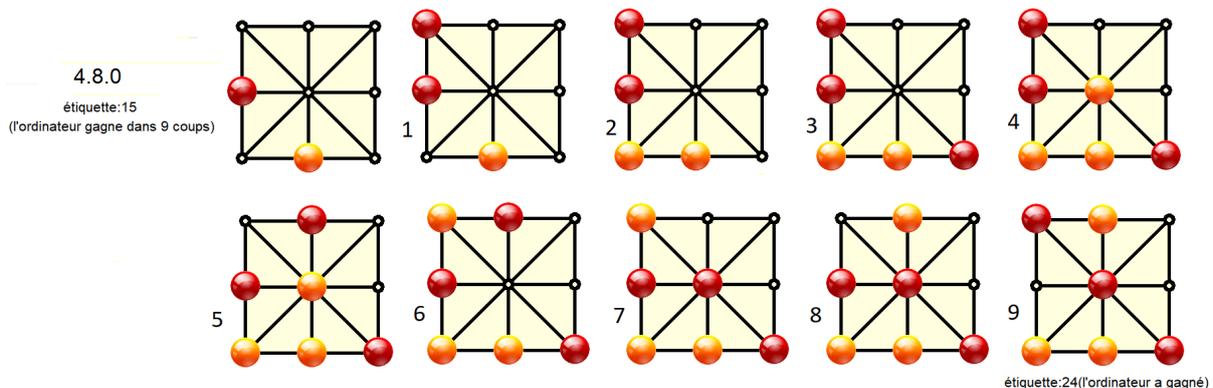


FIGURE 3.22 – Une partie qui se gagne en neuf coups et qui ne concerne qu'une seule configuration à deux pions. Il en existe huit variantes, celles-ci est gagnante pour l'ordinateur et c'est à son tour de jouer.

Je ne résiste pas à une exploration plus poussée de ces stratégies gagnantes pas si faciles à découvrir sans l'aide apportée par ce graphe étiqueté. Je retrouve une des parties victorieuses décrites dans la 1<sup>re</sup> ligne à gauche de la figure 3.7 lorsque, après une ouverture sur le côté, le joueur répond au centre. La victoire est alors assurée en sept coups. Le début de cette partie y est noté 45!29!17, ce qui équivaut à tourner d'un quart de tour celle donnée par l'image ci-dessous.

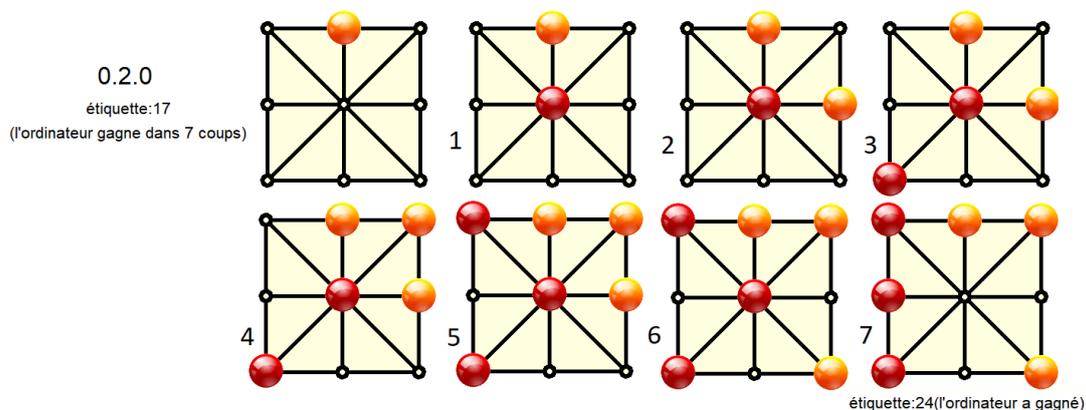


FIGURE 3.23 – Une partie qui se gagne en sept coups et qui ne concerne qu'une seule configuration à un pion. Il en existe quatre variantes, celles-ci est gagnante pour l'ordinateur et c'est à son tour de jouer.

Tous les tests effectués me confirment dans la justesse des résultats. J'en déduis que cette étape étant validée, je vais pouvoir utiliser mon graphe étiqueté pour réaliser le jeu entre l'ordinateur et un joueur humain qui ne répond pas forcément le meilleur coup disponible ce qui, en principe devrait écourter la victoire de l'ordinateur dans la plupart des cas. Il restera toujours les cas où, dès le début le jeu est dans une configuration favorable pour l'humain. Dans ces cas, en jouant bien, l'humain peut conserver son avantage jusqu'à la victoire. De même, si le jeu est indécis depuis le début, l'humain peut espérer finir la partie à égalité avec l'ordinateur, cependant l'algorithme exploitera toutes les erreurs de l'humain, s'il en fait.

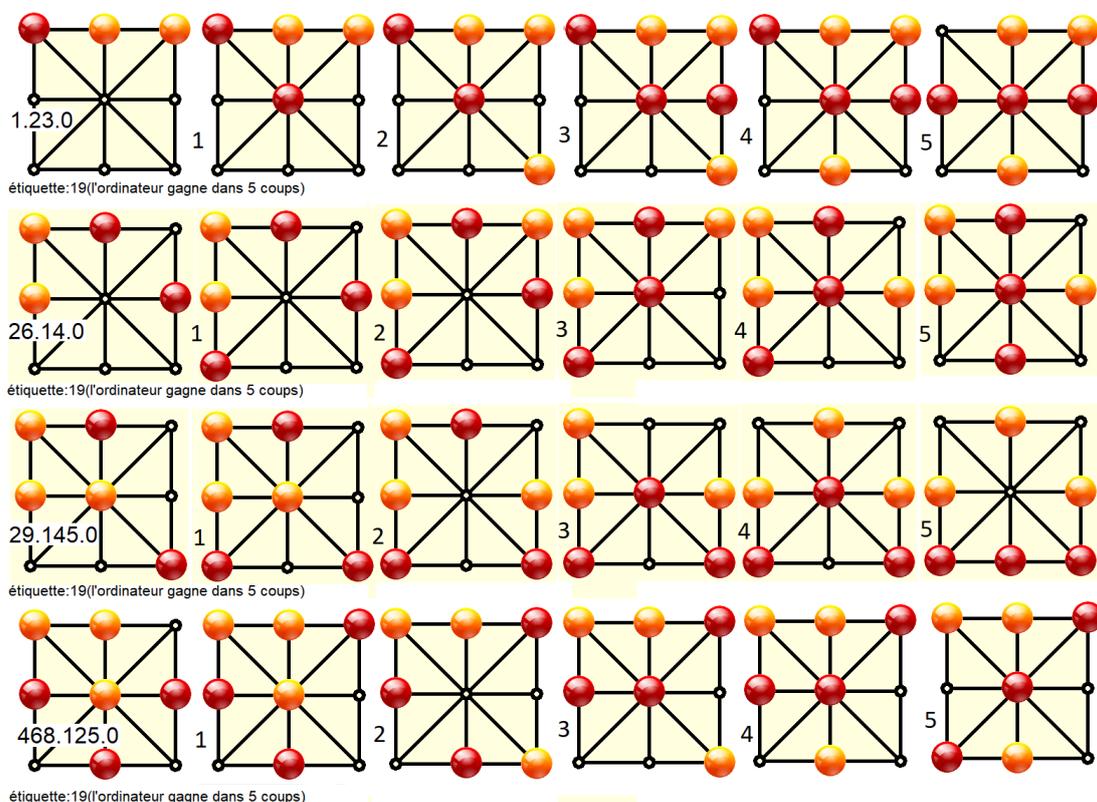


FIGURE 3.24 – Quatre parties que l'ordinateur gagne en cinq coups ; elles concernent des configurations à trois, quatre, cinq et six pions où c'est au tour de l'ordinateur de jouer.

### c. Réalisation du jeu

La réalisation pratique du jeu interactif dépend, bien sûr, de certains choix techniques : quel langage utiliser, quels outils graphiques, quelles interactions autoriser avec le joueur humain, etc. Pour ma part et dans un premier temps, j'ai choisi d'utiliser le langage *Python* avec l'outil graphique *tkinter* et une interaction avec la souris exclusivement. Ce choix empêche d'utiliser ce jeu sur un portable ou sur internet mais je peux y jouer sur mon ordinateur portable et en partager le programme avec des utilisateurs disposant d'un environnement de développement *Python*.

Je passe donc sur la plupart des détails techniques de programmation. Ce qui m'importe est de pouvoir introduire mon graphe étiqueté dans ce programme et de le faire utiliser par l'ordinateur lorsque c'est à lui de jouer. Dès le lancement du programme, le fichier du graphe est donc lu et mis en mémoire dans une liste que j'ai organisé en trois niveaux :

1. deux sous-listes correspondant au joueur dont c'est le tour de jouer
2. dans chacune de ces sous-listes de tour, sept sous-listes correspondant au nombre de billes en jeu
3. dans chacune de ces sous-listes du nombre de billes, la liste des différentes configurations du graphe concernées

Ce rangement des configurations facilite le traitement et la compréhension du programme. Je peux ainsi écrire une instruction comme `liste_graphe[0][6][25][:-2]` dont le sens est clair : il s'agit d'une configuration jouée par l'ordinateur (`joueur=0`) contenant les six pions (`pions=6`) et référencée par le numéro 25 dans la sous-liste concernée (`rang=25`). Le dernier crochet est utilisé pour extraire de la chaîne de caractères l'identifiant de la configuration en excluant l'étiquette qui occupe les deux derniers caractères (`liste_graphe[0][6][25]=234.159.0.0` est une position gagnante pour l'humain, `liste_graphe[0][6][25][:-2]=234.159.0` est son identifiant dans le graphe).

Pour le jeu de l'humain, je n'utilise pas les étiquettes du graphe. Le programme laisse l'utilisateur déplacer un de ses pions où il le souhaite en veillant juste à la régularité du déplacement. Techniquement, le mouvement d'un pion du joueur humain se réalise en cliquant sur le pion et en le plaçant à l'endroit souhaité. Bien sûr cela suppose un calcul de l'emplacement concerné à l'aide des coordonnées du pointeur de souris, un contrôle de la disponibilité de l'emplacement et de la régularité du coup, puis un enregistrement de la configuration et une traduction de celui-ci dans le système de codification du graphe.

Une procédure examine après chaque coup si un alignement a été réalisé ce qui déclencherait la fin de la partie, la mise à jour du score et une nouvelle partie proposée par un clic sur le bouton **rejouer**. Le joueur qui commence une partie est toujours celui qui a perdu la partie précédente sauf pour la première partie où le choix est laissé au hasard. De cette façon, le joueur humain peut progresser car il ouvre les parties tant qu'il perd.

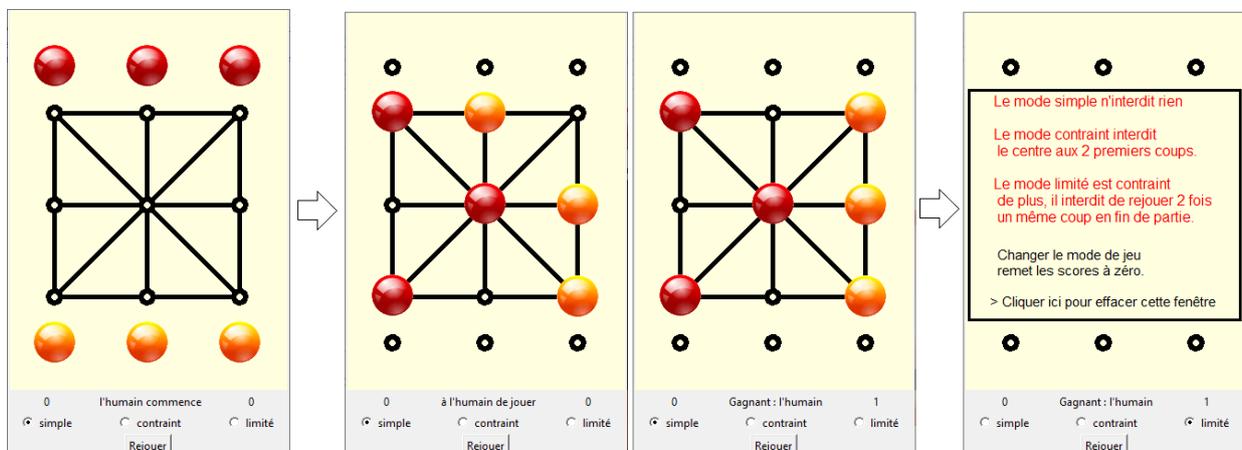


FIGURE 3.25 – Aspects du jeu réalisé en *Python* montrant le visuel d'une partie simple à commencer par l'humain (à gauche); au centre de l'image, les deux derniers coups de la partie, gagnée par l'humain (ça peut arriver!). Le panneau rappelant les différents mode de jeu (à droite) apparaît lorsque le joueur clique sur un nouveau mode de jeu (disponible en fin de partie uniquement).

Les différents mode de jeu implémentés nécessitent quelques adaptations du programme.

- ✦ Le mode **contraint** empêche les joueurs de jouer au centre lors des deux premiers coups. Pour cela j'ai introduit dans le programme les quelques sous-listes à utiliser dans ce cas pour le jeu de l'ordinateur (`liste_graphe[0][1]` ne contient plus la configuration "0.5.0.8", `liste_graphe[1][1]` ne contient plus la configuration "5.0.1.16", `liste_graphe[1][2]` ne contient plus 72 configurations mais seulement 56, exit les "5.1.1.12", "5.2.1.18", "3.5.1.7", etc.). Pour limiter le jeu de l'humain c'est encore plus facile.
- ✦ Le mode **limité** permet de mettre fin à la partie lorsque, dans la phase dynamique, une même configuration est utilisée deux fois. J'avais prévu au départ de repousser à trois le nombre d'utilisations d'une même configuration mais, finalement, il m'a paru plus simple d'arrêter la partie dès la seconde utilisation. L'avertissement que j'avais prévu n'était pas évident à présenter au joueur : faudrait-il une liste de coups (difficile à interpréter) ou une représentation visuelle des coups déjà joués (surcharge d'information) ? Comme l'incidence sur le jeu est limitée (si les joueurs sont vraiment à égalité, autant la déclarer le plus tôt possible) j'ai renoncé à faire chercher à l'ordinateur le coup qui évite au maximum la partie nulle. Mais on pourrait éventuellement revenir sur ce point ultérieurement.

Pour les besoins du développement j'ai affiché sur la console des informations sur chaque coup. La figure suivante montre une partie dans le mode **limité** qui se termine en seize coups par l'égalité entre les joueurs. Cette égalité ne modifie pas les scores mais permet de laisser le hasard décider du joueur qui commencera la prochaine partie. C'est un avantage pour le joueur humain s'il sait comment gagner contre l'ordinateur à ce niveau de jeu.

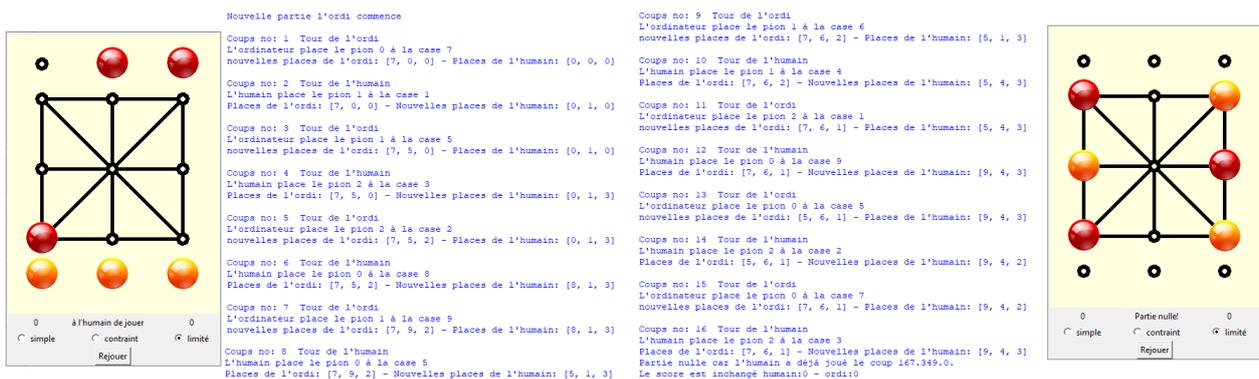


FIGURE 3.26 – Une partie en mode « limitée » qui se termine par une égalité car le joueur humain a déjà joué au coup 12 la même configuration ("167.349.0") qu'il a rejoué au coup 16.

#### d. Étude des parties à égalité

Une partie peut se terminer sur la constatation d'une égalité : l'algorithme détectant une configuration déjà rencontrée arrête la partie. Cette fonctionnalité a été réalisée et fonctionne de façon satisfaisante, cependant j'ai fait le choix (provisoire) de choisir une configuration indécise au hasard parmi les configurations indécises qui succèdent à la configuration indécise laissée par le joueur humain. Comme je l'ai dit plus haut, ce choix permet d'arrêter la partie assez rapidement. Mais il paraît évident que l'ordinateur aurait intérêt à jouer le plus longtemps possible car on peut s'attendre à une erreur du joueur humain qui amènerait une configuration favorable à l'ordinateur.

La question qui m'interpelle alors est la suivante : combien de coups au maximum peut contenir la phase dynamique lorsqu'elle est jouée parfaitement ? Cette phase commence sur une configuration indécise sinon la partie se terminerait sur la victoire de l'un ou l'autre joueur. Je me place ici dans le cadre de la première application où l'ordinateur joue à la place des deux joueurs : recevant une configuration indécise il en retourne une autre. Est-ce possible de faire en sorte que la partie passe par toutes les configurations indécises possibles ? Comme il y en a 272 pour chacun des joueurs, est-ce possible que la phase dynamique contienne les  $272 \times 2 = 544$  coups correspondants aux 544 configurations indécises possibles ?

### Sous-graphe des configurations indécises complètes

Les configurations indécises complètes forment un sous-graphe du graphe des configurations. Ce sous-graphe est biparti et orienté. On a déjà dit pourquoi : ce n'est pas le même joueur qui déplace un pion d'une configuration à celle qui lui succède ou à celle qui la précède. On passe ainsi d'une configuration-humain (où c'est à l'humain de jouer) à une configuration-ordinateur (où c'est à l'ordinateur de jouer). Les deux sous-ensembles de sommets de ce sous-graphe se déduisent l'un de l'autre en échangeant les places. Par exemple la configuration-ordinateur 137.245 se déduit de la configuration-humain 245.137. Les deux sous-ensembles ont par conséquent la même structure, on peut n'en étudier qu'un seul.

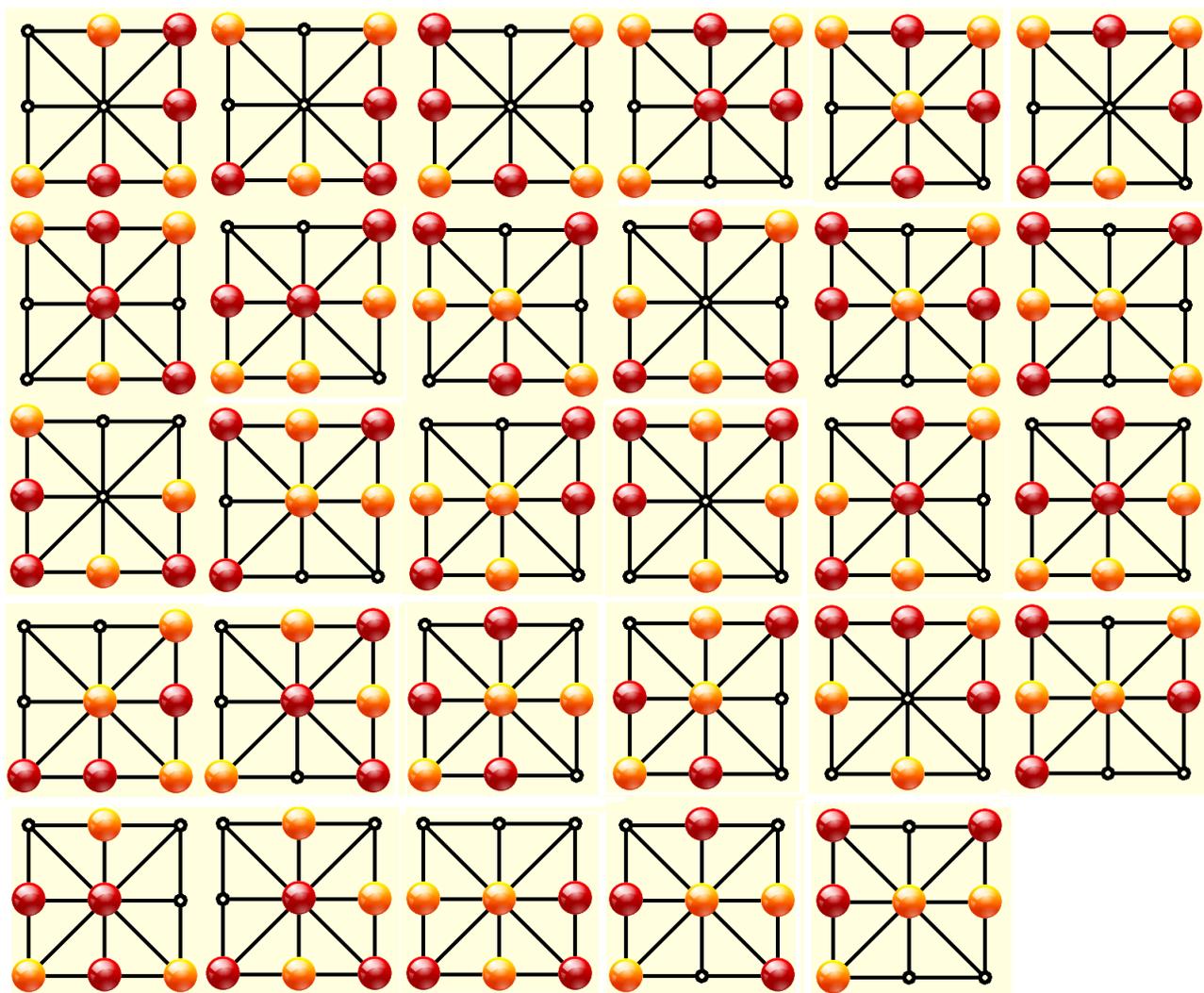


FIGURE 3.27 – Vingt-neuf configurations-ordinateur indécises et complètes ne présentent pas d'élément de symétrie ; ces configurations contiennent chacune huit alias.

La plupart des configurations existent sous huit formes équivalentes qui se déduisent les unes des autres par des rotations d'un quart de tour ou d'un demi tour et par des symétries autour d'un axe vertical, horizontal ou oblique. Il existe dix configurations présentant un élément de symétrie qui réduit le nombre de ses configurations équivalentes – je les appellerai désormais *alias* – à quatre ; vingt-neuf autres groupes contiennent chacun huit alias d'une configuration sans élément de symétrie. Les configurations-ordinateur sont donc bien  $10 \times 4 + 28 \times 8 = 272$ , de même pour les configurations-humain.

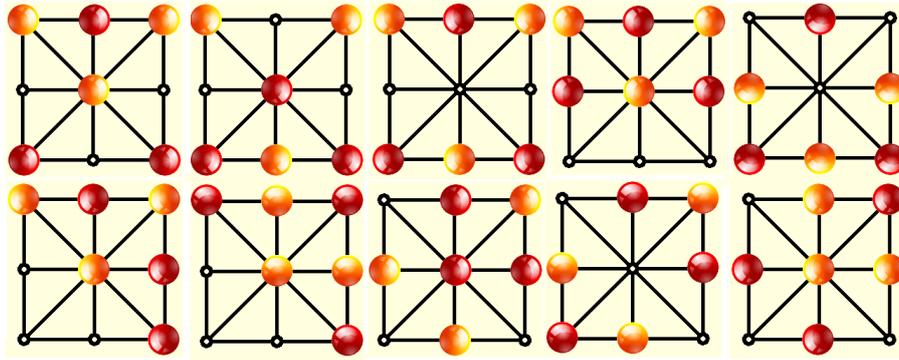


FIGURE 3.28 – Dix configurations-ordinateur indécis et complètes présentent un axe de symétrie : pour la moitié d'entre elles (ligne du haut) l'axe est parallèle aux bords, pour l'autre moitié (ligne du bas) l'axe est une diagonale. Ces configurations contiennent chacune quatre alias, un seul est montré ici.

### Diamètre du sous-graphe

Pour mieux connaître ce sous-graphe des CIC (configurations indécisées complètes), j'ai eu l'idée de rechercher pour chacune des 33 catégories de configuration, l'ensemble des configurations qui pouvaient être accessibles sans sortir de ce sous-graphe. Pour cela, j'ai étiqueté les 544 configurations du sous-graphe selon la distance à laquelle elle se trouve de la configuration initiale. Si une configuration n'a pas été encore étiquetée, elle reçoit l'étiquette  $-1$ . Ainsi, au départ, une seule configuration a reçu l'étiquette  $0$  (la configuration initiale). Ensuite, on marque tous les successeurs indécis de cette configuration initiale avec l'étiquette  $1$ . On marque ensuite tous les successeurs indécis de chacune des configurations étiquetées  $1$  avec l'étiquette  $2$ . Et ainsi de suite jusqu'à ce qu'on ne trouve plus aucune configuration étiquetable. Les configurations qui restent étiquetées  $-1$  sont inaccessibles à partir de la configuration initiale. La distance maximum qui sépare la configuration initiale d'une configuration accessible est appelée « diamètre » du sous-graphe.

```

Etiquetage des sommets relativement à la configuration N° 1 : 267.139.0
distances= [1, 3, 4, 8, 23, 31, 36, 46, 38, 28, 22, 12, 4]
Configurations atteintes à la distance 0 : ['267.139.0']
Configurations atteintes à la distance 1 : ['256.139.1', '567.139.1', '257.139.1']
Configurations atteintes à la distance 2 : ['256.138.0', '256.349.0', '567.349.0', '257.138.0']
Configurations atteintes à la distance 3 : ['246.138.1', '259.138.1', '267.138.1', '268.349.1', '156.349.1', '267.349.1', '279.138.1', '167.349.1']
Configurations atteintes à la distance 4 : ['279.135.0', '279.358.0', '279.158.0', '279.348.0', '279.168.0', '167.359.0', '167.345.0', '167.459.0',
'167.249.0', '167.348.0', '246.135.0', '246.358.0', '246.158.0', '268.359.0', '268.459.0', '268.345.0', '156.249.0', '259.168.0', '267.348.0',
'267.358.0', '267.158.0', '267.345.0', '267.459.0']
Configurations atteintes à la distance 5 : ['267.359.1', '168.359.1', '249.135.1', '267.135.1', '256.348.1', '279.348.1', '167.348.1', '567.348.1',
'257.348.1', '267.358.1', '168.459.1', '249.158.1', '267.345.1', '279.358.1', '167.345.1', '249.168.1', '279.168.1', '168.249.1', '167.249.1',
'167.459.1', '279.158.1', '137.249.1', '156.249.1', '567.249.1', '179.348.1', '267.348.1', '259.348.1', '379.168.1', '259.168.1', '257.168.1',
'156.348.1']
Configurations atteintes à la distance 6 : ['179.458.0', '179.358.0', '179.345.0', '137.245.0', '137.459.0', '137.259.0', '379.568.0', '379.156.0',
'379.158.0', '168.245.0', '168.349.0', '168.279.0', '168.459.0', '168.259.0', '249.568.0', '249.138.0', '249.167.0', '249.158.0', '249.156.0',
'267.458.0', '267.138.0', '267.349.0', '279.138.0', '279.468.0', '279.345.0', '279.156.0', '167.349.0', '167.248.0', '167.259.0', '167.358.0',
'156.349.0', '259.138.0', '168.249.0', '249.168.0', '168.379.0', '249.137.0']
Configurations atteintes à la distance 7 : ['245.137.1', '459.137.1', '259.137.1', '568.379.1', '156.379.1', '158.379.1', '245.168.1', '349.168.1',
'459.168.1', '568.249.1', '138.249.1', '158.249.1', '157.349.1', '567.349.1', '168.349.1', '579.138.1', '257.138.1', '249.138.1', '279.568.1',
'349.568.1', '167.245.1', '138.245.1', '279.458.1', '167.458.1', '579.468.1', '257.468.1', '259.468.1', '157.248.1', '156.248.1', '567.248.1',
'567.138.1', '257.349.1', '459.138.1', '158.349.1', '138.279.1', '468.279.1', '156.279.1', '158.279.1', '349.167.1', '248.167.1', '259.167.1',
'459.167.1', '349.156.1', '138.259.1', '168.259.1', '249.156.1']
Configurations atteintes à la distance 8 : ['157.348.0', '157.249.0', '579.168.0', '579.348.0', '349.157.0', '349.156.0', '349.567.0', '349.168.0',
'349.267.0', '138.579.0', '138.259.0', '138.257.0', '138.267.0', '138.249.0', '568.279.0', '568.349.0', '245.167.0', '245.138.0', '468.579.0',
'468.257.0', '468.259.0', '248.157.0', '248.156.0', '248.567.0', '349.268.0', '138.246.0', '138.279.0', '138.459.0', '349.167.0', '349.158.0',
'257.348.0', '567.348.0', '158.279.0', '459.167.0', '459.168.0', '158.249.0', '168.345.0', '249.358.0']
Configurations atteintes à la distance 9 : ['348.157.1', '249.157.1', '168.579.1', '348.579.1', '135.279.1', '358.279.1', '348.279.1', '168.279.1',
'359.167.1', '345.167.1', '249.167.1', '348.167.1', '135.246.1', '358.246.1', '158.246.1', '359.268.1', '459.268.1', '345.268.1', '348.257.1',
'348.567.1', '345.168.1', '139.267.1', '348.267.1', '358.267.1', '158.267.1', '358.249.1', '345.267.1', '459.267.1']
Configurations atteintes à la distance 10 : ['139.256.0', '139.567.0', '139.257.0', '359.267.0', '359.168.0', '135.249.0', '135.267.0', '348.256.0',
'348.279.0', '348.167.0', '348.567.0', '348.257.0', '358.267.0', '345.267.0', '358.279.0', '345.167.0', '348.267.0', '249.567.0', '348.179.0',
'348.259.0', '168.257.0', '348.156.0']
Configurations atteintes à la distance 11 : ['458.179.1', '358.179.1', '345.179.1', '458.267.1', '138.267.1', '349.267.1', '138.256.1', '349.256.1',
'345.279.1', '358.167.1', '349.567.1', '138.257.1']
Configurations atteintes à la distance 12 : ['458.279.0', '458.167.0', '138.567.0', '349.257.0']

```

FIGURE 3.29 – En partant de la configuration-ordinateur indécisée notée 267.139.0, on peut atteindre trois configuration-humain indécisées, notées 256.139.1, 567.139.1 et 257.139.1 ainsi que 252 autres configurations (ordinateur ou humain). De ce point initial, il y a donc 288 configurations inaccessibles.

En procédant ainsi, j'obtiens un résultat aussi intéressant qu'imprévu : certaines configurations ne sont pas accessibles dans un cheminement indécis. Ces configurations inaccessibles sont nombreuses puisqu'il y en a 144 pour chacun des joueurs. Elles sont susceptibles, bien entendu, d'être des points de départ d'un cheminement indécis mais on ne peut les atteindre si on part d'une configuration qui

n'appartient pas à ce sous-ensemble inaccessible. Par exemple la configuration notée 245.168.0 n'est accessible par un chemin indécis que lorsqu'on la prend comme point de départ ou bien lorsqu'on part de 245.169.1 ou 245.167.1 (et aussi 245.189.1 et 245.167.1 mais ces configurations étant symétriques des deux précédentes ne seront pas envisagées). La première configuration ne peut être précédée d'une configuration-ordinateur indécis car ses prédécesseurs potentiels sont 345.169.0, 234.169.0, 257.169.0, 247.169.0 ou 248.169.0 toutes cinq gagnantes pour l'ordinateur (en un ou trois coups ainsi qu'on peut le vérifier à partir de l'image suivante). La seconde configuration ne peut être précédée d'une configuration-ordinateur indécis car elle est elle-même gagnante en trois coups pour l'humain.

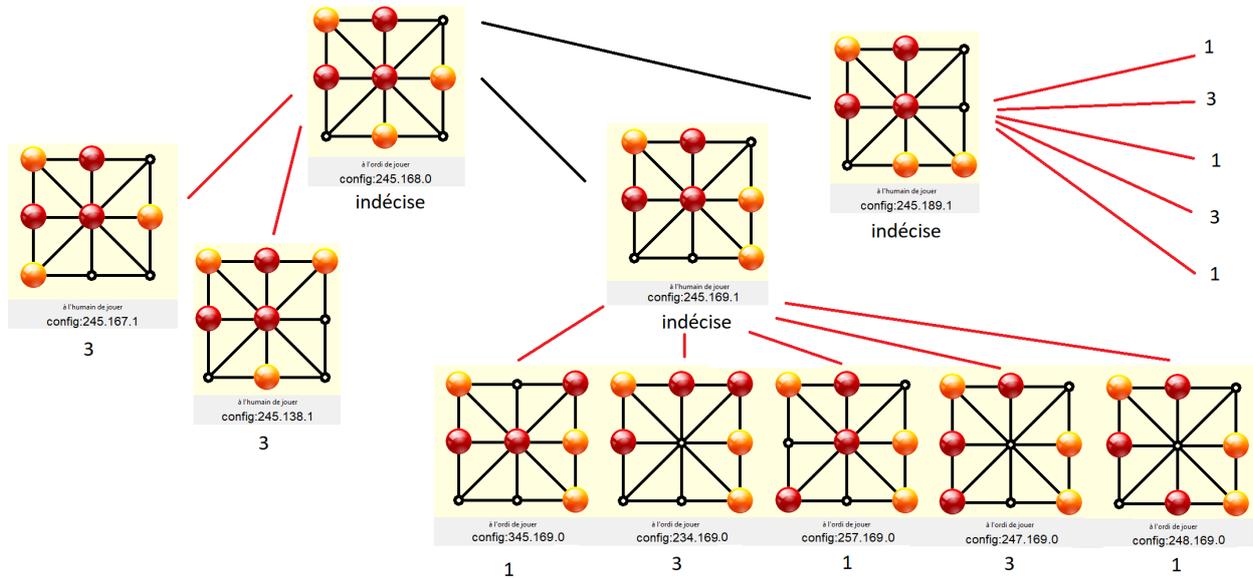


FIGURE 3.30 – La configuration-ordinateur 245.168.0 ainsi que les deux configurations-humain 245.169.1 et 245.189.1 sont indécisées et inaccessibles par d'autres configurations indécisées car leurs prédécesseurs, reliés en rouge, sont gagnants pour l'un ou l'autre des joueurs.

L'entrée dans ce sous-graphe des configurations complètes indécisées pouvant se faire par une de ces 288 configurations inaccessibles ou par une des 256 autres configurations, un chemin indécis peut comporter au maximum 258, 257 ou 256 étapes selon le niveau d'accessibilité. Ma question initiale sur ce sous-graphe doit être reformulée ainsi : *la phase dynamique jouée parfaitement peut-elle passer par ces 256 configurations indécisées sans jamais passer deux fois sur une même configuration* (ou 257 ou 258 si on la commence avec une des configurations inaccessibles) ? Le problème se recentre donc sur le noyau entièrement accessible du sous-graphe des configurations indécisées qui ne contient plus que 256 configurations. Dans la suite j'appellerai « noyau indécis » (NI) ce sous-graphe du sous-graphe des CIC.

```
Configurations-ordina= ['267.139.0', '348.179.0', '249.137.0', '168.379.0', '248.157.0', '468.579.0', '268.359.0', '246.135.0',
'267.348.0', '249.168.0', '168.249.0', '348.267.0', '167.359.0', '349.157.0', '138.579.0', '279.135.0', '167.349.0', '138.279.0',
'279.138.0', '349.167.0', '379.568.0', '139.256.0', '137.245.0', '179.458.0', '279.468.0', '138.246.0', '349.268.0', '167.248.0',
'249.568.0', '267.458.0', '348.256.0', '168.245.0', '256.138.0', '568.279.0', '256.349.0', '568.349.0', '245.167.0', '458.279.0',
'245.138.0', '458.167.0', '348.279.0', '267.138.0', '267.349.0', '249.138.0', '168.349.0', '168.279.0', '249.167.0', '348.167.0',
'246.358.0', '268.459.0', '248.156.0', '468.257.0', '468.259.0', '246.158.0', '248.567.0', '268.345.0', '348.567.0', '267.358.0',
'249.158.0', '348.257.0', '168.459.0', '249.156.0', '267.345.0', '168.259.0', '259.138.0', '158.279.0', '257.138.0', '567.349.0',
'358.279.0', '459.167.0', '345.167.0', '156.349.0', '345.267.0', '567.348.0', '259.168.0', '358.267.0', '459.168.0', '158.249.0',
'156.249.0', '257.348.0', '249.567.0', '348.259.0', '168.257.0', '168.345.0', '267.459.0', '249.358.0', '348.156.0', '267.158.0',
'279.345.0', '167.259.0', '138.459.0', '349.257.0', '279.156.0', '138.567.0', '349.158.0', '167.358.0', '167.345.0', '349.156.0',
'349.567.0', '279.358.0', '279.158.0', '138.259.0', '167.459.0', '138.257.0', '167.249.0', '279.348.0', '349.168.0', '279.168.0',
'138.267.0', '138.249.0', '167.348.0', '349.267.0', '359.267.0', '157.348.0', '157.249.0', '579.168.0', '157.249.0', '359.168.0', '579.348.0',
'135.249.0', '135.267.0', '379.156.0', '179.358.0', '379.158.0', '139.567.0', '137.459.0', '137.259.0', '139.257.0', '179.345.0']
```

FIGURE 3.31 – Le noyau indécis est un graphe de 128 configurations-ordinateur et autant de configurations-humain. Ici ne sont notées que les configurations-ordinateur mais pour chacune, il existe la configuration-humain correspondante, obtenue en échangeant les deux blocs de code. Par exemple à la configuration-ordinateur 267.348.0 correspond la configuration-humain 348.267.1.

## Recherche d'un chemin hamiltonien dans le noyau indécis

Un chemin hamiltonien est un parcours des différents sommets d'un graphe ne passant qu'une seule fois par tous les sommets du graphe. Ma question peut donc s'énoncer ainsi : dans le graphe appelé « noyau indécis » des configurations complètes indécises et accessibles, existe-t-il un chemin hamiltonien ? La recherche d'un tel chemin par programme est un problème dit « NP-complet », c'est-à-dire difficile à résoudre dans un temps raisonnable. On peut imaginer un algorithme naïf qui teste exhaustivement toutes les permutations des 256 sommets de ce graphe mais il y a des chances que ce soit long puisqu'il y a  $256!$  telles permutations, un nombre que ma calculatrice dit égal à  $\infty$  (le quasi infini). Comme je peux commencer par un joueur ou par l'autre et qu'aux rotations et symétries près il n'y a que 33 configurations de départ, cela reste largement hors de portée.

Nonobstant ce décourageant préambule, je me lance dans la confection d'un programme qui, s'il ne débusque pas par hasard de chemin hamiltonien, pourra me donner sa plus longue chaîne sans répétition. L'algorithme que j'ai adapté à ce problème est le *backtracking*, soit, dit en français, le retour sur trace ou retour arrière. Son avantage est qu'il abandonne la construction d'une solution si une étape est impossible, auquel cas il rebrousse chemin (d'où son nom) jusqu'au point où il est possible de choisir une autre voie. Comparé à la recherche exhaustive naïve, le gain en terme de temps de traitement est significatif et, comme j'avais déjà déployé un tel algorithme pour chercher une solution à un problème de tangram, l'adaptation à cette nouvelle situation est relativement aisée.

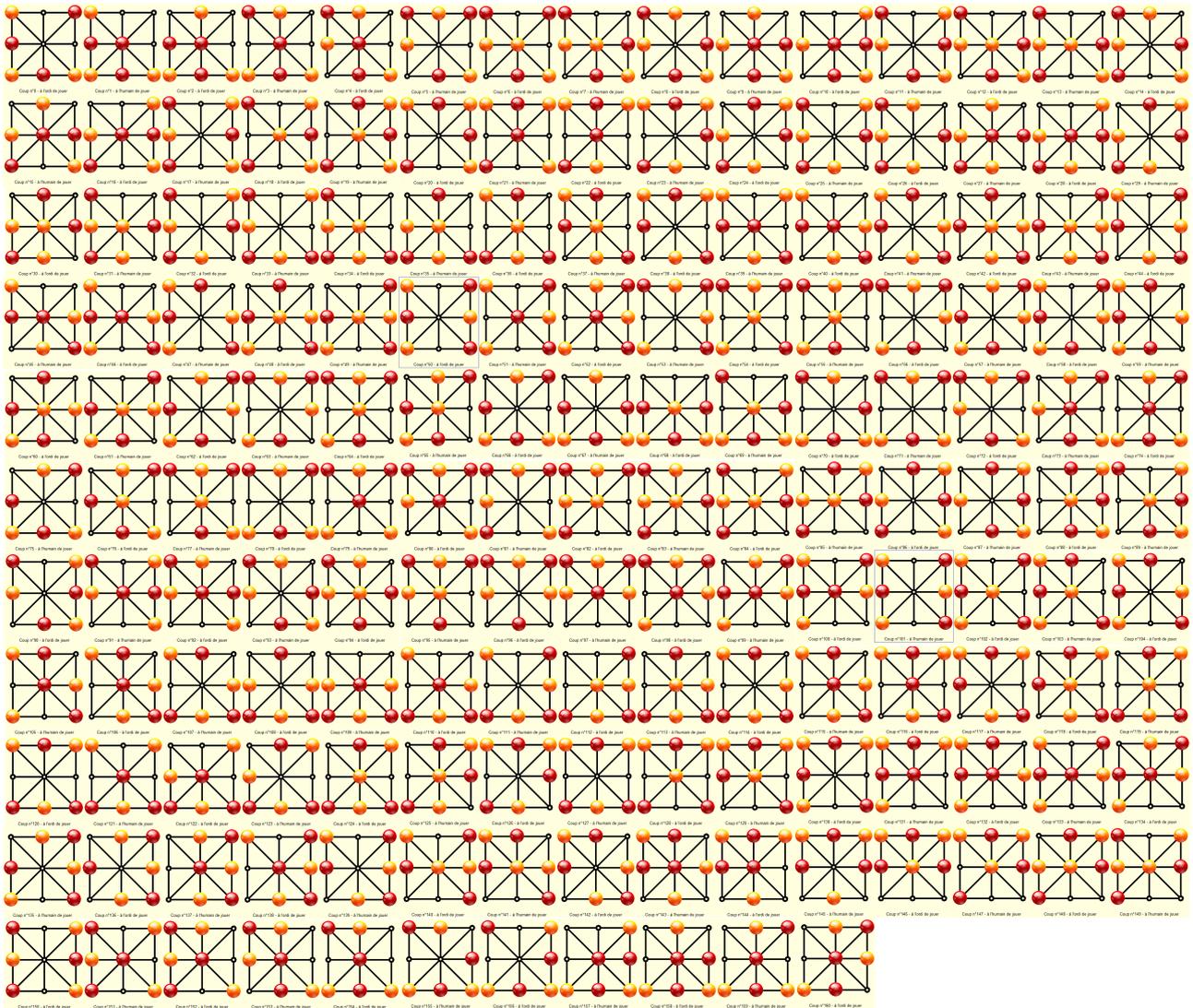


FIGURE 3.32 – Une chaîne hamiltonienne en 161 coups partant de la configuration-ordinateur 348. 179. Aucune de ces configurations n'y est identique malgré les ressemblances parfois déroutantes : comparez la 50<sup>e</sup> et la 101<sup>e</sup> (encadrées), les pions sont placés de la même façon, ce qui les distingue est le joueur qui doit jouer.

La durée du traitement pour obtenir un résultat définitif et complet est sans ambiguïté très importante. Malgré cela, j'obtiens rapidement des résultats substantiels en faisant afficher le plus long chemin trouvé dès que la longueur d'un chemin dépasse la longueur maximum déjà atteinte. Pour la configuration initiale 348.179.0, j'obtiens très rapidement, en quelques minutes, un chemin en 128 étapes, mais pour être certain qu'il n'y en a pas de plus long il faut le laisser tourner encore des heures. Comme j'ai également fait afficher l'identifiant des configurations enlevées du chemin lors des retours arrières, je peux suivre la progression du traitement et tenter d'en évaluer la durée. Au bout de 2 heures avec la même configuration initiale j'ai obtenu un chemin contenant 161 étapes et le retour arrière est alors très important : il descend jusqu'au coup 80 sans en trouver de plus long, mais descendre encore en dessous demande environ 2 nouvelles heures et ce retour amène le chemin à revoir le 73<sup>e</sup> coup. L'attente est alors encore plus longue...

## Finalisation du jeu

Comme rien ne se passe pendant plus de douze heures, le mieux à faire pour ce traitement c'est de l'oublier, l'ordinateur continuant gentiment à brasser des morceaux de chemins hamiltoniens, il finira bien par s'arrêter. Pendant ce temps, je peux commencer à réfléchir sur une éventuelle utilisation de ces résultats pour le jeu. Que gagnerait l'intelligence artificielle à chercher à prolonger un chemin indéci? Cela me paraît évident, elle espérerait une erreur du joueur humain, plus désarmé qu'elle, qui amènerait une position favorable à l'ordinateur. Mais l'autre question qui émerge est : comment arriver à prolonger le jeu indéci? Le programme ne peut pas recalculer de chaîne hamiltonienne sans y passer des heures et quand bien même il en aurait une à disposition, elle ne lui servirait à rien car il n'y a pas de meilleure façon de jouer le long d'une chaîne hamiltonienne, toutes les configurations traversées sont associées à la même valeur de jeu (12 : indéci). Il suffit donc, pour assurer un coup le moins proche possible d'une fin de partie à égalité, de jouer de manière à éviter une répétition, cela ne devrait pas être trop difficile à mettre en œuvre.

Je vais tenter de laisser au joueur humain le soin de terminer la partie sur une égalité, c'est-à-dire de trouver la répétition le plus vite possible, ce qui lui évitera de commettre une erreur éventuelle. Mon idée de départ était d'avertir le joueur humain qu'une configuration avait déjà été jouée, pour lui donner l'occasion de la mémoriser pour éventuellement l'éviter. Mais si je fais ça, le joueur humain, averti qu'il ne peut espérer au mieux qu'une partie nulle, devrait tout faire pour jouer à nouveau cette configuration. Si le programme l'en avertit avant de sanctionner la répétition par une égalité, le joueur est injustement retardé dans cette possibilité qu'il avait peut-être sciemment cherchée et cela lui porte préjudice. Cette conclusion m'incite à revoir mon idée première : il ne faut pas avertir le joueur d'une éventuelle répétition, celle-ci arrivera à plus ou moins longue échéance et, tant qu'elle n'est pas trouvée, le joueur doit éviter de faire une erreur. Je finalise sur cette base le jeu en Python.

Son fonctionnement est irréprochable de mon point de vue : je me rends compte qu'il n'est plus intéressant de jouer sans la contrainte des deux premiers coups car l'ordinateur et moi-même connaissons le coup initial qui tue le jeu. Les conséquences de la suppression du coup au centre au 2<sup>e</sup> coup sont moins bien connues par le joueur humain que je suis et, par conséquent, j'en suis réduit à jouer le jeu indéci pour obtenir une égalité. Une autre conséquence imprévue de ce travail : je propose à des proches de se mesurer à l'ordinateur et suis étonné de constater que certains refusent sachant qu'ils vont perdre! Je n'aurais pas dû leur parler de mon objectif qui était de rendre l'ordinateur imbattable...

Je me suis imposé un dernier effort avant de boucler ce chapitre : traduire le programme Python en un programme Javascript afin que le jeu soit jouable sur internet. Mes premiers essais remontent à quelques années et je n'étais pas trop content du résultat final car le jeu en Javascript que je proposais n'était pas doté d'une intelligence remarquable. J'avais écrit un programme en Python qui fonctionnait bien mieux, basé sur l'algorithme du minimax mais imparfait : je n'étais pas allé au-delà d'une certaine profondeur d'analyse n'ayant pas exploré tout le graphe de jeu pour en sortir une fonction d'évaluation imparable. Ceci dit, je me met à la traduction, une instruction après l'autre, confronté à cette difficulté technique de devoir reconstruire cet assemblage délicat qu'est un programme informatique. On croirait qu'il n'y a que des mots à échanger mais c'est plus compliqué

que cela, il faut tout revoir et régler avec minutie comme une horlogerie de précision. Quelques jours de travail sont nécessaires, le plaisir de l'accomplissement n'étant pas réservé au point final de l'œuvre mais diffusé tout au long de l'aventure. Pourquoi ne puis-je déplacer le pion de l'humain ? Je vérifie toutes mes affectations, cherche un booléen qui n'est pas modifié au bon endroit, affiche dans la console les valeurs de nombreux paramètres et scrute les valeurs qu'ils prennent les comparant à celles qu'ils sont supposés prendre. Finalement, je découvre que c'est mon instruction qui est mal écrite : en Javascript on n'écrit pas `sorted(ordiPlaces)` comme en Python lorsqu'on veut ordonner une liste en place, ni `ordiPlaces.sort()` qui modifie la liste initiale, mais `ordiPlaces.toSorted()`. Cet exemple parmi cent autres qui arrêtent la progression mais qui, une fois résolus, me rapprochent de la version fonctionnelle finale et me procurent la petite dose d'endorphine naturelle accompagnant toute réussite, aussi dérisoire qu'elle puisse paraître.

### 3. Annexes

#### a. L'analyse de la phase statique par Berlekamp, Conway et Guy

Il s'agit ici du Tic-Tac-Toe ou morpion, un jeu légèrement différent du Tapatán puisqu'il s'arrête à la fin de la pose des pions (dénommée phase statique dans mon texte) et que ceux-ci peuvent être plus que trois, la fin de la partie étant marquée par un alignement ou un plateau plein. Le plateau a beau être le même et les règles identiques, hormis la petite différence mentionnée, les conséquences sur les éléments stratégiques sont radicalement différentes. Comparer ce tableau avec les figures 3.5 ou 3.7 où j'ai repris l'idée et les notations de ce brillant trio. Ma codification des emplacements étant légèrement différente, la transposition n'est pas aisée, désolé.

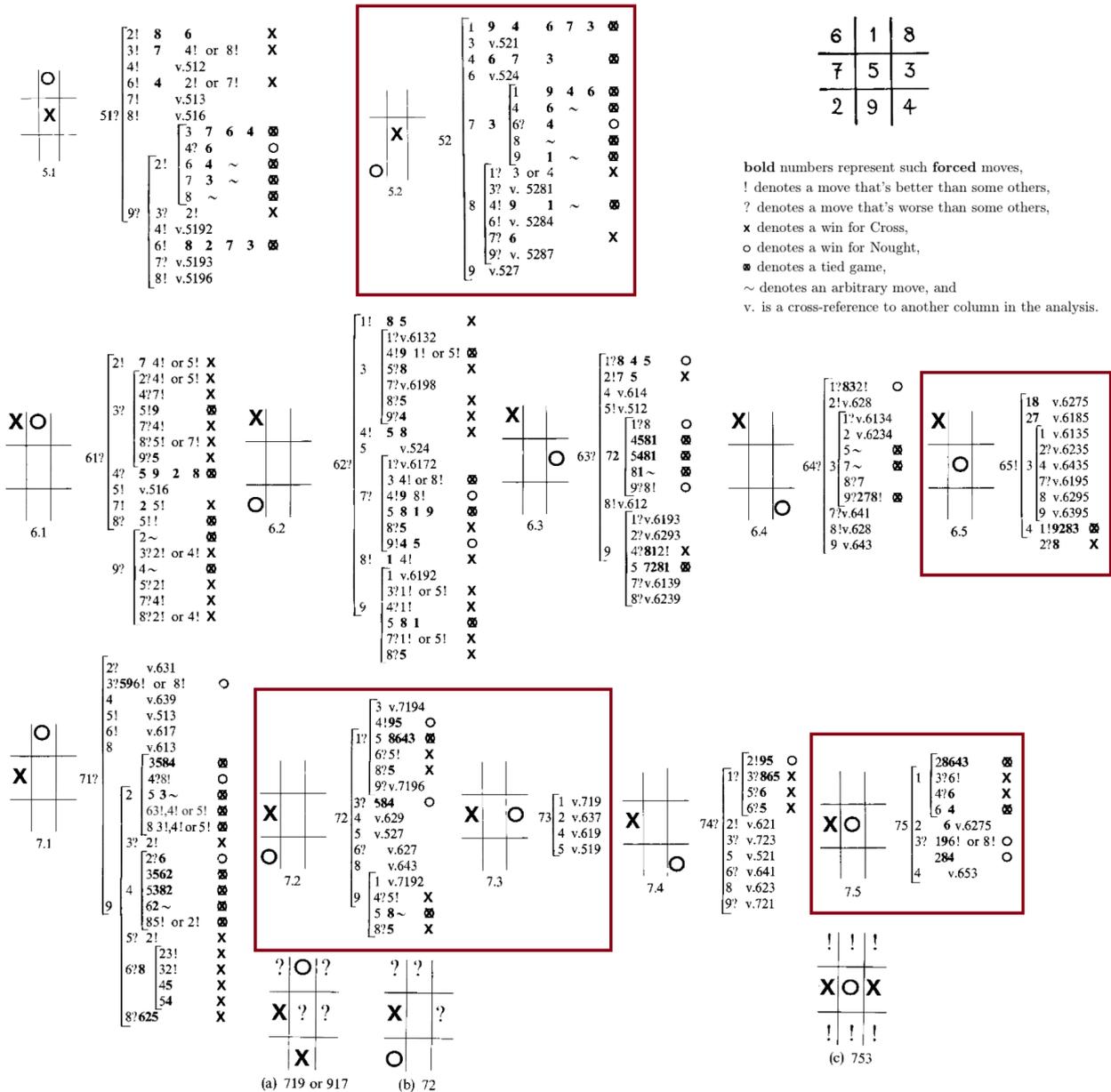


FIGURE 3.33 – La numérotation des cases par ces auteurs suit le carré magique en haut à gauche. C'est toujours le joueur noté  $x$  qui commence. Les indications  $!$  ou  $?$  indiquent les meilleurs ou les pires coups (j'ai encadré les meilleurs 3<sup>e</sup> coups). Noter les indications de référence croisée (telle que  $v. 512$ ) qui évitent les répétitions aux auteurs et allègent la présentation.

## b. Différents algorithmes utilisés

L'étude des graphes se prête à de nombreuses applications qui se résument parfois à des problèmes assez proches pour que les solutions algorithmiques proposées soient adaptables à d'autres situations du même type. Dans ce chapitre, j'ai notamment rencontré :

- ✦ L'algorithme du minimax : il s'agit d'un problème d'intelligence artificielle, utile pour les problèmes de la théorie des jeux à deux joueurs, somme nulle et information complète où il s'agit de minimiser la perte maximum. Le théorème du minimax de von Neumann<sup>2</sup> assure l'existence d'un tel algorithme mais il n'est pas souvent aisé de le trouver. Cet algorithme amène l'ordinateur à envisager toutes les possibilités pour un nombre limité de coups et à leur attribuer une valeur qui dépend du bénéfice qu'en tirera un joueur et son adversaire. Le meilleur choix pour le joueur est celui qui minimise ses pertes en estimant que son adversaire cherche à les maximiser. Dans le cas du Tapatan, le jeu n'a pas un graphe trop vaste pour ne l'explorer que jusqu'à une certaine profondeur. On a pu partir des situations gagnantes et en déduire les valeurs de jeu de toutes les configurations qui leur étaient liées en amont.
- ✦ L'algorithme de recherche du diamètre d'un graphe : on cherche le plus court chemin entre chaque paire de sommets ; la plus grande longueur de l'un de ces chemins est le diamètre du graphe. Le graphe est parcouru *en largeur* pour un sommet en marquant tous les sommets qui sont atteints depuis le sommet initial par la longueur de l'arc (1 si tous les arcs ont la même longueur comme c'est le cas avec le Tapatan). Ensuite, on procède de même entre chacun des sommets qui viennent d'être marqués et les sommets qui leurs sont accessibles ; on leur attribue la somme des longueurs des deux arcs qui les joignent au sommet initial. On parcourt ainsi tout le graphe et la plus grande valeur attribuée est son diamètre.
- ✦ L'algorithme de backtracking (retour sur trace ou retour arrière) : cet algorithme permet de tester systématiquement l'ensemble des affectations potentielles du problème. Le graphe est parcouru *en profondeur*, une variable caractérisant le problème étant choisie, on examine pour chaque affectation possible de cette variable si une solution au problème peut être construite. En cas d'impossibilité, on revient en arrière d'un étape ou plusieurs, jusqu'à trouver une nouvelle affectation qui n'a pas encore été explorée, et on examine à nouveau les suites de cette nouvelle affectation. On s'arrête s'il faudrait revenir avant l'affectation initiale car cela signifie que le graphe entier a été parcouru et qu'aucune solution n'a été trouvée.

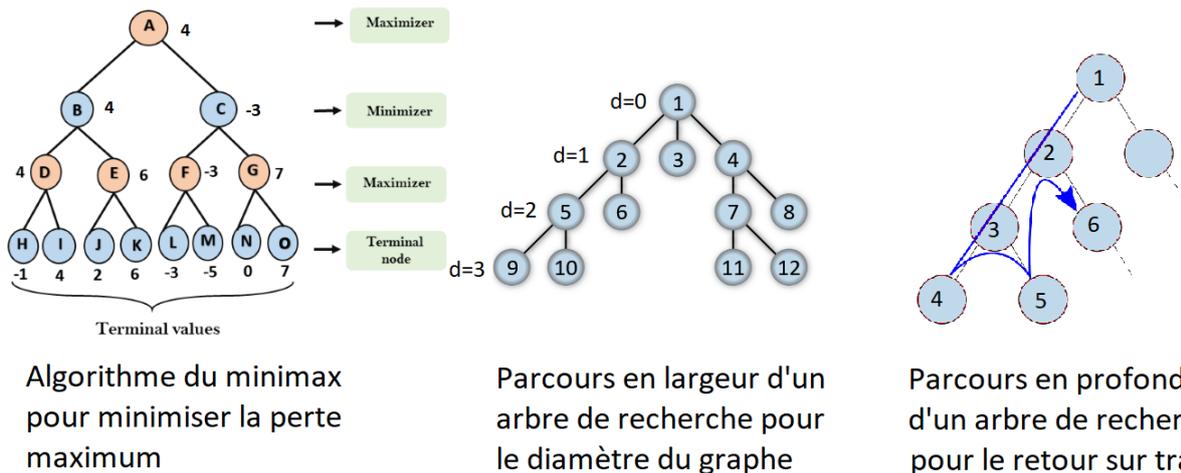


FIGURE 3.34 – Représentations schématiques des trois grands types d'algorithme utilisés dans ce chapitre.

2. John von Neumann (1903-1957) est un mathématicien et physicien américano-hongrois qui apporté d'importantes contributions notamment en mécanique quantique, en analyse fonctionnelle, en logique mathématique, en informatique théorique et en sciences économiques. Il a de plus participé aux programmes militaires américains. En informatique, on lui doit notamment l'architecture utilisée dans la quasi-totalité des ordinateurs modernes et le concept novateur d'automate cellulaire. Son théorème du minimax date de 1926.

### c. Epilogue du chemin hamiltonien

N'ayant pas complètement renoncé à découvrir un éventuel chemin hamiltonien au sein des 256 configurations indécisées du noyau central, ce sous-graphe intrigant des configurations indécisées accessibles, j'ai imaginé soudainement que si un tel chemin existait il se pourrait qu'il soit divisible en quatre parties d'une longueur égale à  $256 \div 4 = 64$ . Une de ces parties conduirait la configuration initiale à tourner d'un quart de tour vers la droite ou la gauche. Ainsi, les quatre parties amèneraient la configuration initiale à se répéter pour la 1<sup>re</sup> fois au bout de 256 étapes, après avoir été successivement tournée de quatre quarts de tour. Ce serait même un cycle hamiltonien puisque ce chemin reviendrait à la configuration initiale, et non à une configuration quelconque du noyau.

Enthousiasmé par cette idée et presque convaincu que ce cycle devait exister, je me lance dans une adaptation de mon programme de recherche pour qu'il choisisse les configurations correctement : en effet, pour que la rotation d'une partie soit possible, il faut lui garantir d'avoir à disposition toutes les rotations des soixante-quatre étapes de la 1<sup>re</sup> partie. Pour cette raison, mon programme de recherche doit sélectionner les configurations de cette 1<sup>re</sup> partie avec une grande attention. J'ai donné plus haut des éléments de la structure de ce noyau central (voir notamment la figure 3.28) :

- ♦ huit groupes de configurations ayant un élément de symétrie ne contiennent que quatre alias
- ♦ douze groupes de configurations n'ayant aucun élément de symétrie contiennent huit alias

```

      1         2         3         4
1 [[ '267.139.0', '348.179.0', '249.137.0', '168.379.0'], 1
2 [ '248.157.0', '468.579.0', '268.359.0', '246.135.0'], 2
3 [ '267.348.0', '249.168.0', '168.249.0', '348.267.0'], 3
4 [ '167.359.0', '349.157.0', '138.579.0', '279.135.0'], 4
5 [ '167.349.0', '138.279.0', '279.138.0', '349.167.0'], 5
6 [ '379.568.0', '139.256.0', '137.245.0', '179.458.0'], 6
7 [ '279.468.0', '138.246.0', '349.268.0', '167.248.0'], 7
8 [ '249.568.0', '267.458.0', '348.256.0', '168.245.0'], 8
9 [ '256.138.0', '568.279.0', '256.349.0', '568.349.0', '245.167.0', '458.279.0', '245.138.0', '458.167.0'], 9
10 [ '348.279.0', '267.138.0', '267.349.0', '249.138.0', '168.349.0', '168.279.0', '249.167.0', '348.167.0'], 10
11 [ '246.358.0', '268.459.0', '248.156.0', '468.257.0', '468.259.0', '246.158.0', '248.567.0', '268.345.0'], 11
12 [ '348.567.0', '267.358.0', '249.158.0', '348.257.0', '168.459.0', '249.156.0', '267.345.0', '168.259.0'], 12
13 [ '259.138.0', '158.279.0', '257.138.0', '567.349.0', '358.279.0', '459.167.0', '345.167.0', '156.349.0'], 13
14 [ '345.267.0', '567.348.0', '259.168.0', '358.267.0', '459.168.0', '158.249.0', '156.249.0', '257.348.0'], 14
15 [ '249.567.0', '348.259.0', '168.257.0', '168.345.0', '267.459.0', '249.358.0', '348.156.0', '267.158.0'], 15
16 [ '279.345.0', '167.259.0', '138.459.0', '349.257.0', '279.156.0', '138.567.0', '349.158.0', '167.358.0'], 16
17 [ '167.345.0', '349.156.0', '349.567.0', '279.358.0', '279.158.0', '138.259.0', '167.459.0', '138.257.0'], 17
18 [ '167.249.0', '279.348.0', '349.168.0', '279.168.0', '138.267.0', '138.249.0', '167.348.0', '349.267.0'], 18
19 [ '359.267.0', '157.348.0', '579.168.0', '157.249.0', '359.168.0', '579.348.0', '135.249.0', '135.267.0'], 19
20 [ '379.156.0', '179.358.0', '379.158.0', '139.567.0', '137.459.0', '137.259.0', '139.257.0', '179.345.0']] 20
      1         2         3         4         5         6         7         8

```

FIGURE 3.35 – Liste des 128 configurations-ordinateur du noyau central. Pour chacune, les trois premiers chiffres indiquent l'emplacement des pions de l'ordinateur, les trois suivantes celles de l'humain, le zéro final indiquant que c'est à l'ordinateur de jouer. Les sous-listes indiquent les groupes. La même liste existe pour les configurations-humain, elle commence par [ '139.267.1', '179.348.1', '137.249.1', '379.168.1' ],.

Ainsi, une 1<sup>re</sup> partie valide devrait contenir une seule configuration d'un groupe ayant quatre alias et deux configurations de chaque groupe ayant huit alias. Ne sachant quelle particularité doit caractériser l'association des deux à prendre de ces groupes de huit (doivent-elles être symétriques l'une de l'autre par rapport à un axe? ou bien tournées d'un quart ou d'un demi-tour?), je laisse le choix de toutes les combinaisons possibles du moment qu'il n'y en ait pas plus de deux par groupe. Pour les groupes ayant quatre alias d'une même configuration, le choix est encore plus simple. N'oubliant pas non plus que ce sous-graphe est biparti, c'est-à-dire qu'il existe autant de configurations-ordinateur que de configurations-humain et que des deux côtés il faut respecter ces mêmes contraintes. Cependant je n'ai aucune certitude sur les associations entre ces deux parties : y aurait-il des associations bénéfiques ou impératives? Ne voyant pas plus loin que le bout de mon nez, je laisse la liberté du choix, ce qui implique un maximum de durée de traitement... encore une fois j'ai peur de me heurter à ce fatal ennemi, le temps.

Ainsi calibré, je lance mon programme de recherche dans l'espoir de découvrir un chemin hamiltonien de soixante-quatre configurations respectant ces contraintes de choix au sein du noyau central et s'achevant sur la configuration initiale tournée d'un quart de tour (dans un sens ou dans l'autre). Très rapidement j'obtiens un chemin de quarante-neuf configurations et puis je n'assiste plus qu'à la lente destruction du chemin, l'algorithme supprimant les dernières configurations agrégées au chemin pour le reconstruire inlassablement d'une autre façon, jusqu'à constater qu'il ne peut aller plus loin et se résout à la redescendre à nouveau en détruisant progressivement jusqu'à enlever les configurations initialement agrégées. Je me suis armé de patience pour ce travail sisyphien que l'ordinateur exécute sans défaillance jusqu'à enlever la 5<sup>e</sup> configuration initialement agrégée.

Comme il lui a fallu une semaine pour en arriver là, j'ai l'impression qu'il lui faudra encore un bon mois pour achever le travail, que ce soit sur un échec ou une réussite plus précoce. Cependant un évènement imprévu interrompt brutalement cette recherche : Windows décide surnoisement de redémarrer mon ordinateur, ce qui porte un coup fatal à mon projet. Positivant la situation, je médite la leçon de cette intervention pseudo-divine et concocte des ajustements méthodologiques pour contourner cet obstacle.

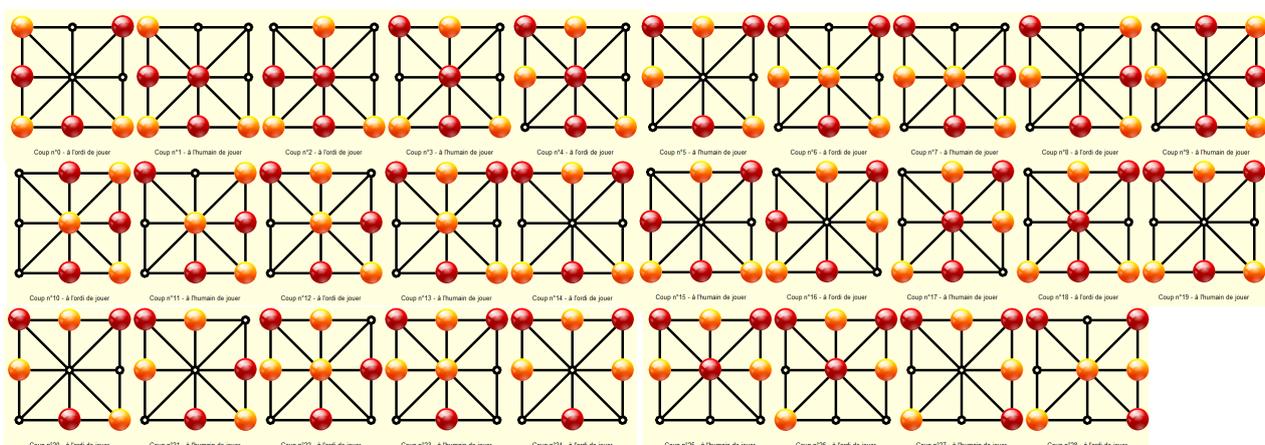


FIGURE 3.36 – Fruit d'une tentative avortée de découvrir un chemin hamiltonien de soixante-quatre configurations indécisées du noyau central pour aboutir sur la configuration initiale tournée de 90°, ce chemin incomplet en quarante-neuf configurations obéit aux contraintes mais ne peut être poursuivi sans y déroger.

Je me souviens avoir lu dans l'autobiographie d'Alexandre Grothendieck qu'il se glissait en imagination dans les structures mathématiques qu'il explorait afin d'en comprendre la structure interne, ses symétries, et d'en découvrir ses lois. J'essaie de l'imiter sur ce point de méthode et laisse courir mon imagination sur les chemins de la symétrie espérant l'intuition qui fera avancer cette recherche ou, éventuellement, la rendra obsolète. Ce qui vient surtout, ce sont des questions sans réponse : pourquoi devrait-il y avoir un cycle hamiltonien ? À supposer qu'un tel cycle existe, pourquoi devrait-il se scinder obligatoirement en quatre parties semblables mais indépendantes ?

Une question m'interpelle à laquelle je peux répondre : puis-je décrire plus précisément ce graphe indécis ? J'ai déjà remarqué qu'il était biparti, orienté et chaque configuration-humain est la réplique en négatif d'une configuration-ordinateur. Tout chemin ou cycle hamiltonien, s'il en existe, peut être dédoublé en deux versions, l'une « positive » et l'autre « négative ». Voici le début de la chaîne hamiltonienne illustrée ci-dessus dans ses versions :

- ♦ Positive : 348.179.0, 458.179.1, 458.279.0, 158.279.1, 158.249.0, 138.249.1
- ♦ Négative : 179.348.1, 179.458.0, 279.458.1, 279.158.0, 249.158.1, 249.138.0

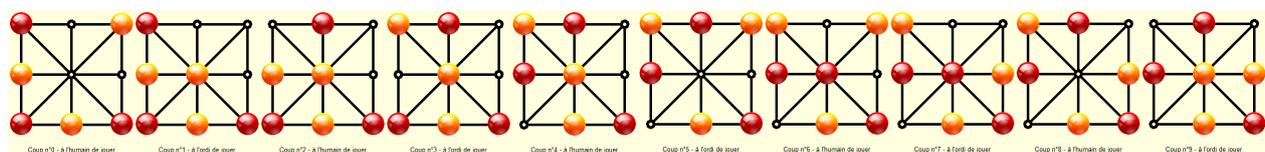


FIGURE 3.37 – Les dix premières configurations du chemin hamiltonien de la figure précédente, mais en version « négative » (inversion des joueurs).

Fort de ces avancées, je m'aperçois que je ne dispose pas encore d'une table d'incidence qui m'indiquerait les successeurs de chaque configuration. Jusqu'ici je faisais recalculer au programme les successeurs d'une configuration. Cette table devrait pourtant m'être précieuse pour mieux analyser mon problème. Pour la rendre plus lisible pour l'observateur humain que je suis, je vais numéroter les configurations avec deux nombres : le premier entre 1 et 20 indique le groupe de configurations et le second entre 1 et 4 (pour les huit premiers groupes) ou entre 1 et 8 (pour les autres groupes) indique quel alias est utilisé. La figure 3.35 montre ces nombres, en rouge pour le premier et en bleu pour le second. Si j'adjoints une lettre, O ou H selon le joueur qui doit jouer, j'obtiens une codification utilisable des configurations indécises du jeu. La chaîne hamiltonienne de la figure 3.36 est entièrement décrite par la succession des codes suivante : O1-2 H6-4 O9-6 H17-5 O14-6 H10-4 O16-3 H14-5 O3-3 H8-1 O9-4 H18-3 O12-5 H13-6 O18-1 H15-1 O13-4 H5-4 O4-1 H19-1 O10-3 H17-3 O14-2 H10-8 O7-4 H2-1 O19-2 H12-4 O13-3 H18-5 O15-8 H13-2 O5-3 H4-3 O19-6 H1-2 O6-4 H9-8 O18-7 H3-4 O8-2 H9-6 O17-5 H14-6 O10-4 H7-2 O2-4 H19-8 O12-7

Je peux dresser maintenant ma table d'incidence. Dans la figure qui suit j'ai coloré en jaune le fond des cases où sont écrits les codes des configurations utilisées par le joueur O (une même table peut être dressée pour les configurations utilisées par le joueur H, comme le 2<sup>e</sup> joueur utilise la configuration initiale du 1<sup>er</sup> joueur au 35<sup>e</sup> coup je ne peux colorer les configurations utilisées par le joueur H sur la même table). Ce coloriage me permet de vérifier que mon programme a bien respecté les contraintes. Il me permet aussi d'identifier quelles configurations n'ont pas été atteintes par ce chemin :

groupe des configurations-ordinateur	1	2	3	4					Manque
1	6-2 20-4 20-7	6-4 20-2 20-8	6-3 20-5 20-6	6-1 20-3 20-1					0
2	19-2 19-4	19-3 19-6	19-5 19-1	19-8 19-7					0
3	8-3 10-1 10-8 12-1 12-4	8-4 10-6 10-5 12-5 12-8	8-1 10-4 10-7 12-3 12-6	8-2 10-2 10-3 12-2 12-7					0
4	19-1 19-5	19-4 19-2	19-6 19-3	19-7 19-8					0
5	4-2 17-3 17-2 18-3 18-8	4-4 17-4 17-5 18-4 18-2	4-3 17-6 17-8 18-5 18-6	4-1 17-7 17-1 18-1 18-7					0
6	9-2 9-4	9-3 9-1	9-5 9-7	9-6 9-8					0
7	2-2 11-5 11-4	2-4 11-1 11-6	2-3 11-2 11-8	2-1 11-7 11-3					0
8	9-4 9-2	9-8 9-6	9-1 9-3	9-7 9-5					0
9	7-2 17-6 18-5	7-1 17-4 18-4	7-3 17-2 18-8	7-3 17-3 18-3	7-4 17-1 18-1	7-1 17-5 18-2	7-2 17-8 18-6	7-4 17-7 18-7	Manque
10	5-3 7-1 16-1 17-4	5-2 7-2 16-6 17-8	5-4 7-3 16-4 17-3	5-2 7-2 16-3 17-6	5-4 7-3 16-7 17-2	5-3 7-1 16-5 17-5	5-1 7-4 16-2 17-7	5-1 7-4 16-8 17-1	0
11	14-4	14-5	14-7	14-8	14-3	14-6	14-2	14-1	0
12	13-4	13-5	13-2	13-3	13-6	13-8	13-7	13-1	0
13	5-2 18-6	5-3 18-4	5-2 18-5	5-4 18-8	5-3 18-2	5-1 18-1	5-1 18-7	5-4 18-3	0
14	3-1 10-3	3-4 10-8	3-3 10-6	3-1 10-2	3-3 10-5	3-2 10-4	3-2 10-7	3-4 10-1	0
15	13-4	13-1	13-3	13-7	13-6	13-5	13-8	13-2	1
16	14-1	14-3	14-5	14-8	14-7	14-2	14-6	14-4	1
17	14-1	14-7	14-2	14-4	14-6	14-3	14-5	14-8	1
18	1-3 3-2 12-6 15-1	1-2 3-4 12-4 15-2	1-4 3-3 12-5 15-4	1-4 3-3 12-8 15-3	1-1 3-1 12-2 15-8	1-3 3-2 12-3 15-6	1-2 3-4 12-1 15-7	1-1 3-1 12-7 15-5	0
19	1-1 10-3 12-2	1-2 10-8 12-4	1-4 10-6 12-5	1-3 10-7 12-3	1-4 10-5 12-8	1-2 10-1 12-1	1-3 10-4 12-6	1-1 10-2 12-7	0
20	13-8	13-5	13-2	13-4	13-6	13-1	13-3	13-7	2
	1	2	3	4	5	6	7	total	7

FIGURE 3.38 – Table des successeurs des 128 configurations-ordinateur ; sur fond jaune, les cases empruntées par le chemin hamiltonien de la figure 3.36. On constate qu'il manque 7 configurations pour achever l'objectif qui était fixé.

Cette table m'apporte surtout, et de façon quasi immédiate, la raison de l'impossibilité du projet de découvrir un cycle ou une chaîne hamiltonienne incluant les 256 configurations du noyau central : pour atteindre les configurations du groupe 20 il n'y a que les configurations du groupe 1 qui conviennent. Or celles-ci, du fait de leur axe de symétrie, ne sont que quatre et il faudrait atteindre les huit configurations du groupe 20... Cela est impossible sans repasser deux fois au moins par une même configuration du groupe 1. De même, pour atteindre les 8 configurations du groupe 11 il n'y a que les 4 configurations du groupe 7 qui conviennent ce qui conduit forcément à des répétitions.

Le noyau central n'est pas un graphe hamiltonien. On ne peut pas le parcourir en entier sans passer deux fois par une même configuration. Cet argument clôture ce chapitre et m'incite à ne pas négliger le raisonnement. Persuadé qu'une chose existe on peut espérer la découvrir en utilisant la puissance de l'ordinateur qui explore des milliards de chemins sans trop de peine, mais pour éviter de longs et inutiles calculs il est plus prudent de s'assurer par tous les moyens qu'on ne peut, à l'avance et par le moyen simple du raisonnement, prouver son inexistence.