

Travail de maturité - Puissance4IA

Torrenté Florian

5 septembre 2021

Table des matières

1	Introduction	2
1.1	Description et règles du jeu	2
1.2	L'intelligence artificielle	2
1.3	Objectifs	3
1.4	Lexique	3
1.5	Théorie	3
2	Déroulement du travail	6
2.1	Mise en place	6
2.2	Implémentation du Puissance 4	6
2.3	Implémentation de l'algorithme Minimax	6
2.4	Améliorations de l'algorithme	6
2.5	Tests grandeur nature	6
3	Conclusion	6
3.1	Atteinte des objectifs	6
3.2	Ouverture vers le futur	6
3.3	Remerciements	6
4	Bibliographie	7
4.1	Livres	7
4.2	Sites web	7
5	Annexes	7

1 Introduction

1.1 Description et règles du jeu

Le Puissance 4 est un jeu avec des règles très simples. Le but du jeu est d'aligner quatre pions de même couleur (horizontalement, verticalement, ou en diagonale). Le terrain de jeu est une grille de 7x6 (sept colonnes et six rangées). Chaque joueur possède les pions d'une couleur (généralement jaune et rouge). Chacun son tour, les joueurs déposent un pion dans la colonne de leur choix, le pion descend alors le plus bas possible dans la colonne. Le premier joueur à aligner quatre pions de sa couleur gagne. S'il n'y a plus de place pour jouer, la partie est nulle.



FIGURE 1 – Un exemple de partie gagnée par le joueur rouge.

Les règles sont résolument simples mais il y a une raison supplémentaire pour laquelle j'ai choisi ce jeu : c'est un jeu à information complète. Cela veut dire que chaque joueur connaît :

- tous les coups qu'il peut jouer ;
- tous les coups que son adversaire peut jouer ;
- les gains résultants de ces actions ;
- le but de l'autre joueur.

1.2 L'intelligence artificielle

L'humanité s'est donné le nom scientifique **homo sapiens**—l'homme sage—parce que nos capacités mentales sont extrêmement importantes pour nous et notre sentiment d'identité. Le domaine de l'intelligence artificielle (ou IA) tente de comprendre cette intelligence. C'est pourquoi l'étudier peut nous permettre d'en apprendre davantage sur nous-même. Contrairement à la philosophie et à la psychologie, qui s'intéressent aussi à l'intelligence, l'IA essaye de *construire* des entités intelligentes et de les comprendre. Une autre raison d'étudier l'IA

est que ces entités construites sont intéressantes et utiles en elles-mêmes. En effet, ces dernières ont donné naissance à de nombreux résultats significatifs et impressionnants.

Maintenant nous savons pourquoi l'IA est intéressant et important, nous avons toujours besoin de savoir précisément *ce que c'est*. On pourrait simplement dire : "Eh bien, ça a à voir avec les programmes intelligents", mais je pense qu'il est important bien définir des objectifs pour pouvoir les atteindre. Une définition plus cohérente pour moi serait celle-ci : "L'intelligence artificielle a pour objectif de construire des dispositifs simulant les processus cognitifs humains"¹

1.3 Objectifs

Dans ce travail, je voulais renforcer mes connaissances sur les différents langages webs (HTML, CSS3 et JavaScript) mais surtout me rapprocher du domaine de l'intelligence artificielle qui à l'air d'être un sujet très prometteur pour le future.

1.4 Lexique

Implémentation: Réalisation (d'un produit informatique) à partir de documents, mise en œuvre, développement.

Long: En programmation informatique, un entier long (en anglais long integer) est un type de données qui représente un nombre entier pouvant prendre plus de place sur une même machine qu'un entier normal.

RAM: En anglais *Random Access Memory* et en français **mémoire vive**, est la mémoire utilisée pendant le traitement de donnée. En opposition à la mémoire dite morte qui ne sert qu'à stocker des données.

1.5 Théorie

Pour implémenter cet algorithme, il faut définir une structure de donnée très légère et optimisée afin de pouvoir calculer des milliers de parties en quelques secondes. En informatique, un *long integer* est composé de huit octets, c'est à dire 64 bits. On numérote les bits de ce nombre de droite à gauche (sans oublier qu'en informatique on commence à compter à 0)

1 63 62 61 60 ... 48 47 46 45 ... 3 2 1 0

Comme on l'a vu précédemment, le puissance4 se joue sur un *board* vertical, avec sept colonnes et six lignes, ce qui fait 42 cases. Comme on peut le voir sur le diagramme ci-dessous, on ajoute une ligne en haut et deux colonnes à droite pour des raisons de calculs plus tard. Cette représentation est appelée *bitboard*

1. HAIECH, *Parcourir l'histoire de l'intelligence artificielle, pour mieux la définir et la comprendre*.

	6	13	20	27	34	41	48	55	62	Ligne supplémentaire
+-----+										
	5	12	19	26	33	40	47	54	61	Ligne du haut
	4	11	18	25	32	39	46	53	60	
	3	10	17	24	31	38	45	52	59	
	2	9	16	23	30	37	44	51	58	
	1	8	15	22	29	36	43	50	57	
	0	7	14	21	28	35	42	49	56	Ligne du bas
+-----+										

FIGURE 2 – Représentation du bitboard

Les nombres indiquent la position dans la représentation binaire d'un long.
Prenons un exemple en représentant cette position :

.
.
.
.	.	.	0	.	.	.
.	.	.	X	X	.	.
.	.	0	X	0	.	.

0	1	2	3	4	5	6
						Colonne

FIGURE 3 –

Comme le jeu est joué par deux joueurs, on utilisera un bitboard par joueur.

								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	13	20	27	34	41	48
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	12	19	26	33	40	47
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	11	18	25	32	39	46
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	10	17	24	31	38	45
.	.	.	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	9	16	23	30	37	44
.	.	.	X	X	.	.	.	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	8	15	22	29	36	43
.	.	0	X	0	.	.	.	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	7	14	21	28	35	42
-----						Joueur 1						Joueur 2																	
0	1	2	3	4	5	6																							

FIGURE 4 –

On peut aussi le voir "à plat", ce qui ressemble à ceci :

...	0000000	0000000	0000010	0000011	0000000	0000000	0000000	//	Joueur 1
...	0000000	0000000	0000001	0000100	0000001	0000000	0000000	//	Joueur 2
	col 6	col 5	col 4	col 3	col 2	col 1	col 0		

FIGURE 5 – Représentation des bitboards "à plat"

Cette manière de représenter les états du jeu est la clé à la vitesse de l'algorithme. En effet, cette représentation permet en utilisant seulement des opérations de base de jouer des coups, de tester s'il y a une victoire, une nulle, ect... Enfin, cette représentation est très légère et permet donc un stockage assez simple et demande très peu de RAM pour fonctionner.

Maintenant qu'on a vu comment les boards sont représentés, on va pouvoir explorer comment on va traiter ces données, pour jouer des coups et tester s'il y a une victoire. Pour ce faire on n'a besoin que de deux types d'opérations : le *bit shifting* (décalage de bit) et la combinaison de bit (notamment XOR et AND).

Le décalage de bits

Pour décaler des bits en informatique, on utilise 2 opérateurs : `>>` et `<<`. Respectivement *right shift* et *left shift*. Par exemple `0b10101110 << 3` signifie que l'on retire les 3 premiers nombres à gauche et on ajoute 3 zéros à droite. Ici, le préfixe `0b` indique que les chiffres suivants représentent un nombre binaire et non un nombre décimal. Donc `0b10101110 << 3 = 0b01110000`. De manière similaire, `0b10101110 >> 3 = 0b00010101`.

La combinaison de bits

Les deux seuls opérateurs dont nous aurons besoin sont le *XOR* et le *AND*. L'opérateur XOR, ou OR exclusif, (qu'on écrit `^`) prend deux nombres binaires, par exemple `0b10101110 ^ 0b10011111`, et compare les bits de même position. Si les deux bits sont différents, le résultat est 1, sinon c'est 0.

Si on écrits les deux nombres l'un sur l'autre, les bits sont facile à comparer.

```

10101110
^ 10011111
-----
00110001

```

FIGURE 6 – Exemple de l'opérateur XOR

L'opérateur AND (qu'on écrit `&`) prend aussi 2 nombres binaires et les compare bit par bit. Si les deux bits sont un 1, le resultat est 1 sinon 0.

```

10101110
& 10011111
-----
10001110

```

FIGURE 7 – Exemple de l'opérateur AND

2 Déroulement du travail

2.1 Mise en place

2.2 Implémentation du Puissance 4

2.3 Implémentation de l'algorithme Minimax

2.4 Améliorations de l'algorithme

2.5 Tests grandeur nature

3 Conclusion

3.1 Atteinte des objectifs

3.2 Ouverture vers le futur

3.3 Remerciements

4 Bibliographie

4.1 Livres

- [1] Stuart J. RUSSELL, Peter NORVIG et Ming-wei CHANG. *Artificial intelligence a modern approach*. Pearson Education Limited, 2021.

4.2 Sites web

- [2] Socket.IO's DEVS. *Socket.IO documentation*. 2021. URL : <https://socket.io/docs/v4>.
- [3] Jacques HAIECH. *Parcourir l'histoire de l'intelligence artificielle, pour mieux la définir et la comprendre*. Oct. 2020. URL : https://www.medecinesciences.org/en/articles/medsci/full_html/2020/08/msc200112/msc200112.html.
- [4] Ye QIANQIAN et Evelyn MASSO. *p5.js Reference*. 2021. URL : <https://p5js.org/reference/>.
- [5] Node.js's TEAM. *Node.js v16.6.1 documentation*. 2021. URL : <https://nodejs.org/api/>.

5 Annexes