

Intelligence Artificielle pour un jeu de Dames



1

Principe général d'une Intelligence Artificielle pour un jeu à 2 joueurs

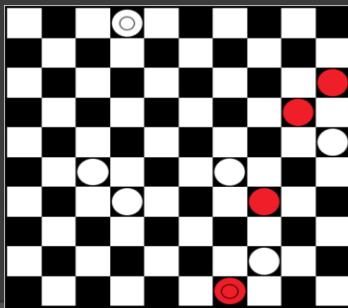
- Anticiper le plus possible



2

Quelques rappels des algorithmes

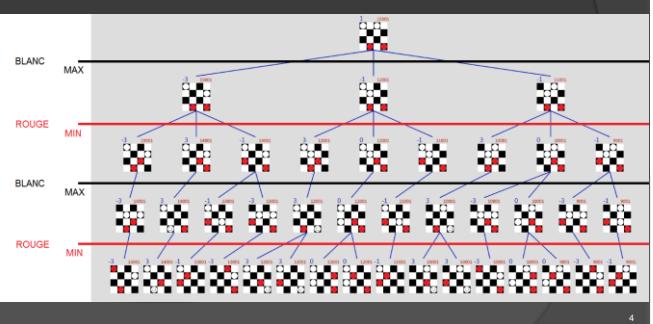
- Fonction de coût :



3

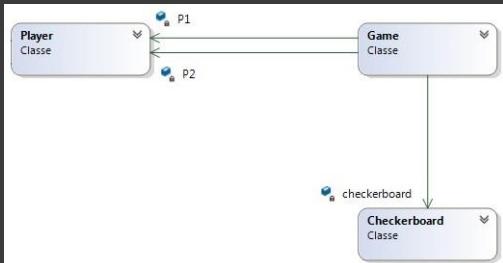
Quelques rappels des algorithmes

- Présentation du Min-Max



Implémentation du jeu

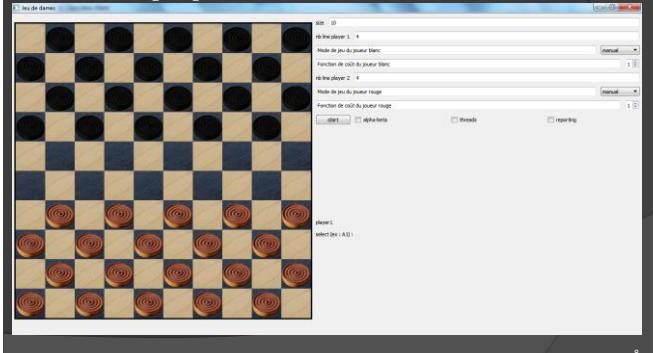
- Schéma UML :



- Structure CHILD

Implémentation du jeu

- Rendu Graphique :



8

Optimisations

- Utilisation de l'Alpha-Beta

Profondeur	Nega-Max	Alpha-Beta	% de temps gagné
2	0,088 s	0,071 s	19%
4	5,341 s	2,428 s	68%
6	216,333 s	57,73 s	74%

9

Optimisations

- Utilisation d'Open MP :

```

#pragma omp parallel for
for(n=0;n<8;n++){
    printf("Element %d processed
by the thread %d \n",n
,omp_get_thread_num());
}
  
```

10

Optimisations

Element 0 processed by the thread 0
Element 4 processed by the thread 1
Element 1 processed by the thread 0
Element 5 processed by the thread 1
Element 2 processed by the thread 0
Element 6 processed by the thread 1
Element 3 processed by the thread 0
Element 7 processed by the thread 1

11

Optimisations

- Utilisation des threads pour paralléliser les calculs :

Profondeur	Nega-Max	Nega-Max Thread
2	0,088 s	0,044 s
4	5,341 s	3,04 s
6	216,333 s	121,328 s

12

Optimisations

- Utilisation d'autres fonctions de coût
 - Fonction de coût 1 : Compte le nombre de pièces
 - Fonction de coût 2 : Garde les pièces groupées
 - Fonction de coût 3 : Envoie les pièces vers le centre
 - Fonction de coût 4 : Fonction aléatoire

13

Optimisations

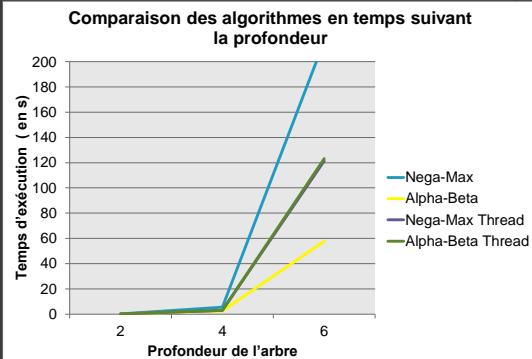
- Utilisation d'autres fonctions de coût

Confrontations	1	2	3	4
1	$\frac{1}{2}$	0	1	1
2	1	$\frac{1}{2}$	1	1
3	0	0	0	$\frac{1}{2}$
4	0	0	1	0

• 0 : défaite $\frac{1}{2}$: match nul 1 : victoire

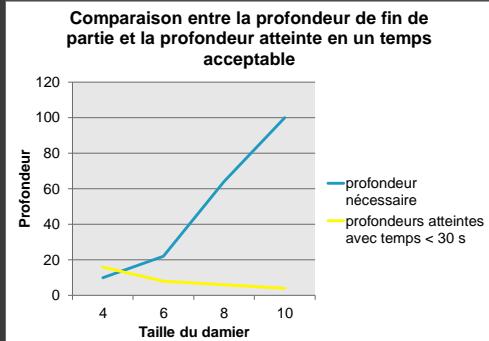
14

Bilan



15

Bilan



16

Conclusion

- Projet abouti
- Améliorations restantes
 - Auto-apprentissage
 - Sauvegarde des calculs précédents
 - Situations types
 - Parcours anticipé des arbres

17