Rapport du projet

Implantation du jeu de dames

Réalisé par :

Abdessalam SEMAOUI

Anis REDJEM SAAD

Lounis RAHMANI

Responsable :

Mme Selmi

Table des matières

I- **Introduction** 3

II- **Analyse et conception** 3

II-1- Analyse du travail à réaliser 3

II-2- Conception et réalisation 4

II-3- Modélisation du projet 6

III- **Structure de projet** 7

IV- **Réalisation et implémentation** 9

IV-1- Algorithmes implémentés 9

IV-1-1- Structure de l'arbre 9

IV-1-2- Description des heuristiques 10

IV-1-3- Algorithmes du cours 11

IV-1-3-1-Algorithme AlphaBeta 11

IV-1-3-2-Algorithme NégaMax 12

IV-1-3-3-Algorithme AlphaBeta(Vesrion NégaMax) 13

IV-1-3-4-Algorithme SSS* 13

V- **Problèmes rencontrées et solutions apportées** 15

VI- **Manuel d'utilisation** 16

VII- **Conclusion** 21

I. Introduction :

Le jeu de dames est un célèbre jeu de réflexion joué a deux ,se composant d'un damier et d'une vingtaine de pions dont le but est de manger ou d'immobiliser toutes les pièces du joueur adverse. Même si le but du jeu de dames reste inchangé il existe toutefois différentes manières d'y jouer, notamment avec les règles internationales, ces règles sont données sur ce http://www.ffjd.fr/Web/index.php?page=reglesdujeu

II. Analyse et conception :

II-1- Analyse du travail à réaliser :

Le projet consiste en l'implantation du jeu des dames dans sa version internationale en utilisant les notions vues en cours à savoir :
Stratégie gagnante, Minimax, Négamax, Alpha-Béta, Alpha-Béta en version négamax, SSS*, fonction d'´evaluation, tables de transpositions.

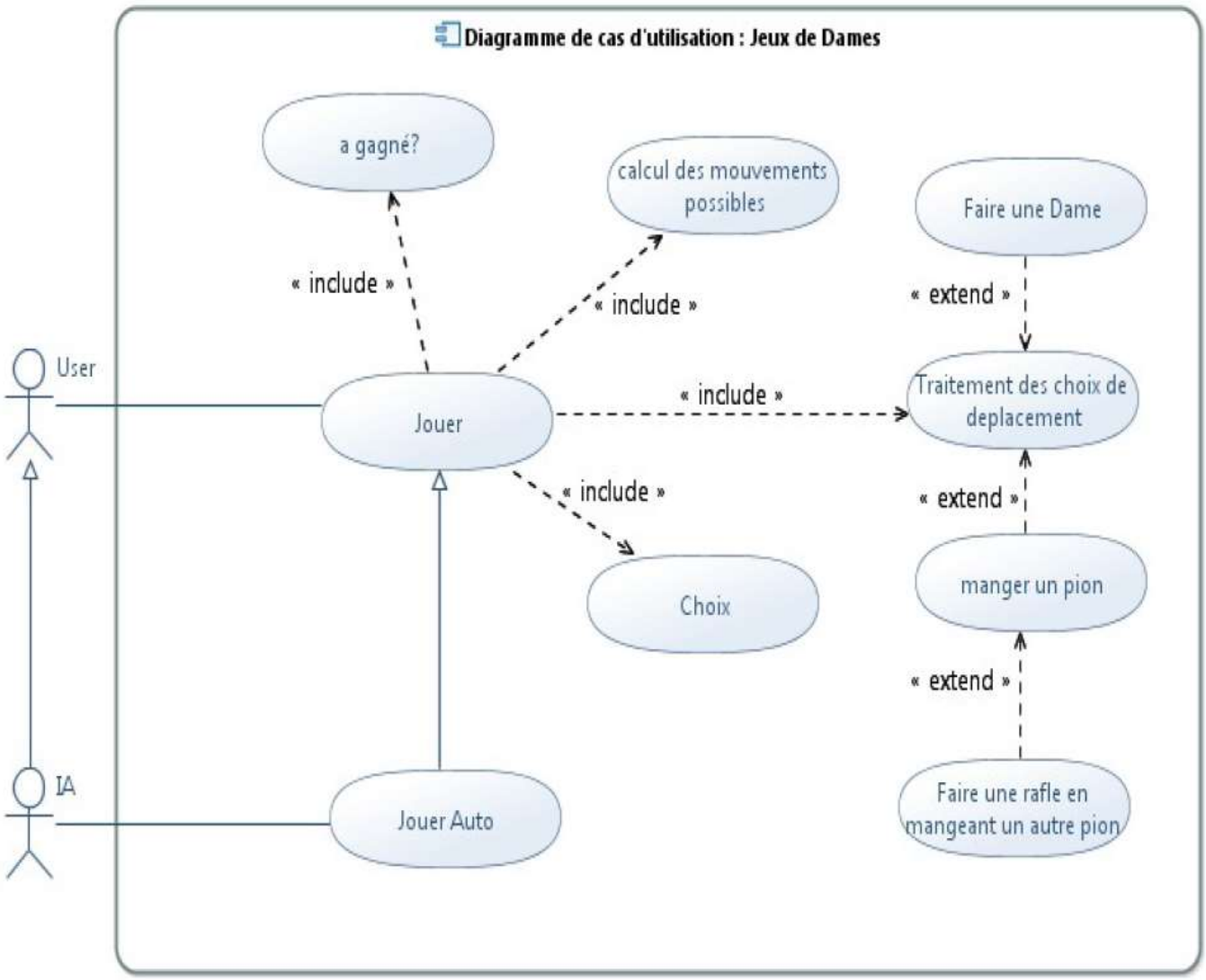
Le programme doit répondre aux besoins suivants :
Il doit permettre :

- A un joueur humain de jouer contre une IA avec au moins trois niveaux de difficultés différents, débutant, moyen et expert. Mais vous pouvez intégrer à votre programme un plus grand nombre de niveaux.

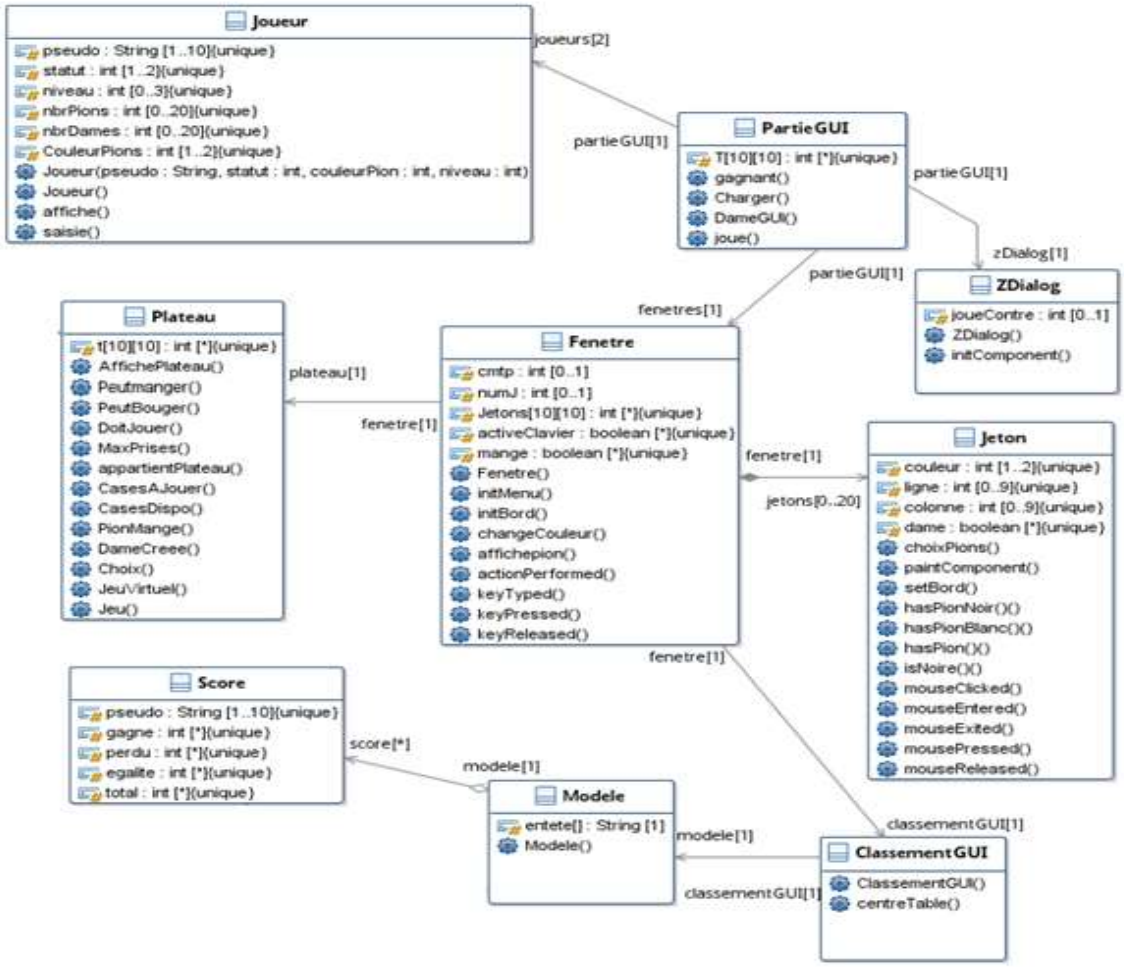
- A deux IA de jouer entre elles.

II-2- Conception du projet :

Pour s'y faire, on a commencé par le schéma des cas d'utilisation qu'on a illustré comme suivant :

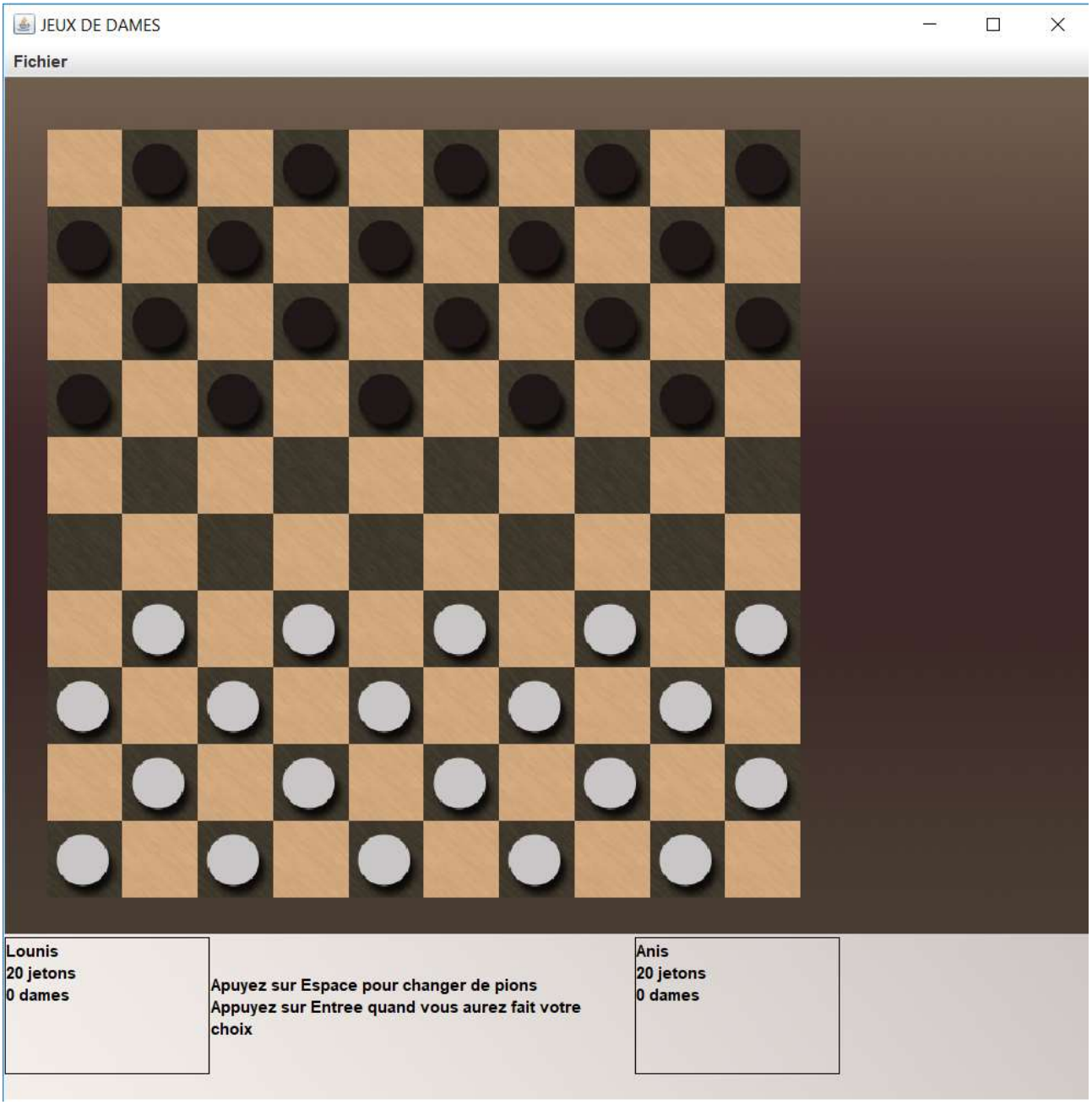


Une partie du diagramme de classe qui nous a permis de structurer notre projet (figure ci-dessous) :



II-3- Modélisation du projet :

On a décidé de modéliser notre projet de la façon suivante :



III. Structure de projet :

Il y a deux classes principales qui permettent au jeu de se dérouler de différentes manières, (il y a donc deux fonctions main). La classe **PartieConsole** fait marcher le jeu sous console et la classe **PartieGraphique** dans une fenêtre.

Ci-suit une description des principales classes utilisées :

Plateau:

La classe indispensable à notre programme. C'est dans cette classe que vont être adaptées toutes les règles du jeu via de nombreuses fonctions.
Elle est composé d'une matrice qui représentera le damier avec pour valeurs:
1: jeton blanc
10: dame blanc
2: jeton noir
20: dame noir
0: case vide

Score:

Cette classe représente un score de Joueur à savoir son pseudo, son nombre de parties jouées, gagnées, perdues. Elle est utilisée par la classe classement qui contient une **ArrayList** de scores.

Fenetre:

La classe indispensable de l'interface graphique.
Elle hérite de **Jframe** est implémente les interfaces **ActionListener** et **KeyListener**. C'est donc ici aussi que les interactions entre la souris ou le clavier et notre interface graphique pourront avoir lieu.
Elle est ordonnée grâce à des **Jpanel** et des **Layout**.

Jeton:

La classe représentant une case qui est soit vide blanche ou vide noire soit constitué d'un pion ou une dame (nous avons créé une classe héritière de **JButton**).

La case change si on change sa valeur principale «couleur»:
1: pion blanc
10: dame blanche
2: pion noir
20: dame noire
0: case vide

PartieGUI:

Cette classe a les fonctionnalités mais suivantes :

- C'est la classe qui va lancer la forme graphique du jeu, donc dans une fenêtre.

Element :

Utilisé pour définir le nœud d'un arbre, elle utilise l'attribut heuristique, vetcteur des fils, l'état damier et le type du nœud (Pour l'alogorithme SSS*, on fait passer un paramètre supplémentaire qui sont droit).

EtatDamier :

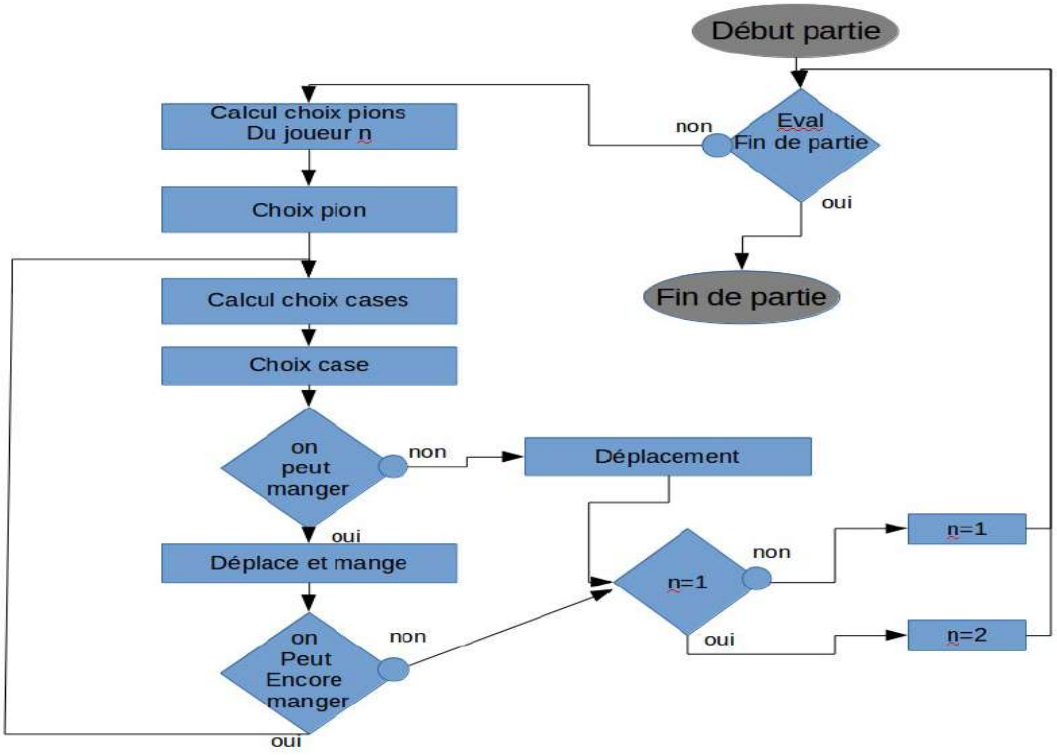
Elle hérite de la classe plateau, elle a comme paramètre le CoupJoué pour avoir atteint ce damier, et le pionjoué pour arriver a cet état.

SSSEelement :

Représente l'élément de file pour l'algorithme SSS*.

IV. Réalisation et implémentation :

Voici une représentation d'un déroulement d'une partie avec de diverses fonctions :



IV-1 Algorithmes implémentés :

Pour implémenter les différents algorithmes imposés, on a utilisé les fonctions suivantes :

IV-1-1- Structure de l'arbre :

On a utilisé la structure de l'arbre suivante pour implémenter tous les algorithmes :

Les différentes qui définissent l'arbre :

EtatDamier :

Elle hérite de la classe plateau, elle a comme paramètre le *CoupJouer :* le déplacement effectué pour avoir atteint ce damier. *PionJouer :* le pion déplacé pour arriver a cet état.

Element :
Utilisé pour définir le nœud d'un arbre, elle utilise l'attribut:
Heuristique : un entier qui définit l'heuristique du noeud
fils : Représente un vecteur d'élément qui sont des fils du nœud
EtatDamier : décrit précédemment.
Type : c'est le type du nœud

Pour l'alogorithme SSS*, on fait passer deux paramètres supplémentaires qui sont *frèredroit* et *père*).

IV-1-2- Description des heuristiques :

L'heuristique est définit en calculant la différence entre le nombre de pions de chaque joueur tout en sachant que un pion vaut 1
et une dame vaut 3
Exemple : joueur 1 possède 5 pions et une dame, et le joueur 2 possède 7 pions, l'heuristique est calculé comme suit :

(1*5+(3*1)) – (1*7) = 1

IV-1-3- Etude de la stratégie gagnante :

Après avoir calculé les heuristiques, on passe au trie des fils de la racine et on choisit le meilleur fils par rapport au type de la racine (Max/Min).

IV-1-3- Algorithmes du cours :

IV-1-3-1- Algorithme AlphaBeta :

Afin d'implémenter l'algorithme AphaBeta, on a utilisé l'algorithme donné dans le cours, la figure suivante montre l'algorithme utilisé en code java ainsi que les différentes classe utilisées décrites précédemment :

```
package Algorithmes;

public class AlphaBeta implements Algorithme {

	/// Algorithme AlphaBeta

	Public static Integer alphaBeta(Element s, Integer alpha, Integer beta) {

		// Sifeuilleretournerheuristique
		if (s.isFeuille())
			return s.heuristique;
		// Si max
		if (s.isMax()) {
			inti = 0;
			while (alpha<beta&&i<s.fils.size()) {
				// mettreajour alpha
				alpha = Math.max(alpha, alphaBeta(s.fils.elementAt(i), alpha, beta));
				i++;
			}
			s.heuristique = alpha;
			return alpha;
		// sinonsiMin
		} else {
			inti = 0;
			while (alpha<beta&&i<s.fils.size()) {
				// mettre a jour beta
				beta = Math.min(beta, alphaBeta(s.fils.elementAt(i), alpha, beta));
				i++;
			}

			s.heuristique = beta;
			return beta;
		}

	}

}
```

IV-1-3-2- Algorithme Négamax :

```
Package Algorithmes;

Publicclass NegaMax implements Algorithme {

Public static void negaMax(Element s) {

                if (s.isFeuille()) {
                        // inverser l'heuristique simin
                        if(s.type==Type.Min) s.heuristique=-s.heuristique;
                        return;
                }

                inti = 0;
                while (i<s.fils.size()) {
                                negaMax(s.fils.elementAt(i));
                                i++;
                }

                s.trierFils(); // rappel : cettefonctiontriedefaçoncroissanteoudécroissanteenfonctiondu type
dunoeud (minou max)

                //Danslesdeuxcas(minou max) prendrelavaleur maximum de l'inverse
                s.heuristique=(s.type!=Type.Max)? -s.fils.elementAt(0).heuristique:-
s.fils.elementAt(s.fils.size()-1).heuristique;

        }
}
```

IV-1-3-3- Algorithme AlphaBeta version Néga :

```
Package Algorithmes;

Public class NegAlphaBeta implements Algorithme {

	Public static int negAlphaBeta(Element s, Integer alpha, Integer beta){

		int val=Integer.MIN_VALUE;
		// si s estunefeuillealorsval= h(s) si Max sinonval=-h(s)
		if (s.isFeuille()) {
			val= (s.type==Type.Max)? s.heuristique:-s.heuristique;
		}
		// sinon
		else {
			inti=0;
			while (alpha<beta&&i<s.fils.size()) {
				val = Math.max(val, -negAlphaBeta(s.fils.elementAt(i), -beta, -alpha));
				alpha=Math.max(alpha, val);
				i++;
			}



		}
		//retournerval
		s.heuristique=val;
		return val;
	}

}
```

IV-1-3-4- Algorithme SSS* :

Dans l'alogorithme SSS*, on a crée une nouvelle classe SSSelement qui représente l'élément de la file utilisé dans l'algorithme,

```
Public class SSS implements Algorithme {


	Public static int algoSSS(Element s) {

		ArrayList<SSSelement>G= newArrayList<SSSelement>();

		G.add(newSSSelement(s,true,Integer.MAX_VALUE));
		Collections.sort(G);// on trieachaque insertion

		while (G.get(0).vivant==false) {
			//On prendle premier elementdela file
			SSSelementelement= G.get(0);
			G.remove(0);
			////////
```

```
                        // sivivantalors
                        if(element.vivant==true) {
                                //sifeuille
                                if (element.noeud.isFeuille()) {
                                        // on insere l'elementenrésolu (vivant==false)
                                G.add(newSSSelement(s,false,Math.min(element.valeur, s.heuristique)));
                                        Collections.sort(G);// on oublie pas de trier
                                }
                                // sinon
                                else
                                {
                                        ///sinoeud max on inseretouslesfils
                                        if (element.noeud.type==Type.Max) {
                                                for(inti=0;i<element.noeud.fils.size();i++) {
                                                        G.add(new
SSSelement(element.noeud.fils.elementAt(i),true,element.valeur));
                                                }
                                        }
                                        ///sinon (min) on inserele plus agauche
                                        else
                                        {
                                                G.add(new
SSSelement(element.noeud.fils.elementAt(0),true,element.valeur));
                                        }

                                        Collections.sort(G);
                                }
                        }
                        //sinonsirésolu
                        else {
                                ///si MIN

                                if (element.noeud.type==Type.Min) {
                                        //insererpere
                                        G.add(newSSSelement(element.noeud.pere,false,element.valeur));
                                        //supprimersucceseur
                                        supprimerSucceseur(G, element.noeud.pere);

                                }
                                // sinon MAX
                                else
                                {
                                        if (element.noeud.frereDroite!=null) {
                                G.add(new SSSelement(element.noeud.frereDroite,false,element.valeur));
                                        }
                                        else {
                                G.add (new SSSelement(element.noeud.pere,false,element.valeur));
                                        }

                                }

                                Collections.sort(G);
                        }
                        }

                        return (G.get(0).valeur);


                }
}
```

V. Problèmes rencontrés et solutions apportées :

Choix du langage :

Pour le choix de langage à utiliser, on avait des hésitations entre python et java, on aurait souhaité réaliser ce projet avec le langage python, mais par manque du temps, on était obligé de choisir JAVA vu qu'on avait déjà des connaissances et que sa nous facilitera la tache.

Rafle :

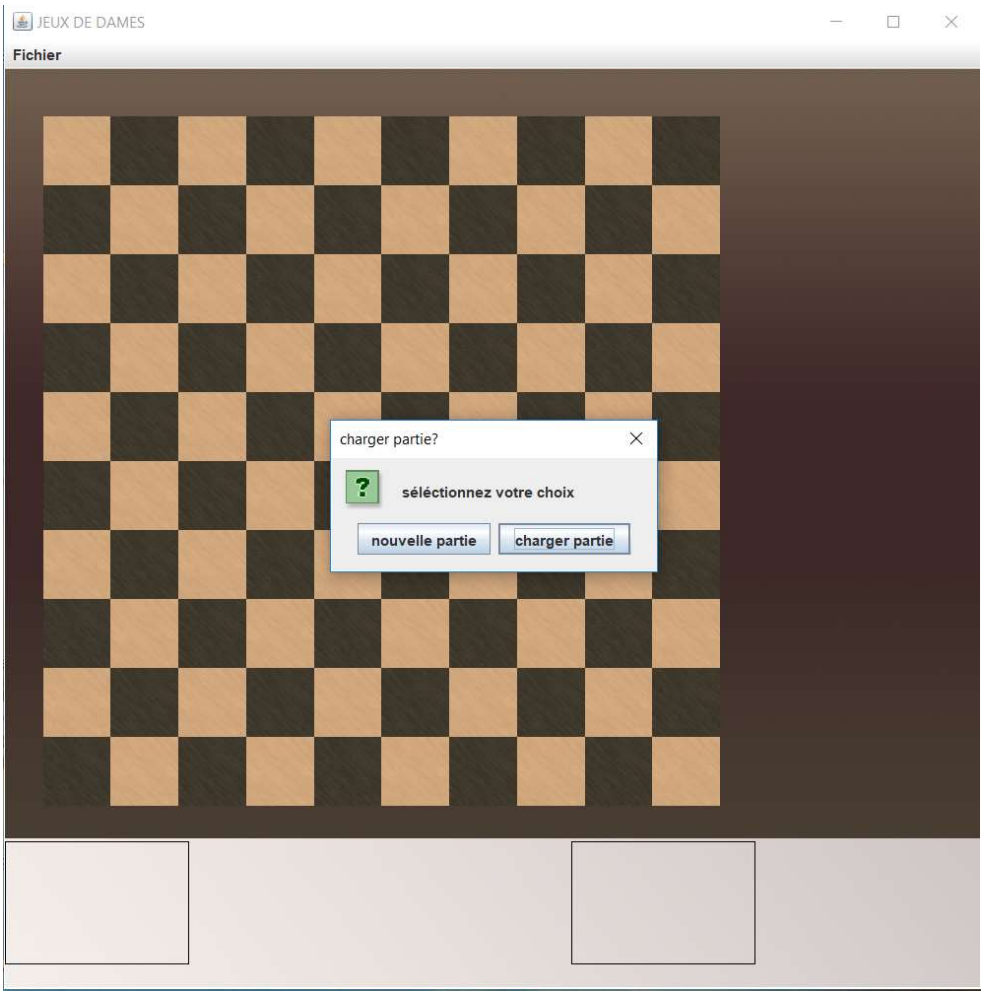
Durant les tests du jeu, nous avons rencontrés de nombreux problèmes avec les rafles, surtout celle concernant les dames. Nous avons été obligés de modifier de temps en temps certaines fonctions et même de recommencer certaines fonctions depuis le début

Intelligence artificielle:

On a rencontré quelques soucis quand a la programmation d'une intelligence artificielle avec un niveau avancé et très intelligente pour manque du temps

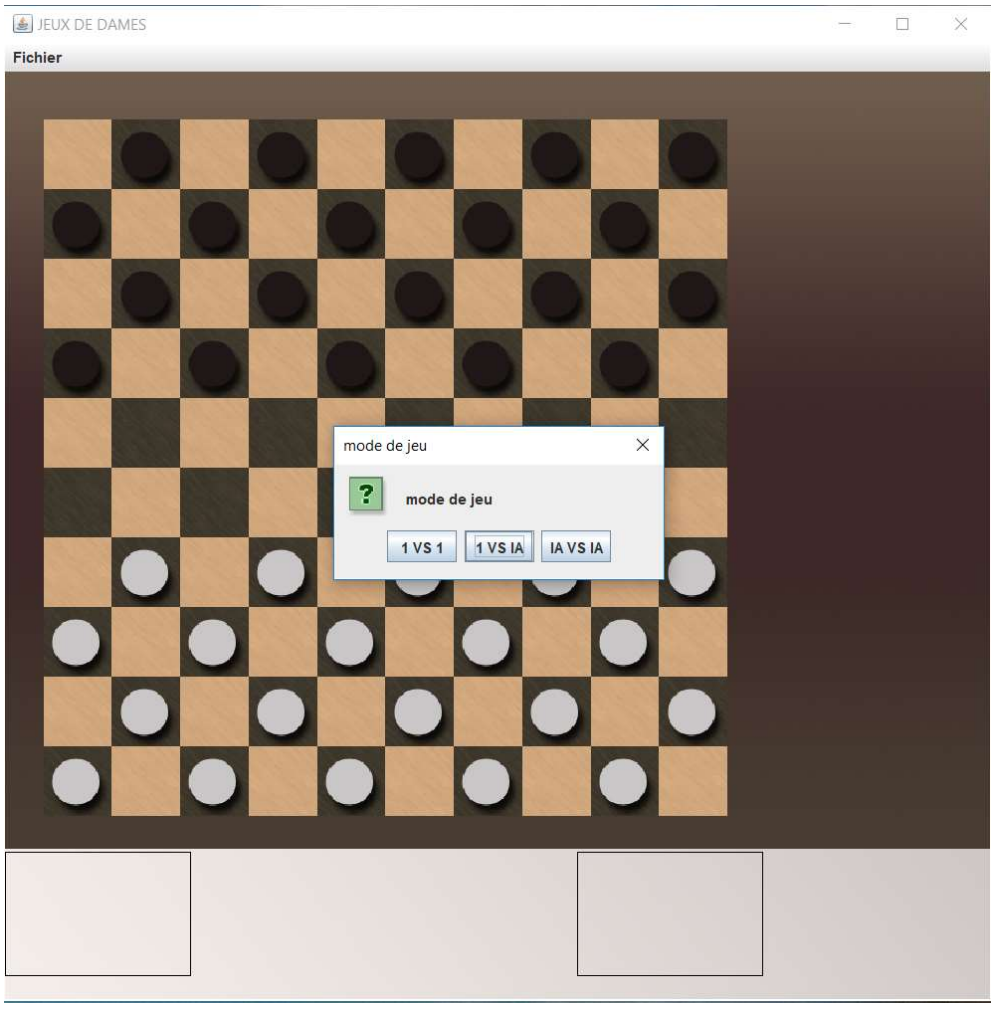
VI. Manuel d'utilisation :

- Pour lancer le jeu, il faut cliquer deux fois sur jeu.jar situé dans le dossier du projet, la fenêtre suivante apparaît :



- On devra ensuite choisir entre charger une partie déjà entamée ou bien démarrer une nouvelle

⇨ La fenêtre suivante apparait le choix d'une nouvelle partie



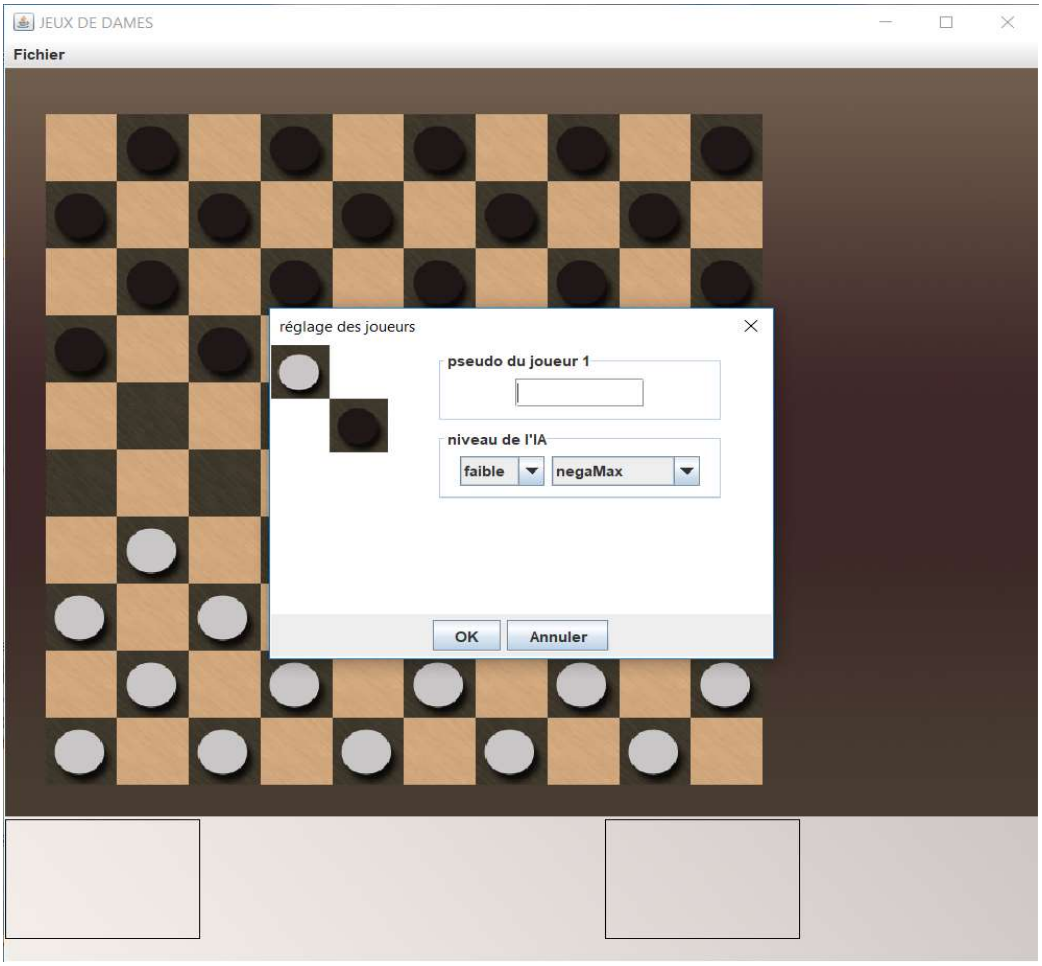
- L'étape suivante est de choisir le mode de jeu parmi les 3 existant :

⇨ 1 vs 1 : joueur contre joueur

⇨ 1 vs IA : joueur contre ordinateur

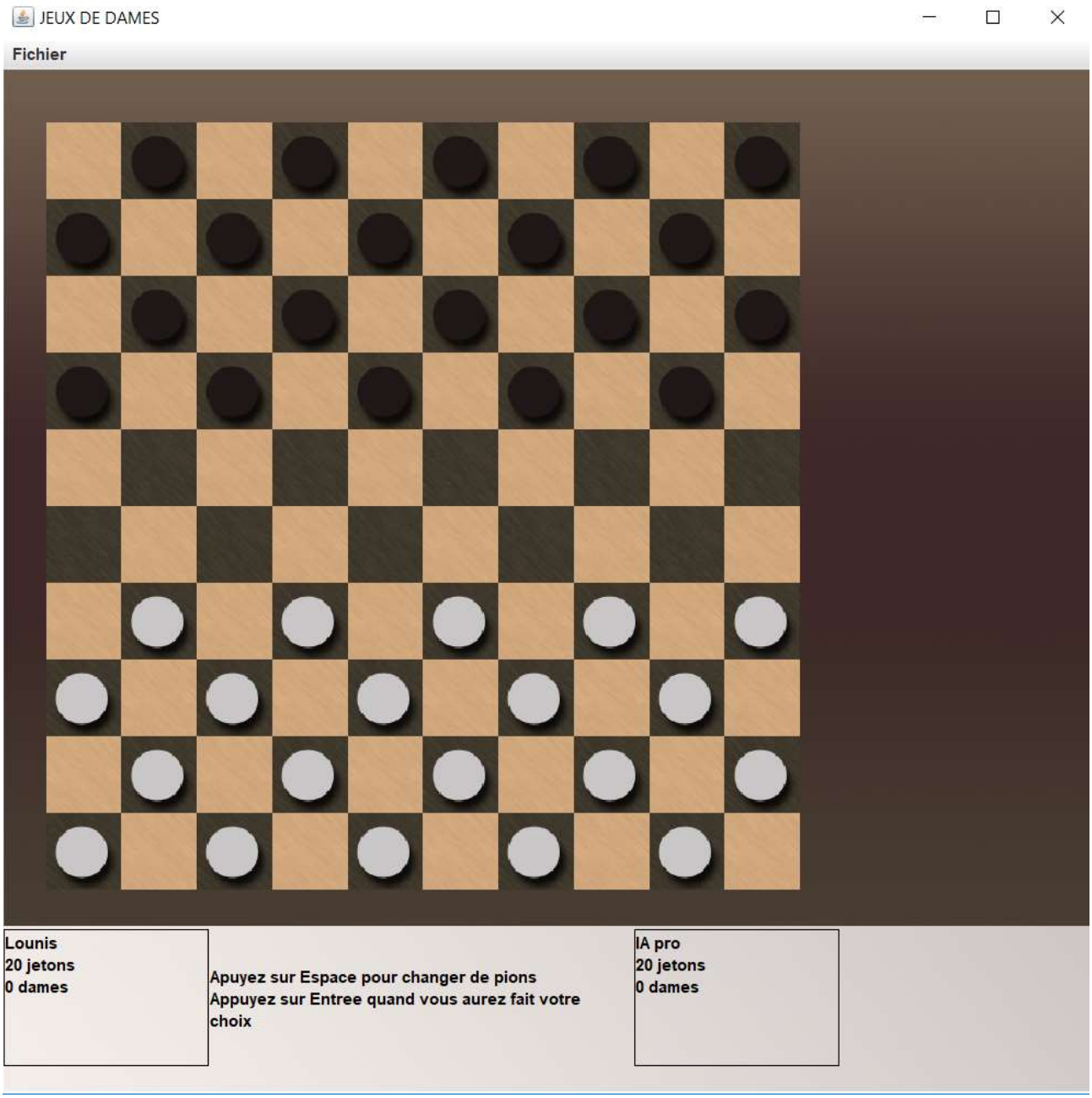
⇨ IA vs IA : ordinateur contre ordinateur

- Après le choix effectué, on aura la fenêtre suivante : (dans notre cas on a choisit 1 vs IA)

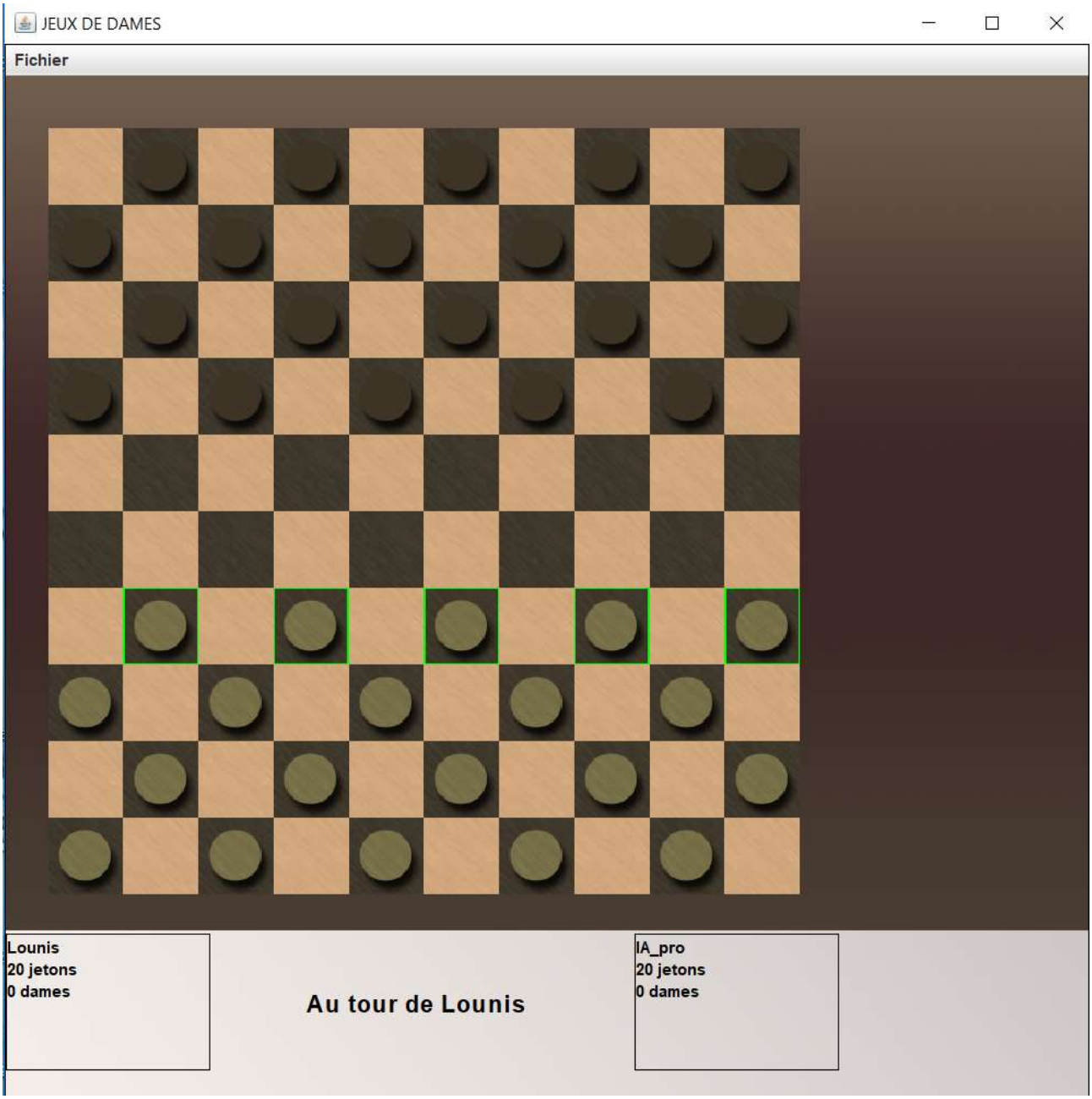


- Ici, on doit configurer le pseudo du joueur et l'algorithme de jeu voulu ainsi que le niveau de difficultés de l'intelligence artificielle

- Une fois le choix effectué, la fenêtre ci-dessous apparait, avec le nom du joueur a gauche et le nom de l'IA a droite.
- On dispose de l'option de changer la couleur des pions, et ceci en appuyant sur **Espace** (4 Modes possibles)
- Pour Commencer en suite la partie, faut appuyer sur la touche **Entrée** du clavier.



Finalement, la fenêtre du jeu sera ainsi, avec un petit box au bas milieu indiquant le tour de joueur et le compteur de jetons



VIII- Conclusion :

Au cours de ce mini projet , on a pu mettre en pratique les algorithmes vu en cours afin de concevoir une intelligence artificielle efficace et performante, et on a pu tester les différences entre les deux algorithmes qu'on a utilisé (Négamax , alpha-bêta, Néga AlphaBeta, SSS*) avec différente profondeur de l'arbre, il s'est avéré qu'avec l'élagage utilisé par alpha-bêta on arrive a un gain de performance significatif et visible au sein du jeu.