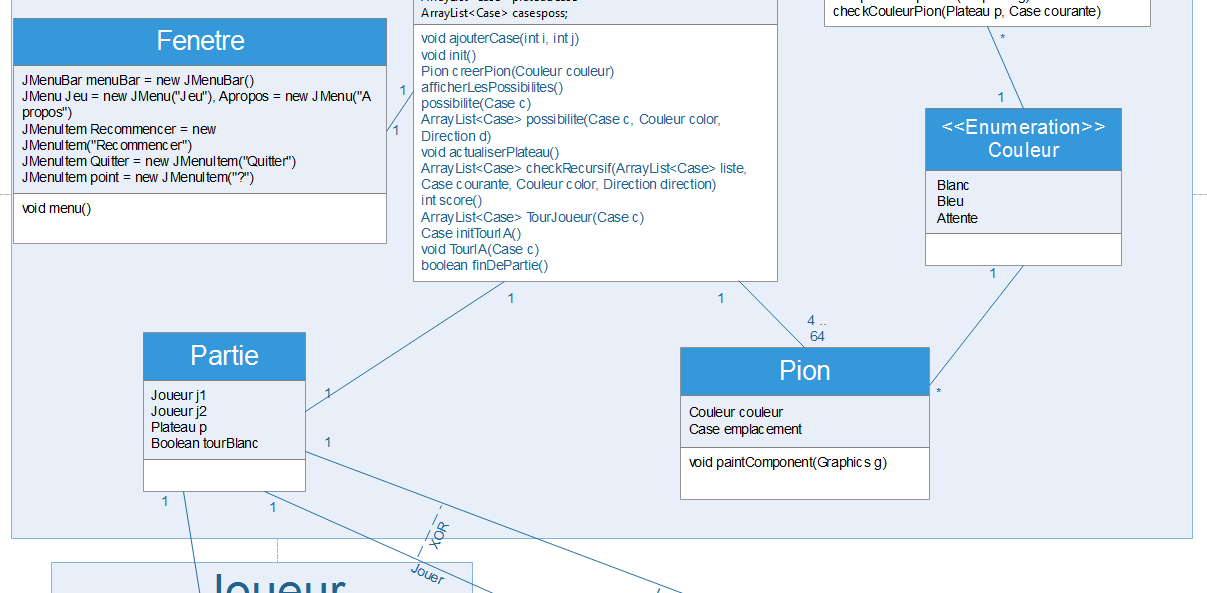
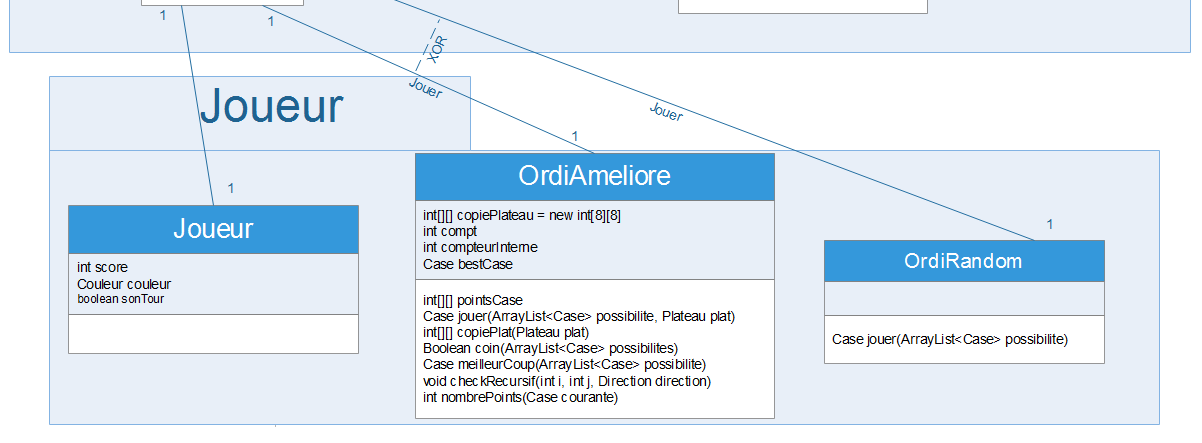
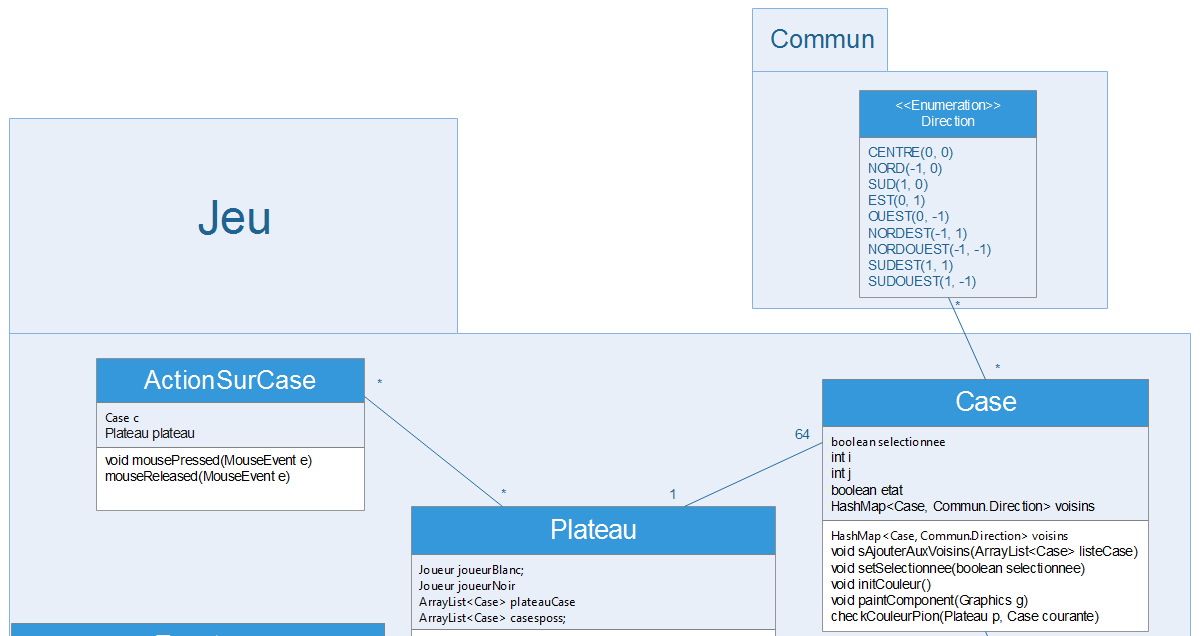
2016

15/05/2016

**PROJET 3A : REVERSI**



# Diagramme de Classe :



# Algorithme et explications de l’IA :

## Explications brèves

Notre intelligence artificielle réalise plusieurs tests avant de jouer son tour :

* Si elle a la possibilité de jouer dans un coin, elle joue immédiatement (test réalisé à l’aide

d’une matrice de points associée à la grille de cases : les coins possèdent 30 points, les autres cases, moins)

* Ensuite, elle calcule combien cela lui rapporterai de points de poser son pion sur

une des cases possibles (cases rouges). Elle posera son pion, sur celle qui mange le plus de pions bleus.

## Pseudo Code

#### Fonction de copie du Plateau : copiePlateau est une matrice 8x8 -> int[8][8]

**SET** compteur as 0

**FOR** chaque case du plateau définie par sa position (i,j)

**SET** Case c AS case(i,j) du plateau

**IF** c n’est pas vide

**IF** c possède un pion bleu

CopiePlateau[i][j] prend la valeur 2

**ELSE**

CopiePlateau[i][j] prend la valeur 1

**ENDIF**

**ELSE**

CopiePlateau[i][j] prend la valeur 0

**ENDIF**

**ENDFOR**

#### Fonction du calcul de nombre de points pour chaque possibilité p

**FOR** toutes les directions depuis p (bas-droite, haut-gauche, bas, droite…)

On passe à la case d’après

**IF** la case est vide ou avec un pion déjà blanc *//copiePlateau[i][j] vaut 0 ou 1*

**RETURN** 0 ;

**ENDIF**

**SET** compteur Interne AS 1 *//On est tombé sur le premier pion bleu*

*//Boucle récursive*

On passe à la case qui suit, toujours dans la même direction

**IF** on sort de la grille de jeu

Compteur Interne prend 1

**RETURN** ; *//On sort de la récursivité*

**ENDIF**

**IF** la case est vide

Compteur Interne réinitialisé à 0

**RETURN** ; //On sort de la récursivité

**ENDIF**

**IF** la case possède un pion bleu *//copiePlateau[i][j] vaut 2*

**INCREMENT** Compteur interne de 1

Rappel de la RECURSIVITE

**ENDIF**

**IF** la case possède un pion blanc *//copiePlateau[i][j] vaut 1*

**RETURN** ; *//On sort de la récursivité*

**ENDIF**

Compteur += Compteur Interne

Compteur Interne réinitialisé à 0

**ENDFOR**

**RETURN** compteur

#### Fonction du tour de l’ia

Appel de la **FCT** copiePlateau

**SET** bestCase AS null

**IF** il y a des possibilités

**IF** une des possibilités est un coin *//case à 30 points*

**RETURN** ce coin ;

**ELSE**

**SET** score AS 0

**FOR** chaque case possible c

Appel de la **FCT** nombreDePoints

**IF** score < nombreDePoints

BestCase prend c

Score prend nombreDePoints

**ENDIF**

**ENDFOR**

**ENDIF**

**ENDIF**

**RETURN** bestCase

## Difficultés rencontrées

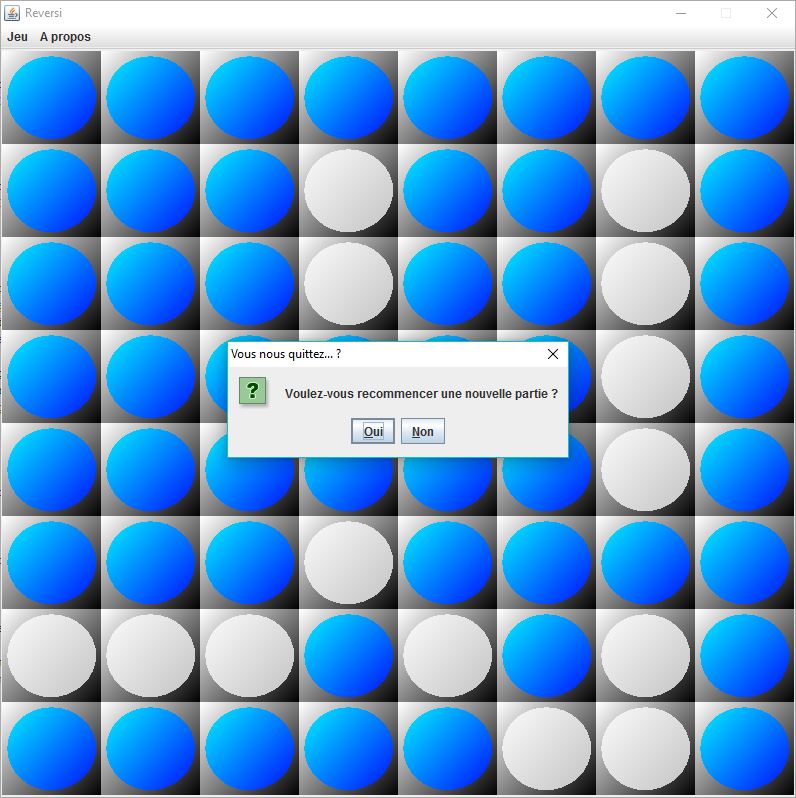
Nous avons eu des difficultés à mettre en place notre première IA, d’une part par manque de connaissances et d’expériences, d’autre part parce que nous avions du mal à visualiser l’algorithme en profondeur.

Nous avons donc réalisé une intelligence artificielle simple, qui choisit au mieux ses positions. Sa profondeur est faible, contrairement par exemple à un algorithme minmax, mais par manque de temps nous n’avons pas pu creuser en profondeur.

Améliorations à envisager :

* Relier les points de chaque case au choix de la case qui mange le plus de pions bleus
* Ajouter des niveaux de profondeur pour aller simuler les 3 ou 4 coups suivants, et déterminer ainsi quelle case est la plus avantageuse
* Ajouter des coups de hasard pour que le jeu soit plus équilibré

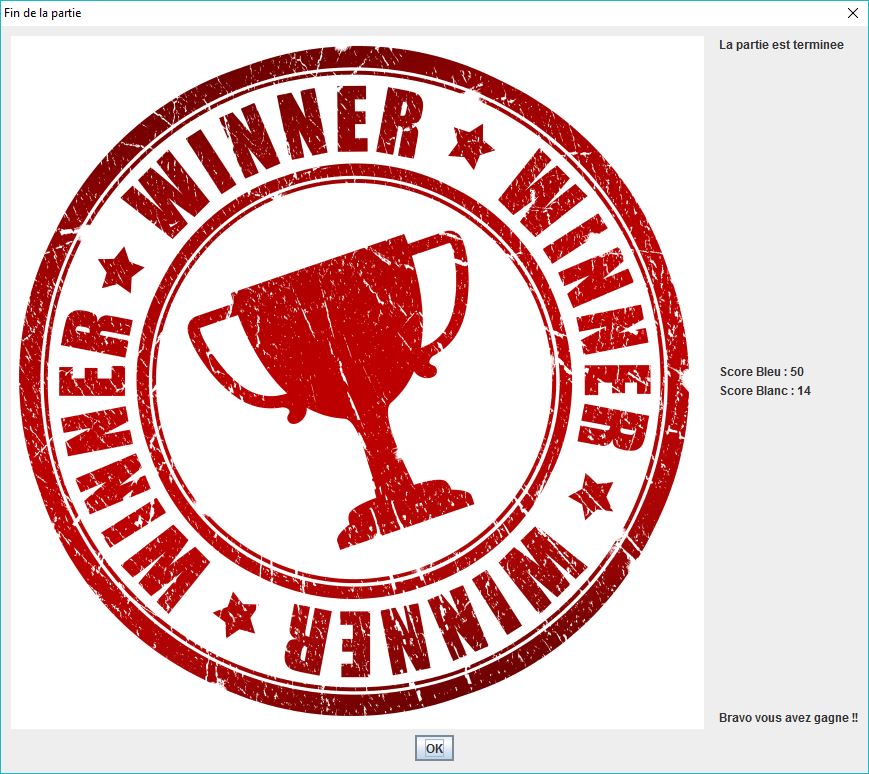
# Éléments implémentés :



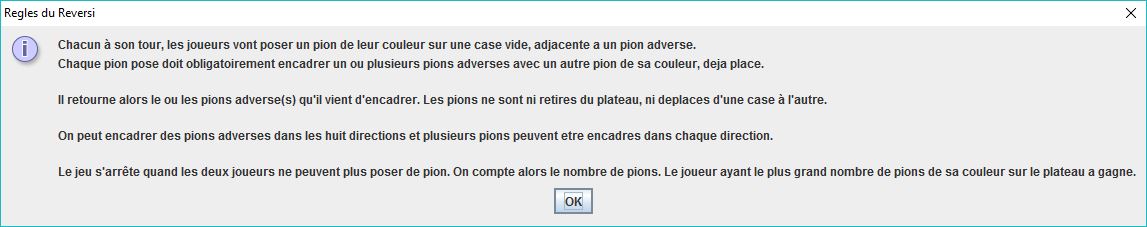
Capture 1 : Recommencer une partie



Capture 2 : Fin de partie perdue



Capture 3 : Fin de partie gagnée



Capture 4 : Règles du jeu

# Conclusion :

Ce jeu nous a permis à la fois d’améliorer nos connaissances du langage JAVA, notamment en découvrant les classes *MouseListener*, *JPanel*, ou encore *JFrame* mais également de faire un premier pas vers les intelligences artificielles. En effet, nous nous sommes intéressées à l’algorithme Minmax même si nous n’avons pas pu le mettre en place dans notre jeu faute de temps.

Notre jeu pourrait encore être amélioré, en rajoutant par exemple l’algorithme Minmax afin de rendre le jeu plus difficile, ajouter la possibilité de choisir la difficulté de l’IA, ou encore de pouvoir à deux joueurs plutôt que contre l’IA.