

## Projet théorie des jeux Développement du jeu Othello



**Enseignante:**  
**Mme. Carla Selmi**

**M1 Informatique - GIL**

**Université de Rouen  
2019 - 2020**

# SOMMAIRE

01

INTRODUCTION

02

SOLUTION TECHNIQUE

03

PRÉSENTATION DE OTHELLO

04

ORGANISATION DU CODE

05

STRATÉGIE

06

FONCTION D'ÉVALUATION

07

DÉMONSTRATION



# INTRODUCTION

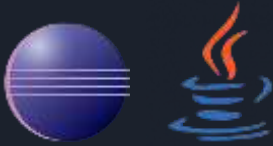
Le projet consiste en le développement du jeu othello en implantant les heuristiques Minimax et AlphaBeta lorsque le joueur s'affronte avec une IA

Le jeux aussi permet à deux IA de s'affronter en utilisant différents algorithmes.

On implante une stratégie et une fonction d'évaluation en suivant les règles du jeu Othello.



# SOLUTION TECHNIQUE PRÉSENTATION DE OTHELLO



Nous avons utilisé JAVA sous l'IDE Eclipse pour le développement du jeu.



Nous avons utilisé l'application zoom pour nos réunions de travail durant la période du confinement.



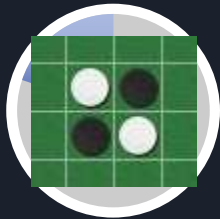
pour la rédaction de nos documents, nous avons utilisé l'outil google drive qui permet la modification des documents par plusieurs personnes en même temps.

# PRÉSENTATION DE OTHELLO



# PRÉSENTATION DE OTHELLO

## Règles du jeu



Position de départ



Noir joue et entoure  
le pion blanc entre  
ces deux pions noir



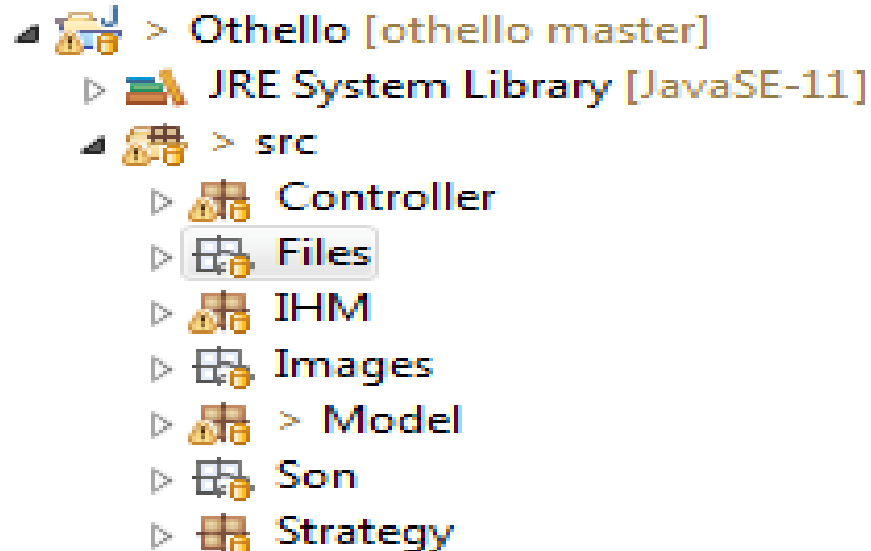
Retourner le pion  
blanc et devient  
noir



Les coups possibles  
pour blanc



# ORGANISATION DU CODE

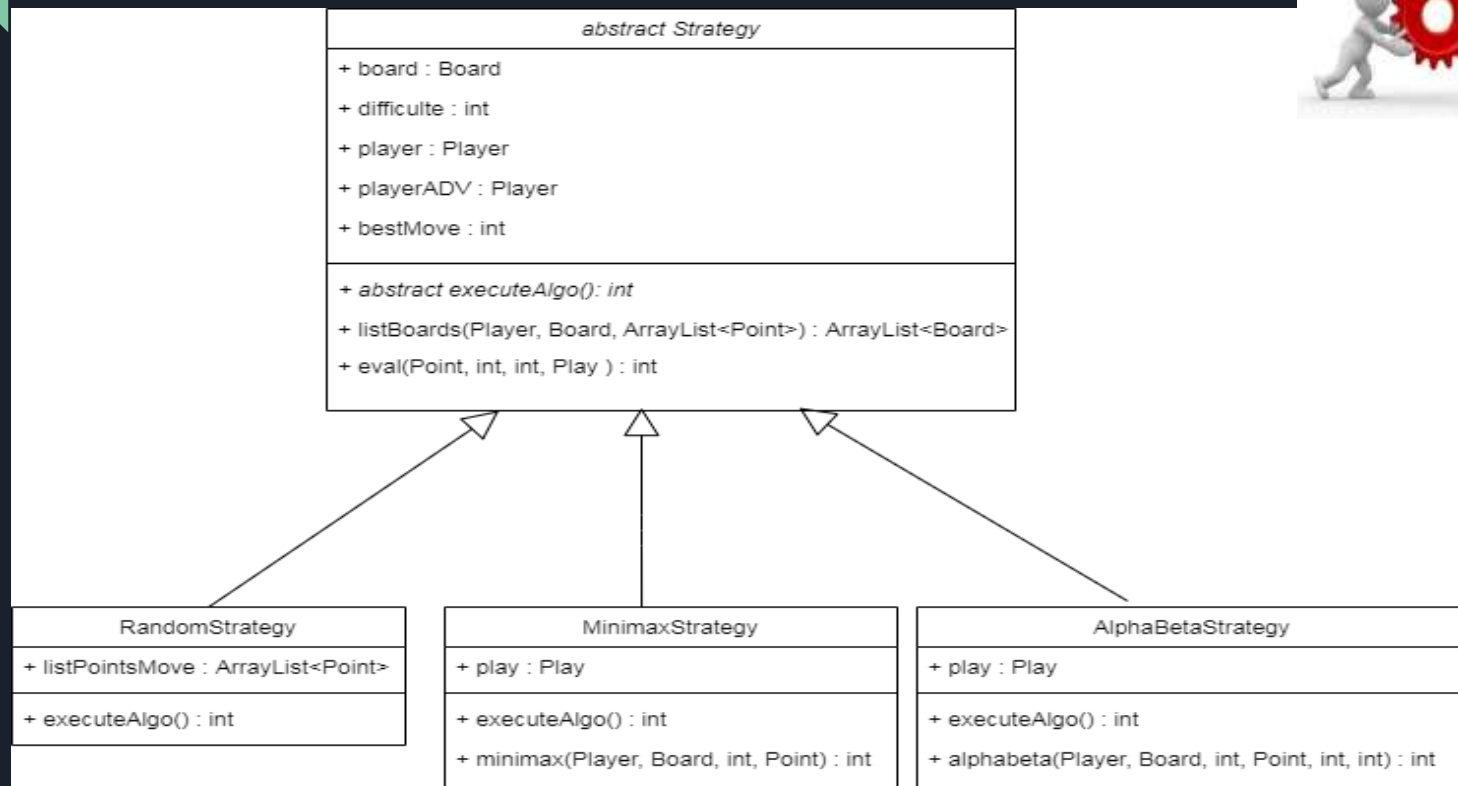


The screenshot shows the project structure of 'Othello [othello master]' in an IDE. The 'src' directory is expanded, showing the following packages and files:

- JRE System Library [JavaSE-11]
- src
  - Controller
  - Files
  - IHM
  - Images
  - Model
  - Son
  - Strategy



# STRATÉGIE







# FONCTION D'ÉVALUATION (I)

```
// methode permet recuperer le profondeur à  
// utilisé selon la difficulté du partie  
public int getDifficulte(String difficult) {  
  
    if(difficult == "EASY") {  
        return 1;  
    }else if(difficult == "MUDIEM"){  
        return 2;  
    }else if(difficult == "HARD"){  
        return 4;  
    }  
    return 0;  
}
```

## FONCTION D'ÉVALUATION (2)

```
// if nbrcoup <= 30 mobilité et position
// if nbrcoup > 30 et <= 55 mobilité et position et material
// if nbrcoup > 55 material
if(play.getInbrCoup() <= 30)
    return evalPowerPoint[powerPoint.x][powerPoint.y] + mobilite;
else if ((play.getInbrCoup() > 30) && (play.getInbrCoup() <= 56))
    return evalPowerPoint[powerPoint.x][powerPoint.y] + mobilite + materiel;
else return materiel;
```

```
{ 500, -150, 30, 10, 10, 30, -150, 500},
{ -150, -250, 0, 0, 0, 0, -250, -150},
{ 30, 0, 1, 2, 2, 1, 0, 30},
{ 10, 0, 2, 16, 16, 2, 0, 10},
{ 10, 0, 2, 16, 16, 2, 0, 10},
{ 30, 0, 1, 2, 2, 1, 0, 30},
{ -150, -250, 0, 0, 0, 1, -250, -150},
{ 500, -150, 30, 10, 10, 30, -150, 500},
```



# DÉMONSTRATION

**Une petite démonstration du résultat du projet.**

**Lien :** <https://www.youtube.com/watch?v=scgBAG1naoE>



# Merci !

Nous avons eu le plaisir de développer  
ce projet et vous remercions pour  
votre écoute !

