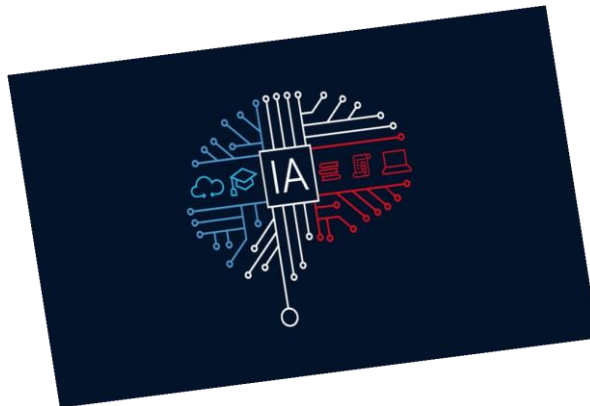
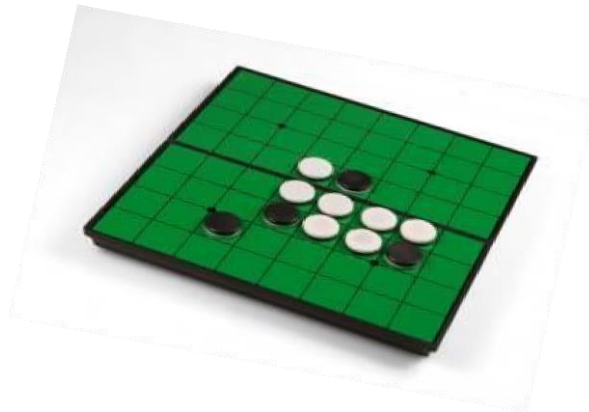


# Rapport de projet individuel

**UE Projet : Jeu d'Othello**



# ***Rapport du projet : Jeu d'Othello***

***AMZAL AMINE***

***Les informations d'identification du document :***

<b><i>Référence du document :</i></b>	
<b><i>Version du document :</i></b>	<b><i>Initiale</i></b>
<b><i>Date du document :</i></b>	<b><i>04/06/2021</i></b>
<b><i>Auteurs et numéro d'étudiant :</i></b>	
<b><i>AMZAL AMINE</i></b>	
<b><i>12011737</i></b>	

**Introduction**

**Règles de jeu:**

**Répartition du travail :**

**Description de ma contributions :**

**Mon expérience de coordinateur :**

**Difficultés principales rencontrées et les solutions trouvées :**

**Réflexion libre:**

**Introduction :**

Le jeu d'Othello est un jeu de société qui oppose deux joueurs : les noirs et les blancs.

Il a été créé en 1971 et est une marque déposée de Lansay. Une variante, le Reversi, est quand à elle libre de droit.

Le but de ce projet est d'appliquer nos connaissances dans le langage JAVA et en algorithmique afin de réaliser non seulement le jeu d'Othello à deux joueurs humains, mais aussi avec un joueur ordinateur ayant plusieurs niveaux de difficultés.

## **Règles de jeu:**

- Othello est un jeu simple: il y a les noirs et il y a les blancs. Les noirs commencent.
- Le but du jeu est de finir la partie avec plus de pions que son adversaire.
- La configuration de départ comprend 4 pions, deux noirs et deux blancs.
- On ne peut poser un pion que contre un pion de la couleur opposée et à condition de pouvoir "encadrer" un ou des pions adverses entre 2 pions de sa propre couleur, que ce soit sur une ligne horizontale, verticale ou diagonale.
- Les pions encadrés sont alors automatiquement retournés en pions de votre couleur.

- Si vous ne pouvez pas jouer, votre tour sera automatiquement passé.
- La partie est finie lorsque le plateau est rempli ou qu'aucun joueur ne peut jouer.
- Le gagnant est celui qui possède le plus de pion à la fin.

## **Répartition du travail :**

Pour commencer on a établi un plan de travail qui consiste en tous ce que chacun de nous va devoir faire et les fonction/classes à implémenter

Après on a divisé le travail comme suit

- Nadine et Kamilia ont implémenté la stratégie et la méthode qui permet à l'ordinateur de jouer
- La fonction legal() on l'a faite a 3 anis , ghiles et moi ,anis a commencé par trouver la méthode pour juger si une case est légale ou pas et ghiles et moi on implémenté le retournement des pions
- Anis a implémenté l'algorithme du jeux et ghiles et moi on est juste intervenu au moment où on avait besoin d'informations graphique
- Ghiles lui a aussi fait une partie qui consiste à dessiner les lignes, les colonnes et les pions
- Moi j'ai aussi fait une partie consiste à afficher les cases qui sont légales avec des petits points noirs et le calcul du score courant et final

## **Description de ma contributions :**

- Dans la partie de la fonction `legal()` que j'ai faite entièrement en coordination avec ghiles on ajouter un paramètre `inverser` qui nous permet de savoir s'il fallait renverser les pions ou pas

Si `inverser` est à `true` on part de la case courante dans le sens inverse de la direction jusqu'à atteindre la case original tout en inversant les pions de l'adversaire

- Après que ghiles est fini de dessiner les lignes, colonnes et pions j'ai me suis occupé des points noir qui désignent les cases légales

En me servant des méthodes `setColor()` et `fillOval()` de Swing j'ai dessiné des points ovales de taille 6x6 et je les ai coloré en la couleur des pions a qui était le tour

Après j'ai vérifié si la partie était pas encore terminée cela en parcourant le plateau du jeu et en vérifiant s'il restait encore des emplacements possible pour un jouer

Si c'est le cas je vérifie juste le score courant et affiche en rouge la chaîne de caractère indiquant le joueur qui a l'avantage actuellement avec la fonction `drawString()`

Sinon la partie est terminée et on affiche le vainqueur en rouge aussi et à l'aide de la même fonction

## **Mon expérience de coordinateur :**

Durant cette période on a tous appris à travailler en groupe et à entendre ce que chacun d'entre nous avait à dire que ce soit une idée d'implémentation de la structure déjà convenu ou une éventuelle modification, ce qui nous a permis d'avoir un maximum d'idées de la manière dont chacun a pu penser, je trouve que ceci nous a permis de réussir à très bien nous coordonner et à s'entraider à chaque fois que l'un d'entre nous avait rencontré des difficultés dans sa partie

Cette expérience m'a appris à travailler en équipe car bien que j'ai eu une partie que je devais faire

moi-même, j'ai toujours gardé un œil sur le travail et les modifications effectuées par le reste du groupe pour adapter mon code, notamment dans le programme principale et toutes les parties où ils ont eu besoin d'information graphique, et ceci est valable pour chacun des membres du groupe car nous étions tous à jour durant la conception du projet.

## **Difficultés principales rencontrées et les solutions trouvées :**

- Au début avec ghiles et anis on avait commencé par des affichages sur console et avec des coordonnées numérique saisi à chaque fois à la main, mais après on s'est rendu compte que mieux vaut passer directement au graphique, ce qui nous a amené à utiliser les bibliothèques de swing
- C'était une nouvelle pour nous car on l'avait pas faite en cour, on a donc était obligé de se formé chaqu'un de son côté avec des tutos sur Youtube ou openclassroom
- Personnellement la difficulté que j'ai eu était dans la manière de voir le plateau du jeu , car au début j'essayais de cibler les cases à modifier et les mettre à jour individuellement ce qui m'a perturbé car je devais écrire beaucoup pour avoir un tout petit changement , mais après avoir bien lu une documentation de swing (la fonction repaint()) j'ai eu l'idée de mettre à jour le plateau tout entier à chaque fois qu'un emplacement est modifié ce qui m'a permis de traiter tous les cas de modifications dans un seul block.



Et bien évidemment j'ai un peu eu du mal avec ghiles pour visualiser la manière dont on doit procéder pour inverser les pions adverses dans la fonction legal() jusqu'à ce qu'on finisse par penser à ajouter un paramètre qui nous indique si c'est faisable , ainsi qu'un parcours inverse en partant de la case vide légale jusqu'à la case de départ .

- Enfin pour le problème de partage du code qu'on a rencontré car des membres ont pas l'habitude d'utiliser github a pu être résolu en s'organisant avec le compte de l'un des membres qui l'utilisent déjà à chaque fois qu'il y avait du nouveau.

## **Réflexion libre:**

- Durant toute la période de réalisation du projet on eut à travailler en groupe ce qui m'a permis d'apprendre à gérer le fait d'avoir plusieurs idées et point de vu, et que je trouve très constructif pour nous autant que futur membre d'une équipe qu'on ne connais pas forcément , donc on appris à ce connaitre et accepter les proposition de chacun et envisager des changement a tous moment

- En ce qui est du déroulement du travail si je devais commencer maintenant je demanderais a chaqu'un d'entre nous d'ouvrir un compte github et de lui expliquer le fonctionnement de base
- Enfin et par rapport au code je me serais pas autant attardé sur la modification individuel des cases du plateau et le côté esthétique, et j'aurais consacré plus de temps pour implémenté une IA qui monte en puissance et en intelligence au fur et à mesure des parties .