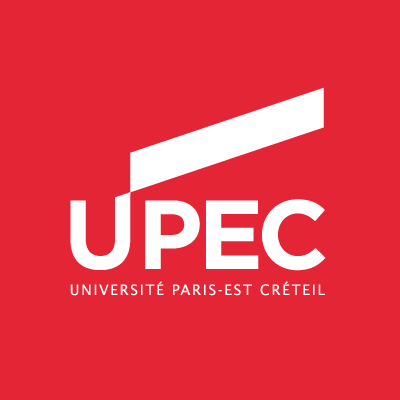
**UNIVERSITE PARIS-EST CRETEIL**

**UFR de Sciences et technologies**

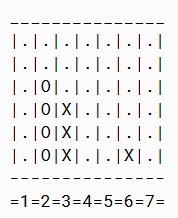
**VAL DE MARNE**

**Compte Rendu Du Projet**

Projet programmation L2

2020/2021

Lamia Benamara



**Remerciement :**

* Notre équipe remercie Madame Lamia Benamara ainsi que les professeurs de l’université de Créteil, qui ont contribué à notre formation avec les supports adéquats dans cette crise inédite.

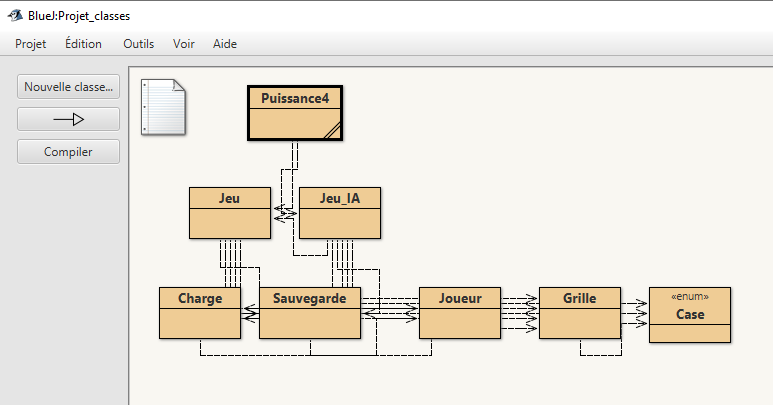
**Membres du Projet :**

|  |  |
| --- | --- |
| Numéro d’étudiant | NOM/Prénom |
| U21993992 | SALAH Hamza |
|  | KOCEILA Kemiche |

**But du projet :**

* L’objectif principal de ce projet est de développer nos compétences en programmation et de mettre en pratique les différentes notions théoriques acquises en programmation orienté objet ainsi qu’en algorithmique et structures de données.

Le projet consiste à implémenter différentes versions du jeu de puissance 4 : la version basique permettant d’affronter deux joueurs humains et la version permettant à un joueur humain d’affronter un ordinateur (IA). La qualité de l’IA est déterminée en fonction de la stratégie (algorithme) adoptée. Le cahier des charges est rédigé par niveau de difficulté. Vous devez suivre l'ordre des questions afin de progresser efficacement.



**Règles du jeu :**

* Le but du jeu est d'aligner une suite de 4 pions de même couleur sur une grille comptant 6 rangées et 7 colonnes. Chaque joueur dispose de 21 pions d'une couleur. Tour à tour les deux joueurs placent un pion dans la colonne de leur choix, le pion coulisse alors jusqu'à la position la plus basse possible dans la colonne à la suite de quoi c'est à l'adversaire de jouer. Le vainqueur est le joueur qui réalise le premier un alignement (horizontal, vertical ou diagonal) consécutif d'au moins quatre pions de sa couleur. Si, alors que toutes les cases de la grille de jeu sont remplies, aucun des deux joueurs n'a réalisé un tel alignement, la partie est déclarée nulle.

Analyse du problème, Cahier des charges :

1. **PARTIE1** : Créer le jeu de base (humain vs humain).
2. Choisir la hauteur et la taille de la grille.
3. Sauvegarder et charger une partie en cours dans un fichier texte.
4. **PARTIE2**  **A**: IA naïve,
5. **PARTIE2**  **B**: IA basée sur Minmax et Alpha-Beta.

**Consignes importantes :**

* Ce projet doit être réalisé en binôme.
* Langage de programmation : JAVA.
* Visionnage sous GIT.

Répartitions des taches :

* Les répartitions des taches ont était égale entre les deux partie du binôme et ont était respecter par les deux partie : **<en plus de l’entre aide mutuelle dans les taches>**

|  |  |
| --- | --- |
| Etudiant : | Taches effectué : |
| SALAH  Hamza | Création des classes < Jeu\_IA\_Simple, Grille, Case et Joueur, Aide à l’implémentation Minmax, Création et implémentation Alphabeta>  Création, désigne et rédaction du rapport de Début – x |
| KOCEILA Kemiche | Création des classes < Création de Jeu, Jeu\_IA\_MinMax, Création et implémentation Alphabeta >  Aide a la rédaction du rapport de page x – fin |

**Source du Projet :**

Puissance 4 : <https://fr.wikipedia.org/wiki/Puissance_4>

Arbre du jeu : https://openclassrooms.com/forum/sujet/puissance-4-ia

Minmax :

* <https://fr.wikipedia.org/wiki/Algorithme_minimax>.
* <https://www.youtube.com/watch?v=da1uzaj549A&t=180s&ab_channel=SylvainSaurel>
* <https://forums.futura-sciences.com/programmation-langages-algorithmique/442444-aide-intelligence-artificielle-jeu-puissance4-matlab.html>

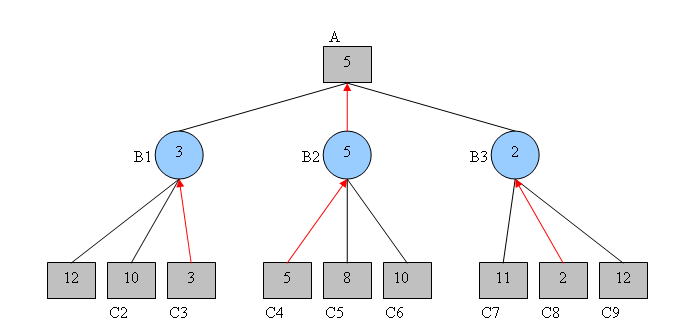


Illustration Minmax

Alpha Beta :

* <https://fr.wikipedia.org/wiki/%C3%89lagage_alpha-b%C3%AAta>
* https://www.youtube.com/watch?v=j4bFdwkAll0&ab\_channel=HugoLarochelle

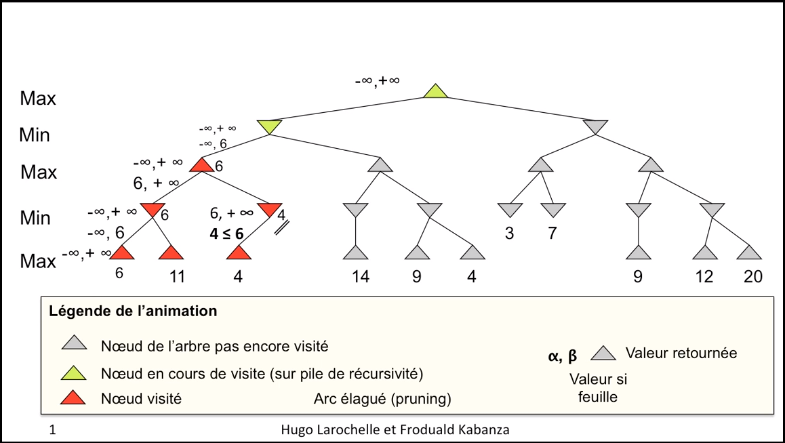
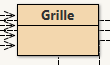


Illustration Alpha Beta

Problèmes rencontrées :

1. Refus de push/pull sur Gitlab
2. Compréhension de Minmax et d’alpha beta
3. Bug contre IA-MinMax choix de colonne a l’infinie quand elle est pleine <Résolue>

Les Classes du jeu :



Cette classe représente la grille. Elle contient toutes les méthodes nécessaires.

Au jeu.

Cette classe représente IA de notre jeu.

: Cette classe représente une case de la grille. Soit elle est Vide ou pleine par (X ou O).



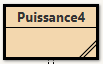
Cette classe représente la partie « J v J » de notre jeu.

Cette classe représente le joueur. Elle contient le nom du joueur et son type de pion.

De pion.

Elle contient le nom du joueur et son type

De pion.



Cette classe le lancement de notre jeu.



Cette classe représente le chargement d’une partie.

D’une partie sauvegarder du jeu.

Cette classe représente la sauvegarde d’une partie.





Cette classe représente l’algorithme d’élagage Alpha beta.

D’une partie sauvegarder du jeu.

Cette classe représente l’algorithme d’élagage Alpha beta.

D’une partie sauvegarder du jeu.

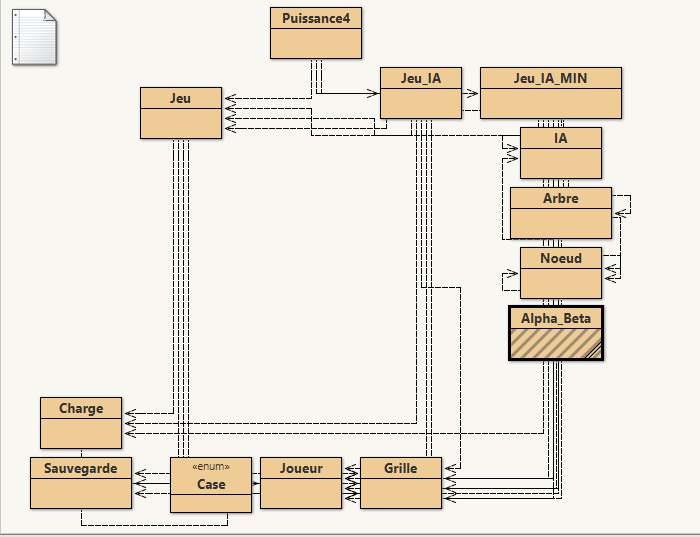
Cette classe représente l’algorithme MinMax et son code de lancement.

D’une partie sauvegarder du jeu.

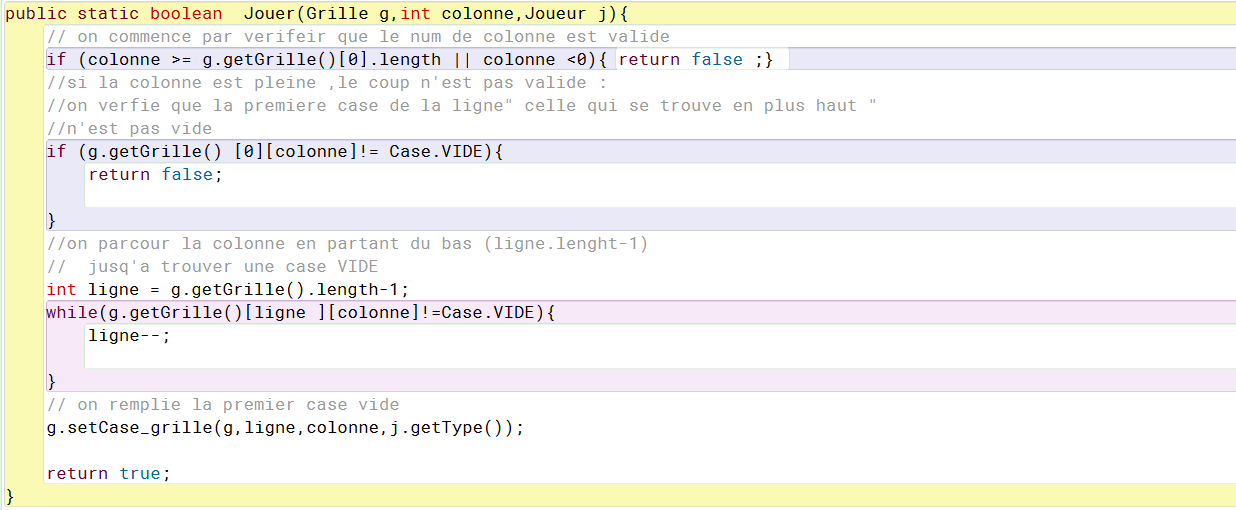
Cette classe représente l’algorithme d’élagage Alpha beta.

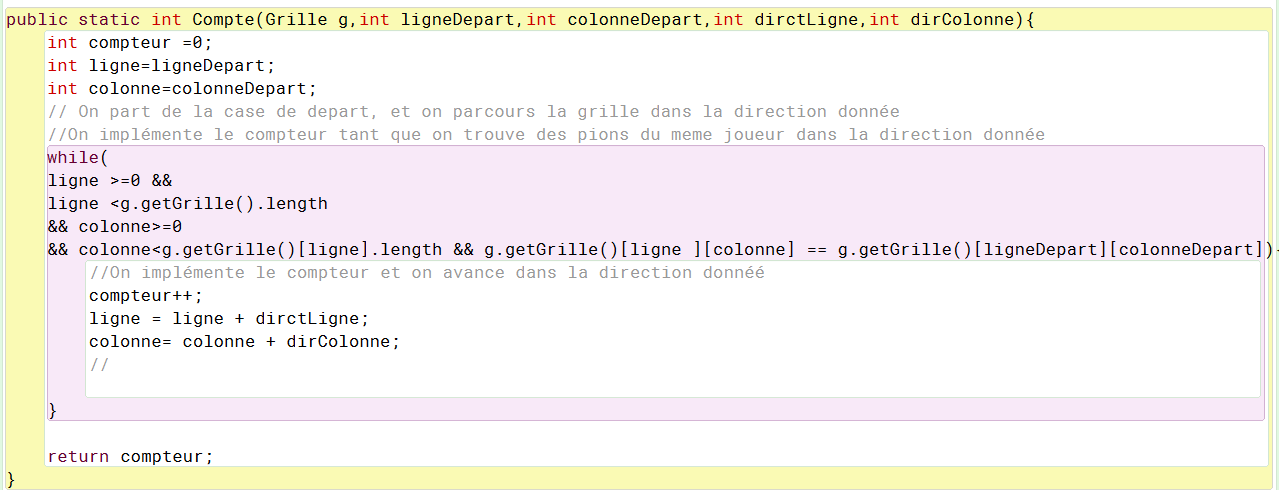
D’une partie sauvegarder du jeu.

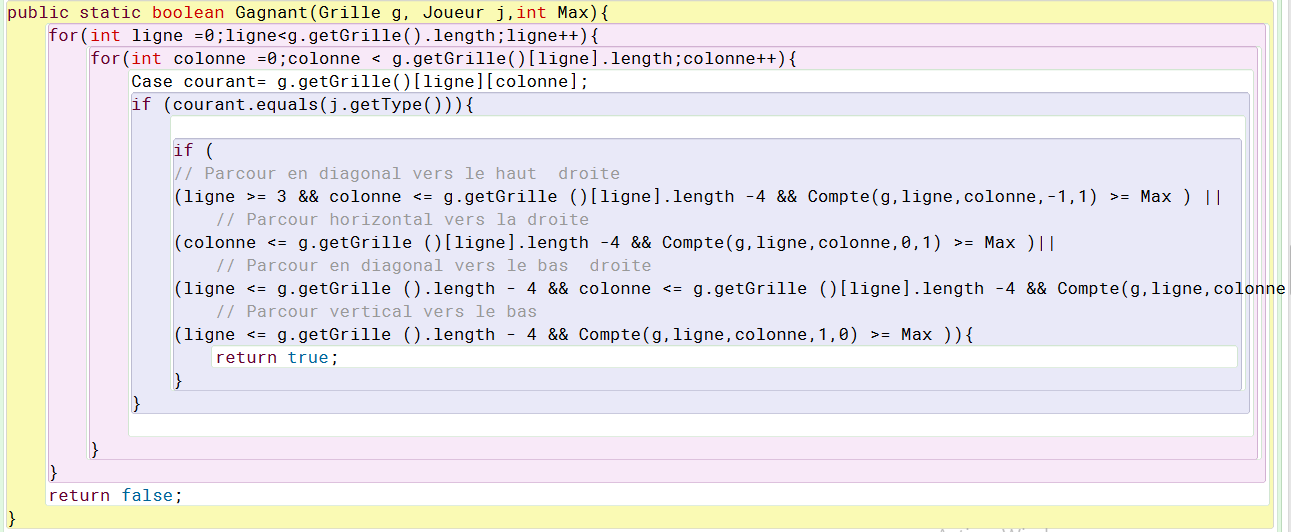
Diagramme Final:



Code pertinent :

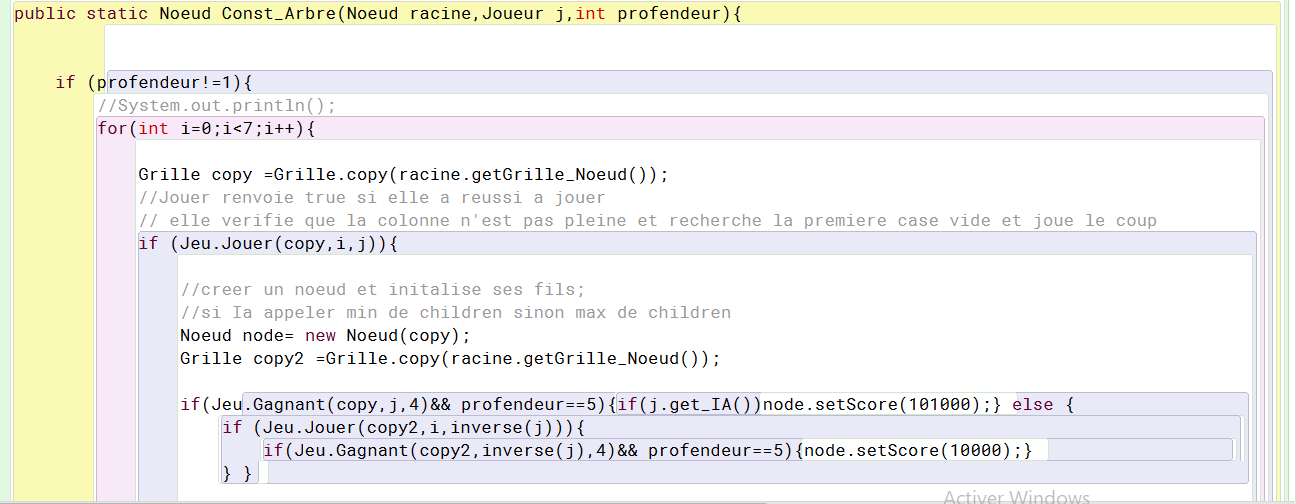
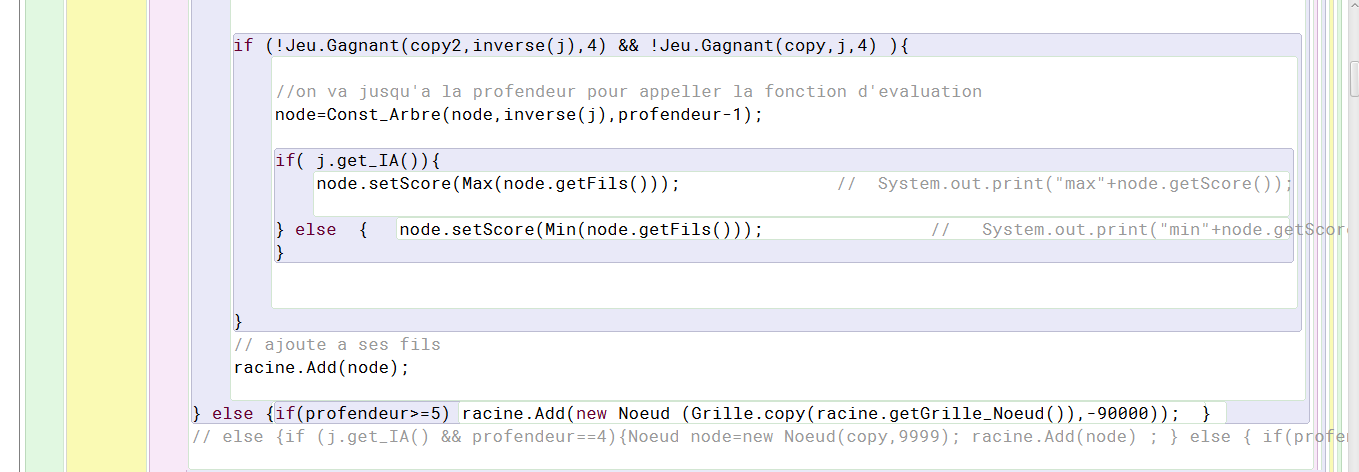
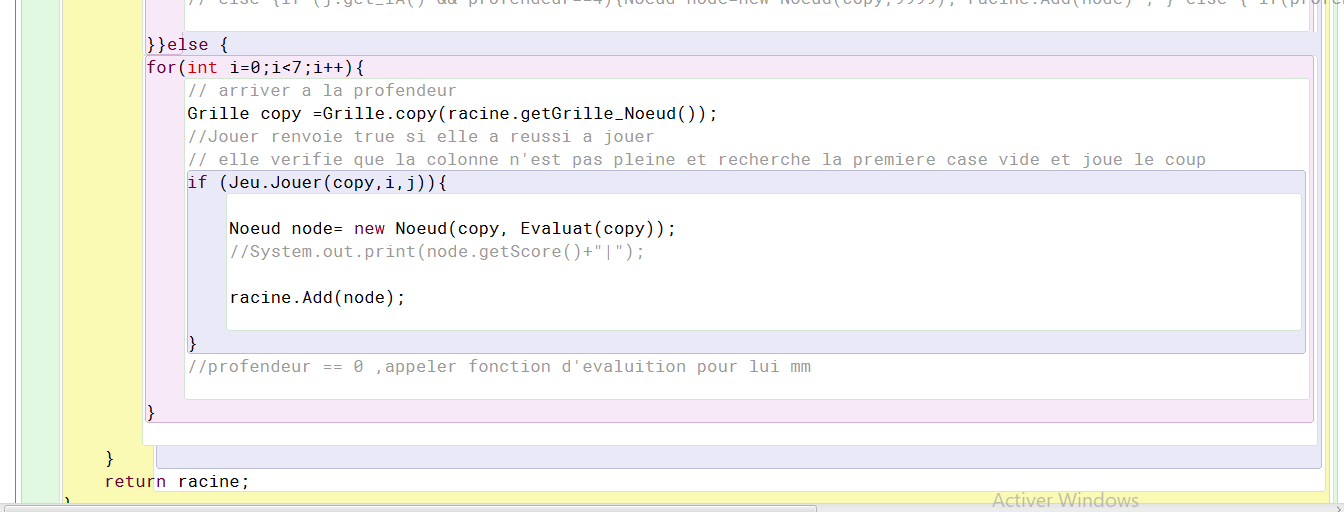
**Description de a méthode jouer :**

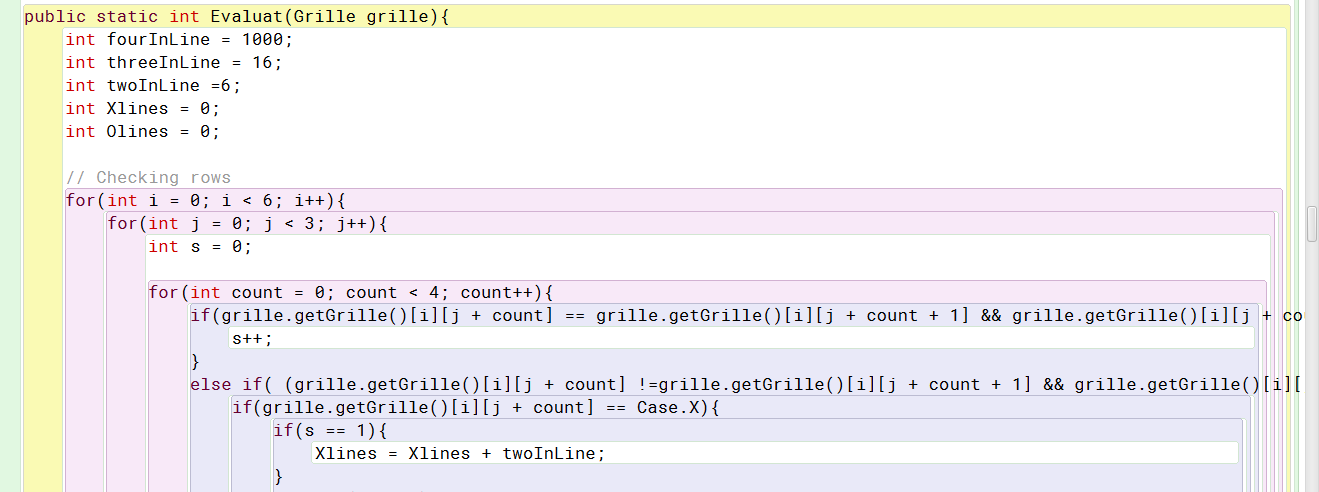
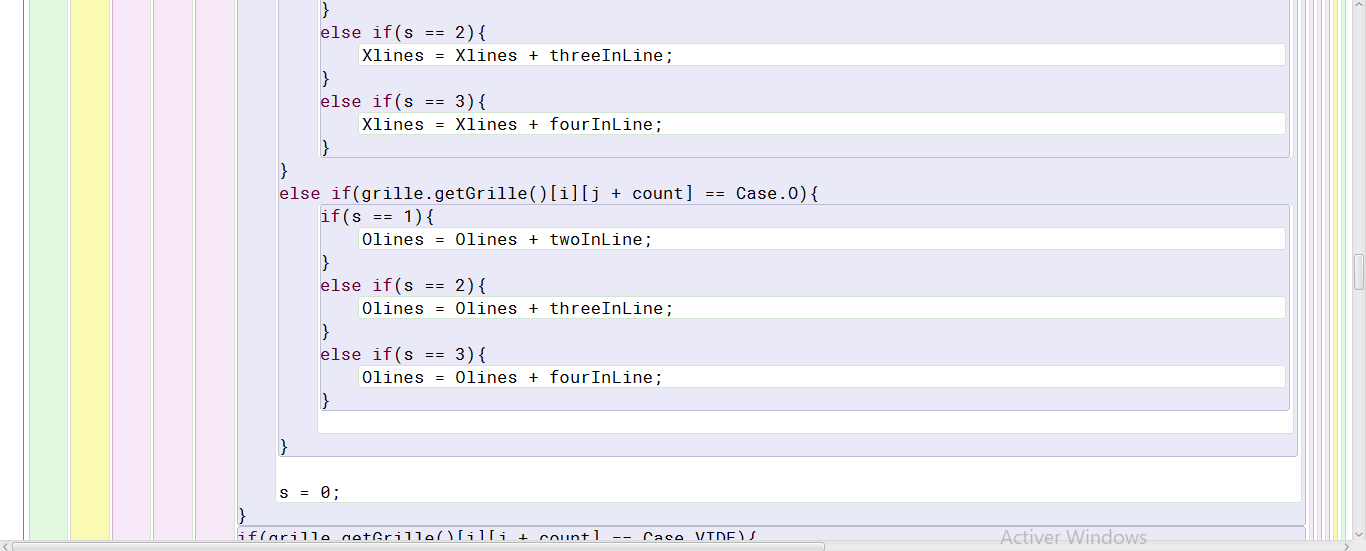
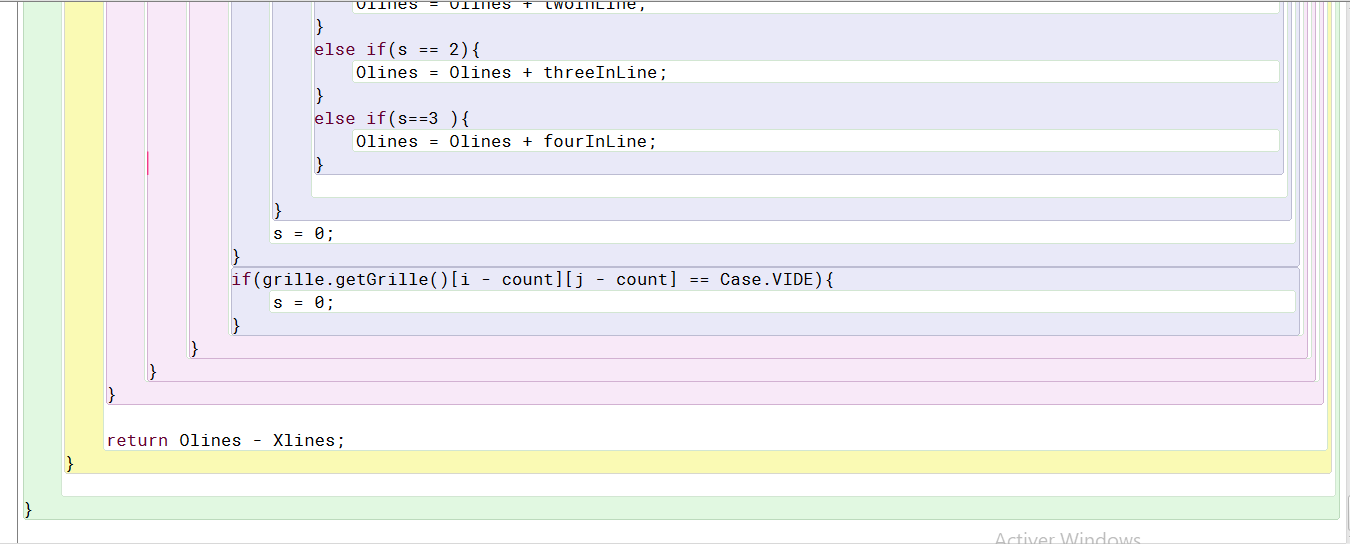
**Description de la méthode compte :**

**Description de la méthode Gagnant :**

**Description de la méthode Choix Colonne**

****

**Description de la méthode Gagnant**

**Description de la méthode Gagnant**