**Puissance 4**

**Introduction**

Encore appelé « 4 alignés », puissance 4 est un jeu de stratégie très connu qui se joue à deux. Le but du jeu est d'aligner le premier 4 pions de sa propre couleur. Les 4 pions peuvent être alignés horizontalement, verticalement ou encore en diagonale. C’est dans cette optique qu’il nous a été demandé dans le cadre de notre projet java de développer une puissance 4.

**Fonctionnement**

Chaque joueur dispose de 21 pions d'une couleur (par convention, en général jaune ou rouge). 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 ladite 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.

**Moteur du jeu**

Il s’agit ici de la partie centrale et la plus importante du jeu. En effet c’est à ce niveau que toute la structure jeu est mise en place c’est-à-dire, la construction du plateau de jeu ainsi que les lignes et case de celui-ci.

**Classe Case** : Il s’agit de la classe permettant de stocker l’état d’une case du plateau de jeu. La case peut se trouver sous 3 états, l’état 0 si la case est vide et l’état 1 ou 2 si lase est occupée.

**Classe Ligne** : Il s’agit de la classe permettant de construire une ligne de Case. Cette classe fournit l’ensemble des méthodes assurant la manipulation de ses données dont les plus importantes sont SetLine et Win.

**SetLine** : Il s’agit de la méthode de permettant de mettre à jour les données d’une ligne.

**Win** : Une des méthodes les plus importantes du jeu. Celle-ci permet de terminer la partie en vérifiant s’il y’a un alignement d’au moins 4 pions sur la ligne.

**Classe Plateau** : Cette classe permet de stocker une grille complète de jeu composé d’un tableau de ligne. Cette dernière est la classe autour de laquelle s’articule la majeure partie des actions du jeu.

**FullColumn** : Il s’agit ici de la méthode qui vérifie la disponibilité d’une colonne.

**AddPoint** : Cette méthode permet d’ajouter un pion dans le plateau

CheckVertical, CheckHorizontal, CheckDiagonal et CheckReverseDiagonal : L’ensemble de ces méthodes constituent celles qui composent les ligne verticale, horizontale et les diagonales afin de vérifier s’il un y’a au moins un alignement de 4 pions.

**NoCheck** : Cette méthode vérifie si le plateau est plein et qu’il n’y a pas de gagnant.

Classe Player : Cette classe permet de conserver les informations du joueur telles que, le numéro du joueur 1 ou 2 en plus du niveau et de la profondeur du joueur dans le cas d’une intelligence artificielle.

**Classe Game** : Cette classe fournit un ensemble de méthode permettant de créer une partie et d’y jouer en mode console en invitant les deux (2) joueur à tour de rôle de préciser la colonne où il souhaite déposer le pion

**Graphique**

Comme son nom l’indique, cette partie met à disposition du joueur un ensemble d’interface graphique afin d’y jouer avec un visuel plus adapté. Les quelques interfaces fournies sont présentées ci-dessous :

L’interface de choix du mode de jeu : 2 modes de jeux sont disponibles, le mode « Real VS IA » permettant à un joueur physique de jouer avec une intelligence artificielle et le mode « Real VS Real » pour 2 joueurs physiques.

L’interface de sélection du niveau de l’IA

L’interface du plateau de jeu.

IA

Stratégie

Conclusion