**Morpion++**

Règles du jeu :

* l'utilisateur détermine le nombre de joueur, le nombre de lignes et de colonnes du plateau, le nombre de pions alignés nécessaire pour gagner
* le premier joueur est déterminé au hasard, puis on suit le tour. Chaque joueur détermine son pion(symbole).
* au démarrage, le plateau de jeu est rempli de "."
* le joueur choisit la case dans laquelle il veut jouer en précisant la lettre et le numéro dans l'ordre qu'il souhaite
* le jeu s'arrête quand le plateau est plein ou quand l'un des joueur a aligné suffisamment de pions

Méthodes nécessaires

* array creationTableau($lig,$col)

// méthode qui crée un tableau de $lig lignes et $col colonnes et le rempli de "."

* array choixSymbole ($nbJoueur)

// methode qui demande à l'utilisateur le symbole associé à chaque joueur. La méthode renvoi un tableau de symboles, le 1er joueur correspond à la case 0 et ainsi de suite.

Les symboles sont alphanumériques

* int joueurSuivant ($nbJoueur,$joueurEnCours)

// détermine le numéro du joueur suivant, si joueurEnCours est null, il choisit un numéro au hasard, sinon il renvoi le joueur suivant. Si on est à la fin de la boucle, il revient au premier

* void afficherTableau($plateau)

// affiche le plateau de jeu

* array conversionPosition($coordonnees)

// converti les coordonnées A3 en [3,0]. Les lettres représentent les colonnes. L'utilisateur peut saisir 4B pour [4,1]

Le tableau renvoyé contient le n° de ligne dans la case 0 et le n° de colonne dans la case 1

* array selectionPosition($plateau,$symbole)

//demande à l'utilisateur la case dans laquelle il veut jouer, vérifie que cette case appartient bien au tableau, vérifie que la case du plateau est vide pour cette position

* array remplirTableau($plateau, $symbole, $positions)

// renvoi le tableau mis à jour avec le symbole dans la case précisée par les positions

* int compteAlignes($plateau, $positions, $directionX,$directionY,$symbole)

// compte le nombre de symboles alignés successifs dans une direction à partir d'une position (cette position n'est pas comptée)

* int testerGagne($plateau,$alignementPourGagner,$positions, $symbole)

// vérifie si le joueur qui vient de jouer à gagner.

Renvoi 1 si la partie est gagnée, -1 si le tableau est plein, 0 si la partie est toujours en cours

* bool plateauPlein($plateau)

// détermine si le plateau est plein (aucune place disponible). Renvoi vrai si le plateau est plein, faux sinon

* void lancerPartie()

//gère la partie avec les autres fonctions

Variables utilisées

* symboles : tableau contenant les symboles de tous les joueurs
* symbole : symbole du joueur en cours
* $nbLig : nombre de lignes du plateau
* $nbCol : nombre de colonne du plateau
* $nbUser : nombre d'utilisateur
* $alignementPourGagner : nombre de pions alignés nécessaire pour gagner
* $plateau : tableau contenant l'avancement du jeu
* $joueurEnCours : numéro du joueur en cours (les numéros de joueur commence à zéro)
* $positions : tableau contenant la position de la case. La 1ere case contient la ligne, la 2nde la colonne

**Modification pour le Puissance 4**

* void afficherTableau($plateau)

// retirer les numéros de ligne

* array conversionPosition($coordonnees)

// les coordonnées ne contiennent qu'une lettre à transformer en numero de colonne

* array selectionPosition($plateau)

//demande à l'utilisateur la colonne dans laquelle il veut jouer, vérifie que cette case appartient bien au tableau et n'est pas pleine, détermine la ligne a laquelle le pion tombe

* bool plateauPlein($plateau)

// ne vérifier que la ligne 0

* void lancerPartie()

//adapter l'appel aux fonction et le titre

* array TrouverCase($plateau,$numCol)

// à partir du numéro de colonne, la fonction renvoi les coordonnées de la première case disponible ou -1 si la colonne est pleine.