Sujet no.8 : Le puissance 4

On a décidé de programmer un jeu de Puissance 4 dont on rappelle succintement les rèlges : c'est un jeu de société se jouant à deux joueurs l'un contre l'autre. Le support de jeu est constitué d'une matrice rectangulaire de 7 cases (horizontalement) sur 6 (verticalement) disposée verticalement. Les joueurs disposent chacun de jetons de couleurs (jaunes pour un joueur, rouges pour l'autre) qu'ils insèrent successivement sur le dessus du support, de telle sorte qu'ils s'accumulent en bas. Le but du jeu est de former un alignement de 4 jetons de la même couleur, horizontalement, verticalement, ou diagonalement.

On modélisera le support par un tableau de 7×6 cases, les jetons par des rectangles de couleur rouge et jaune. Ce programme est constitué selon le schéma établi, c'est-à-dire :

— Une fonction AfficheGrille()
— Une fonction SaisieColonne()
— Une fonction Termine()

Chacune de ces fonctions seront traitées dans les parties qui suivent.

Le stockage des données

Les données sont stockées dans un tableau double appellé JEU[][], tel que JEU=[ligne6, ligne5, ligne4, ligne3, ligne2, ligne1].

	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
$i=0$	$(0,0)$	$(0,1)$	$(0,2)$	$(0,3)$	$(0,4)$	$(0,5)$	$(0,6)$
$i=1$	$(1,0)$	$(1,1)$	$(1,2)$	$(1,3)$	$(1,4)$	$(1,5)$	$(1,6)$
$i=2$	$(2,0)$	$(2,1)$	$(2,2)$	$(2,3)$	$(2,4)$	$(2,5)$	$(2,6)$
$i=3$	$(3,0)$	$(3,1)$	$(3,2)$	$(3,3)$	$(3,4)$	$(3,5)$	$(3,6)$
$i=4$	$(4,0)$	$(4,1)$	$(4,2)$	$(4,3)$	$(4,4)$	$(4,5)$	$(4,6)$
$i=5$	$(5,0)$	$(5,1)$	$(5,2)$	$(5,3)$	$(5,4)$	$(5,5)$	$(5,6)$

TABLE 1 – Couples (i,j) associés aux positions des cases du support de jeu

Chaque ligne est alors à son tour un tableau comportant 7 valeurs. Lorsque la case est vide, la valeur enregistrée sera 0. Lorsque ce sera un jeton rouge, la case prendra la valeur 1, et 2 pour un jeton jaune. Par exemple, pour accéder à la case située dans la deuxième colonne en partant de la gauche et la troisième ligne en partant et du bas (le couple $(3,1)$), on utilise JEU[3][1].

Cycles

On définit un cycle complet lorsque les deux joueurs ont chacun joué leur tour : un cycle est alors composé de deux demi-cycles. Chaque demi-cycle se décompose comme suit :

— Afficher le plateau : AfficheGrille()
— Demander à l'utilisateur d'entrer une colonne : SaisieColonne()
— Si la colonne est correcte, modifier le tableau JEU, sinon le signaler.
— Controler une éventuelle position gagnante des jetons : Termine()

A la fin de chaque demi-cycle, on opère un changement de joueur au niveau de la variable player (player =1 pour le joueur rouge et player=2 pour le joueur jaune)

AfficherGrille()

La fonction AfficherGrille() est la partie graphique du programme. D'abord conçue pour afficher l'état du tableau JEU à des fins de déboguage (toujours disponible sous le nom AfficherGrilleDebug()), elle a ensuite été refaite pour afficher le support du puissance 4 avec les différentes couleurs de jetons. Elle commence par appeller effaceEcranG(), tracer un fond noir ainsi que les lignes qui déliminteront les cases, puis elle affiche des cercles de couleur, avec le centre aux coordonnées : (diamètre*colonne+un demi-diamètre ; diamètre*ligne+un demi-diamètre)

afficheTexte(texte)

En environnement graphique, pour afficher du texte, on est obligé de réécrire une fonction qui affiche du texte sous la grille. Cette fonction prend en paramètre une chaîne de caratère et l'affiche sous la grille, à 10px du bord gauche de l'écran, après avoir appellé la fonction effaceEcranG() pour nettoyer la zone d'écriture.

effaceEcranG(hauteur)

La fonction effaceEcranG() se charge d'effacer l'écran jusqu'en bas à partir d'une certaine position passée en paramètre, en traçant un rectangle blanc.

SaisieColonne()

La fonction SaisieColonne() se décompose en plusieurs étapes :

— Demander à l'utilisateur d'entrer le numéro de sa colonne
— Vérifier si la colonne choisie est disponible
— Si oui, on procède au changement du tableau JEU[][]
— Si non, on repose la question à l'utilisateur en l'informant de l'erreur

On a distingué deux types d'erreur : lorsque la colonne est pleine, ou lorsque le numéro de la colonne ne correspond pas à une entrée valide (entre 1 et 7).

Termine()

La fonction Termine est composée de trois parties distinctes :

Une première étape qui fait appel à la fonction verifLigne(). Cette fonction se charge de vérifier si il y a au moins quatre jetons de la couleur passée en argument dans la ligne. Dans ce cas, elle renvoie true, sinon elle renvoie false.

La deuxième étape qui fait appel à la fonction verifColonne(). Cette fonction possède le même comportement que verifLigne(), mais pour une colonne.

La dernière étape qui fait appel à la fonction verifDiag() se charge de vérifier si il y a au moins quatre jetons de la couleur passée en argument dans une diagonale. On commence par les diagonales inclinées dans un certain sens, puis on fait celles dans l'autre sens.

Enfin, la fonction Termine() est appellée à chaque demi-cycle. Si une seule de ces trois étapes renvoie true, la fonction Termine() stoppe le programme et affiche le joueur qui a gagné.

verifLigne()

La fonction verifLigne() se charge de renvoyer true si un alignement horizontal de 4 pions de la même couleur à un endroit quelconque est découvert. Sinon, elle renvoie false. Pour cela, on inspecte ligne par ligne. Dans chaque ligne on regarde si les cases 1,2,3,4 sont de la même couleur, puis les cases 2,3,4,5, puis 3,4,5,6 et enfin 4,5,6,7. Si aucune de ces séries n'est vérifiée, on passe à la ligne d'en-dessous.

verifColonne()

La fonction verifColonne() se charge de renvoyer true si un alignement vertical de 4 pions de la même couleur à un endroit quelconque est découvert. Sinon, elle renvoie false. Pour cela, on inspecte colonne par colonne. Dans chaque colonne on regarde si les cases 1,2,3,4 sont de la même couleur, puis les cases 2,3,4,5, en enfin 3,4,5,6. Si aucune de ces séries n'est vérifiée, on passe à la colonne de droite.

verifDiag()

La fonction verifColonne() se charge de renvoyer true si un alignement diagonal de 4 pions de la même couleur à un endroit quelconque est découvert. Sinon, elle renvoie false. Les diagonales orientées du haut-gauche vers le bas-droit seront dabord traitées, puis ensuite, celles allant du bas-gauche vers le haut-droit.