IF121

Contrôle continu n° 1

Nom : Prénom : Groupe :	

Le programme C++ doit être écrit dans un fichier portant votre nom, l'extension ".C" et contenant votre nom en commentaire. Tous les documents sont autorisés.

Sujet : Nous pouvons a present realiser un demineur tres simple. Ce jeu est proche de la bataille navale, puisqu'il consiste a decouvrir des objets caché par l'adversaire (ici on joue contre l'ordinateur).

Le jeu se decompose de la facon suivante : l'ordinateur place un nombre fixe de bombes sur un ecran, sans que l'on sache ou elles sont. Le joueur tente de decouvrir les cases les unes apres les autres, à sa guise. S'il decouvre une bombe il a perdu et meurt.

Pour que le jeu ait un interêt, la machine laisse une chance au joueur en lui indiquant, dans le cas ou la case decouverte ne contient pas de bombes, le nombre de bombes placées dans le voisinage immédiat de la-dite case. (entre 0 et 8). Ce jeu prend alors l'habit d'un jeu de reflexion.

Il vous est demandé de réaliser ce jeu pendant le TP, puis de l'essayer dans le temps qu'il vous restera.

Décomposition du problème Un ecran est compose d'un nombre fixe et identique de lignes et de colonnes, et on le modelise par un tableau d'entiers.

Le nombre de bombes sur l'ecran est aussi une constante du programme.

Après avoir définit les constantes et le type **ecran**, il faut initialiser le jeu et afficher les ecrans.

Initialement chacune des cases d'un ecran visible est inconnue, ce qu'on represente par une case de valeur -1. Par la suite le contenu d'une case contiendra le nombre de bombes dans le voisinage si celui ci est connu.

Ecrire une procédure **initVisible** qui initialise un ecran visible. Ecrire une procédure **afficheVisible** qui affiche l'ecran visible. Cet affichage se fera de facon agreable, en mettant un "." pour une case inconnue, et le nombre de bombes dans le voisinage lorsque celui-ci est connu. Il permettra en outre un reperage de chacune des cases en numerotant les lignes et les colonnes.

Testez vos procédures en ecrivant un programme principal qui initialise un ecran visible et l'affiche.

On s'occupe a present de l'ecran cache. Il est compose de cases remplies avec des 0 ou des 1, un 1 indiquant qu'une bombe est presente. Ecrire une procedure **initCache** qui place le nombre de bombes indiquées au hasard dans la grille. Faites un copier-coller de **afficheVisible** pour proposer un **afficheCache**.

Testez à nouveau vos procédures en ecrivant un programme principal qui initialise un ecranCache et l'affiche.

On va maintenant s'interesser aux differentes phases du jeu. Ecrire une procédure **lireCoup** qui attend qu'un nouveau coup valide soit proposé. Un coup est valide si ses coordonnées pointent bien sur une case de l'ecran visible, et que celle ci n'a pas encore été joué. Quels sont les arguments à passer a la procédure bf lireCoup? Par référence ou par valeur?

Testez votre procédure dans votre programme principal.

Une fois le coup choisi, il faut etre capable de le jouer. Une premiere chose à faire est de connaitre son effet. Pour cela on ecrit une fonction **infoCoup** qui retourne le nombre de bombes dans le voisinage de la case. Cette fonction retourne 99 dans le cas ou la case correspondante contient malheureusement une bombe.

Vous pouvez vous inspirer du code suivant pour écrire cette fonction :

```
int info_coup(int x, int y, ecran c){
  int nb=0;
  if (c[y][x]==1) return 99;

  if (y>=1 && c[y-1][x]==1) nb++;
  if (y<dim-1 && c[y+1][x]==1) nb++;
  if (x>=1 && c[y][x-1]==1) nb++;
  if (x<dim-1 && c[y][x+1]==1) nb++;

  if (x>=1 && y>=1 && c[y-1][x-1]==1) nb++;
  if (x>=1 && y<dim-1 && c[y+1][x-1]==1) nb++;
  if (x<dim-1 && y>=1 && c[y-1][x+1]==1) nb++;
  if (x<dim-1 && y<dim-1 && c[y+1][x+1]==1) nb++;

  return nb;
}
```

Ecrire à present une fonction **joueCoup(x,y,ecranVisible,ecranCache)** qui retourne 1 si le joueur a perdu et 0 sinon. Dans le meme temps, elle modifie l'ecran visible pour prendre en compte le coup joue.

Un schema pour le programme principal est à pr'esent le suivant :

initialiser l'ecran visible, et l'ecran cache.
Afficher l'ecran visible.

Calculer le nombre de coup a jouer (le nombre de cases a decouvrir), et mettre dans une variable l'information que l'on a pas encore perdu.

Tant que le nombre de coup a jouer reste positif, et que l'on a pas encore perdu lire un nouveau coup,le jouer et afficher le resultat.

Lorsque l'on sort de cette boucle, on doit savoir si on a gagne ou perdu.