

Durée: 8h

TP logiciel 7 : Terrain miné (ancêtre du démineur)



Objectifs :

- Utiliser Qt Projet C++ en mode console
- Produire un programme et le coder en langage C++
- Développer des classes, instancier des objets et les manipuler
- Mise en oeuvre des objets, manipuler les attributs et les méthodes pour produire un exécutif
- Mise en oeuvre des instructions traitant les tableaux.

Travail à effectuer :

Coder en langage C++ un programme permettant de traverser un champ de mines.

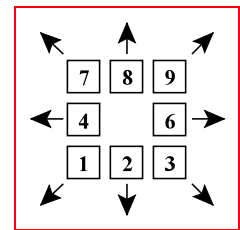
Remarque: Commentez au maximum vos fichiers pour une meilleure compréhension, servez-vous de l'aide de Qt et des sites web appropriés pour produire votre code.

Présentation :

Le but du jeu est de traverser, le plus rapidement possible en un minimum de coup, un terrain miné dans le sens diagonal.



- le terrain est matérialisé par une grille de 20x20.
- les mines sont placées aléatoirement sur la grille.
- le joueur se déplace de la case départ (en bas à gauche) à la case d'arrivée (en haut à droite) en utilisant le pavé numérique tel que :
- un message indique, avant chaque coup joué, le nombre de mines se situant sur le pourtour de la position.



⇒ voir le programme exemple : **jeuMines2022.exe**

Variables :

- **Terrain[20][20]** = tableau d'entiers représentant toutes les zones du champ à traverser. La grille est implémentée sous forme d'un tableau d'entiers tel que si le contenu contient :
 - ⇒ 0 : la case n'est pas minée et les cases autour ne le sont pas.
 - ⇒ 1 : la case n'est pas minée mais vous êtes à côté d'une mine.
 - ⇒ 2, 3, 4, 5, 6 : la case n'est pas minée, mais vous êtes à côté de 2, 3, 4, 5, 6 mines.
 - ⇒ 9 : la case est minée.
- **nbMine** = entier représentant le nombre total de mines
- **posX et poY** = entiers permettant de situer le joueur sur le champs de mines.
- **nbCoups** = entier représentant le nombre de coup joué dans la partie.

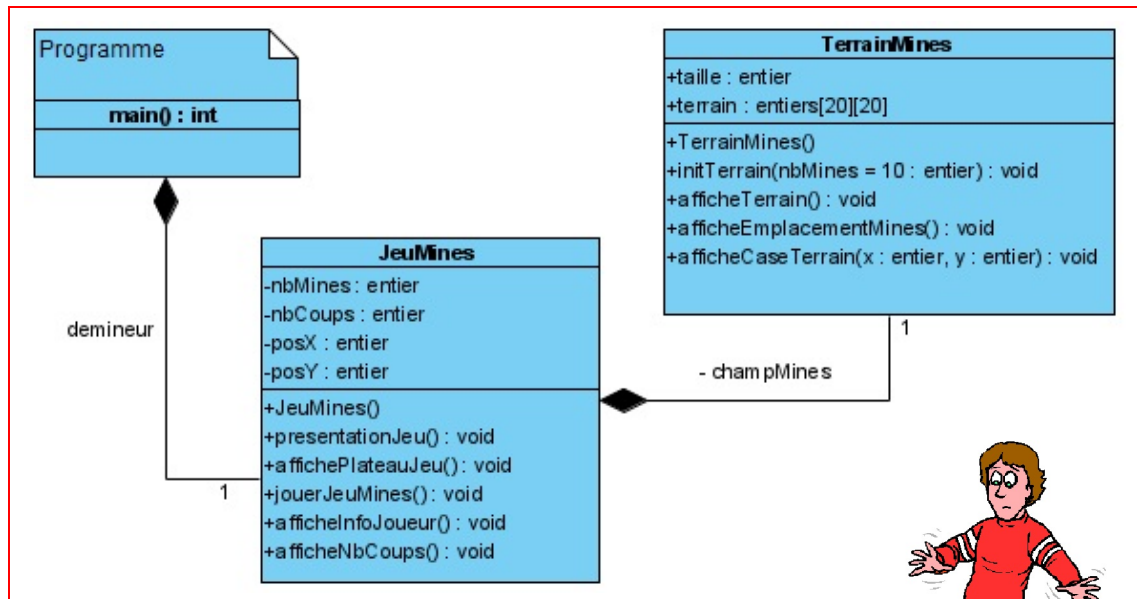
Développement du jeu :

Pour une étude simplifiée, le développement sera effectué en langage orientée-objet, type de programmation où le programme est considéré comme un objet, cet objet est constitué lui-même par un ensemble d'attributs de type simple et de type objet ainsi que de procédures.

Le programme exécute la méthode **jouerJeuMines()** de l'objet **demineur** instancié de la classe **JeuMines**.

Travail demandé :

- 1) Construire entièrement le projet en tenant compte du **diagramme de classe** suivant :



- Inclure les bibliothèques dans le fichier **"terrainmines.h"**
iostream, conio.h, time.h, stdlib.h, consoleJR.h



- 2) Développer la méthode **initTerrain(nbmine = entier)** de la classe **TerrainMines** qui initialise le tableau **terrain** à 0, place les mines de façon aléatoire et détermine la valeur de chaque case en fonction de l'emplacement des mines.
- 3) Coder la méthode **afficheTerrain()** de la classe **TerrainMines** qui affiche chaque case du terrain par le caractère '.' sur l'écran de la console avec l'indication de la case de départ et celle d'arrivée.
- 4) Écrire ensuite la méthode **AfficheEmplacementMines()** de la classe **TerrainMines** qui indique sur l'affichage du terrain, l'emplacement des mines.
- 5) Tester les procédures en les incluant dans la méthode **jouerJeuMines()** tel que :



```

void JeuMines::jouerJeuMines()
{
    champMines.initTerrain(20);
    champsMines.afficheTerrain();
    getch();
    champMines.afficheEmplacementMines();
    getch();
}
    
```

6) Écrire, en langage C++, l'algorithme proposé de la méthode **jouerJeuMines()**

Algorithme de la méthode jouerJeuMines() de la classe JeuMines

Variables de classe : champMines de type TerrainMines
posX, posY, nbMines, nbCoups = entier

Variables : clavier = caractère

début

appel de la méthodes presentationJeu() de la classe JeuMines
appel de la méthode champMines.initTerrain(nbMines)
appel de la méthode affichePlateauJeu() de la classe JeuMines
posX ← 0
posY ← champMines.taille - 1
appel de la méthode afficheInfoJoueur() de la classe JeuMines
appel de la méthode afficheNbCoups() de la classe JeuMines

répéter {

clavier ← fonction getch() // saisie d'une touche clavier
appel de la méthode champMines.afficheCaseTerrain(posX, posY)
nbcoups ← nbcoups + 1

// en fonction de la variable clavier, calculer la nouvelle position posX et posY

si (clavier ='1') alors

si (posX > 0) alors posX ← posX - 1

si (posY < champMines.taille - 1) alors posY ← posY + 1

si (clavier ='2') alors

si (posY < champMines.taille - 1) alors posY ← posY + 1

// ... tester pour clavier '3', '4', '6', '7', '8', '9'

appel de la méthode afficheInfoJoueur() de la classe JeuMines

appel de la méthode afficheNbCoups() de la classe JeuMines

} jusqu'à ((posX = champMines.taille - 1 et posY = 0)

ou champMines.terrain[posX][posY] = 9)

si (champMines.terrain[posX][posY] = 9)

alors {

attendre une touche au clavier

appel de la méthode champMines.afficheEmplacementMines()

Afficher "**** P e r d u ****"

}

sinon

{

appel de la méthode champMines.afficheEmplacementMines()

Afficher "**** G a g n é ****"

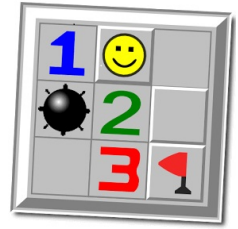
}

attendre une touche au clavier

fin



- 7) Développer la méthode **afficheCaseTerrain(int x, int y)** de la classe **TerrainMines** qui affiche aux coordonnées x et y du terrain, le nombre mines se situant autour du point, valeur présente dans l'attribut tableau **terrain[x][y]**. En cas de valeur nulle, l'affichage se fait par un "." qui change de couleur par rapport au reste de la grille. Ceci permettra de visualiser la progression du joueur sur le terrain.



- 8) Développer la méthode **présentationJeu()** de la classe **JeuMines** qui permet de présenter le jeu et demande au joueur la taille du champs de mines et le nombre de mines à dissimuler sur le terrain. L'attribut **taille** de l'objet **champMines** et l'attribut **nbMines** de la classe **JeuMines** sont mis à jour dans cette méthode à coder.

Tester votre code.



- 9) Écrire le code de la méthode **affichePlateauJeu()** de la classe **JeuMines** qui présente à l'écran le champ de mines. La méthode **afficheTerrain()** de l'objet **champMines** est appelé dans cette méthode, on pourra également afficher les commandes du clavier qui permettent le déplacement du joueur sur le terrain.

- 10) Développer la méthode **afficheInfoJoueur()** de la classe **JeuMines** qui affiche la position du joueur sur le terrain par le symbole 'o' et indique au joueur l'état de la case où il se trouve.

Exemple d'informations :



- Aucune mine détectée -
 - Attention, 1 mine détectée -
 - Attention, 2 mines détectées -
 Etc
 * * * * B o o m * * * *

- 11) Enfin, coder la méthode **afficheNbCoups()** de la classe **JeuMines** qui affiche à l'écran le nombre de coups joué tout au long de la partie.
- 12) Tester l'ensemble du programme et trouver les améliorations pour une utilisation conviviale de ce jeu vis à vis d'un utilisateur novice.
- 13) Améliorer le jeu en proposant une taille de terrain réglable par l'utilisateur (carré de 6 à 25 cases de côté). Prévoir un nombre limite de mines.

