Institut des Techniques d'Ingénieur de l'Industrie

ALSACE

le cnam

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE STRASBOURG

SPECIALITE : INFORMATIQUE

Module : Développement C++

Séance 3 : Programmation Orientée Objet en C++, application via la modélisation d'un jeu vidéo de morpion.

Intervenant : Edouard MANGEL

A l'issue de ce cours théorique, il est temps de passer à la pratique. Vous réaliserez ainsi les programmes suivants dans le logiciel QtCreator (ou dans n'importe quel logiciel avec lequel vous êtes à l'aise, mais en totale autonomie).
Vous écrirez **un** programme composé de plusieurs classes qui seront en rapport avec les différents paragraphes suivants. Vous serez évalués sur la qualité du code, l'application des bonnes pratiques évoquées lors des cours théoriques et l'absence d'erreurs dans vos programmes.
Les questions bonus sont facultatives, et à réaliser uniquement quand toutes les autres questions ont été développées.
Les sources sont à ajouter à la branche master de votre dépôt git au plus tard la veille de la prochaine séance à 23h59.

Objectif : programmer un jeu de morpion complet et évolutif :

Traditionnellement, un morpion se joue à deux joueurs sur une grille de 3 x3. Le plateau de jeu peut être modélisé par un tableau d'entiers à deux dimensions. En C++, une collection en deux dimensions d'entiers sera souvent une collection contenant une autre collection.

I Le jeu de morpion :

En respectant les principes de POO vus dans les slides, vous écrirez un programme permettant de jouer au morpion, à deux sur le même ordinateur, ou contre l'ordinateur.

Chaque tour de jeu se déroule de la façon suivante :

1. Le joueur dont c'est le tour est invité à saisir la case sur laquelle il veut poser un pion,
2. On vérifie que la valeur saisie est bien un entier compris entre 1 et le nombre de lignes, sinon on redemande de saisir.
3. On vérifie que la case sélectionnée est vide et dans les bornes du tableau,
4. Si c'est le cas, on place le pion pour le joueur et on affiche la grille,
5. On vérifie si le joueur a gagné, si oui, la partie est terminée et on félicite le vainqueur.

6. On vérifie si la grille est pleine, si oui on annonce le match nul et on propose de jouer à nouveau.

II Le puissance 4

Vous avez maintenant un morpion fonctionnel, vous allez maintenant développer un puissance 4. Pour cela, vous allez ajouter des classes permettant de jouer au Puissance 4.

On ne joue plus sur une grille de 3 par 3, mais sur une grille dont la largeur sera de 7, et la hauteur de 4, soit 4 lignes et 7 colonnes.

Règle du puissance 4 : chaque joueur joue alternativement (d'abord le premier, le deuxième, ...). Le joueur choisit une colonne libre (non pleine) pour déposer un de ses jetons. Le jeton occupe la case libre la plus basse dans la colonne.

Un joueur gagne la partie lorsque qu'il a réussi à remplir une ligne, une colonne, ou une diagonale de ces jetons. Contrairement au morpion, on choisit uniquement la colonne dans laquelle on joue. On parcourt ensuite la colonne, le pion est placé dans la case la plus basse qui soit vide au moment de jouer.

Contrairement au morpion, le joueur ne gagne pas quand il remplit une ligne, une colonne ou une diagonale, mais quand il aligne 4 pions de sa couleur en ligne en colonne ou en diagonale.

Vous développerez donc des méthodes permettent d'afficher la grille, de placer un pion, de vérifier qu'un joueur a gagné et d'afficher le résultat de la fin de partie.

Vous modifierez votre classe responsable du déroulement du jeu pour qu'elle puisse gérer indifféremment une partie de morpion ou de puissance 4, et diriger une partie de l'un ou de l'autre.

III Bonus :

Que faudrait-il faire pour modifier la classe jeu pour qu'elle puisse prendre en paramètre n'importe quelle Grille, et qu'elle puisse administrer une partie selon les règles de n'importe quel jeu prenant une grille rectangulaire composée de cases carrées ?

Sans la coder, proposez une solution modulaire qui soit facilement adaptable en argumentant sommairement et en agrémentant éventuellement d'un ou plusieurs diagrammes UML qui vous sembleraient adaptés.