**Projet Morpion**

**Classes requises :** -app.cpp

-app.h

-board.cpp

-board.h

-main.cpp

**But du jeu :** Chaque joueur joue chacun son tour, l'un est les croix l'autre est les ronds. Il faut aligner 3 pions de son camp pour gagner en horizontal, en vertical ou en diagonale dans un plateau de taille 3\*3 pions.

**Fonctionnement du jeu :** -Chaque joueur a le choix de placer son pion dans les cases du plateau, qui sont représentées par les nombres allant de 1 à 9.

-Chaque joueur n’a le droit de placer qu’un pion par tour, sur une case libre.

-La partie se termine lorsqu’un des deux joueurs gagnent, ou lorsqu’il toute les cases du plateau sont pleines sans remplir les conditions de victoire.

**Architecture du jeu :** Le jeu de morpion se construit ainsi :

- le main permet de lancer le jeu.

-le board (ou plateau), arène où se déroule le jeu entre les deux joueurs, et l’endroit où l’on place les pions.

-l’app, qui contient la fonctionnalité des joueurs, des conditions de victoires et des vérifications des pions.

**Plateau :**

-L’application génère et affiche un plateau de 9 cases, remplis de nombres allant de 1 à 9 à l’aide de la variable square[i], dont i représente le nombre.

-Les marques apparaissent sur le plateau au même endroit que le nombre indiqué par le joueur. Square[i] = mark signifie que lorsque le joueur choisit un nombre entre 1 à 9, la case dite remplace son chiffre par la mark du joueur correspondant, X ou O.

**Joueur :**

-Le joueur devient X ou O en fonction de la variable player. La variable player varie entre 1 et 2 selon le tour du joueur. Ainsi, lorsque player = 1, mark = X, si player = 2, mark = O.

-Avant d’appliquer le choix du joueur sur le plateau, il faut faire une séquence de vérification. En effet, le jeu vérifie d’abord si la valeur choisie est bien une valeur situé entre 1 et 9. Le joueur entre donc une valeur dans une variable value qui est une string. Cette variable value est ensuite casté dans un int choice. Si la valeur de value est autre chose qu’un chiffre, cela renvoi une erreur, qui est attrapé par la fonction catch, qui renvoi -1 dans la variable z (variable de vérification dont la base est 0), ce qui empêche la validation du choix.

Si la valeur de value est un chiffre, mais qui n’est pas situé entre 1 et 9, cela provoque la réaction du système, qui refuse le précédent choix du joueur, tout en l’invitant à refaire un choix convenable.

Si la valeur de value est bien entre 1 et 9, cela sort le joueur de la boucle (z = 1) et lui permet de passer à l'étape suivante.

Après une première vérification du choix du joueur, on fait une vérification du choix au niveau du plateau.

En effet, le système fera en sorte de vérifier si l’emplacement choisis n’est pas déjà rempli par un des joueurs. Si la case contient un chiffre, cette dernière remplacera le chiffre par la marque du joueur actuel. Si la case contient une marque, le système affichera « Mouvement Invalide », ignorera le choix du joueur tout en empêchant le changement de joueur, et l’invitera à refaire un choix ce qui le renverra dans la première boucle de vérification de l'entrée.

**Condition de victoire :**

Pour gagner à ce jeu, il faut remplir certaines conditions de victoire.

En effet, les deux joueurs devront remplir trois cases alignées en ligne ou diagonale pour gagner la partie.

Ainsi, le système a une fonction checkwin qui vérifie l'état de la partie :

-Si la fonction retourne 1, cela signifie qu’un des deux joueurs a réussi à remplir trois cases alignées et donc à remplir une condition de victoire.

-Si la fonction retourne 0, cela signifie qu’aucun des deux joueurs n’a réussi à remplir les conditions de victoires et que le plateau est plein, menant donc à une égalité.

-Si la fonction retourne -1, cela signifie que le plateau n’est pas rempli, et qu’aucune condition de victoire n’est atteinte. Le jeu continue donc jusqu’au remplissage du plateau ou qu’une des conditions de victoire soient remplies.

Tant que la partie est en cours, le système affichera le plateau, ainsi que le joueur et les choix à prendre.

La fin d’une partie est affichée par la victoire d’un joueur ou l’égalité des joueurs.