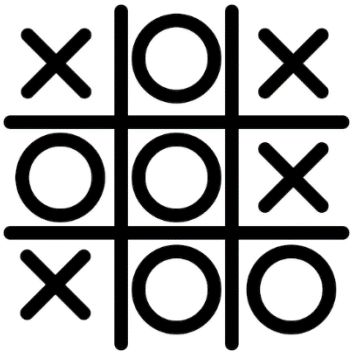
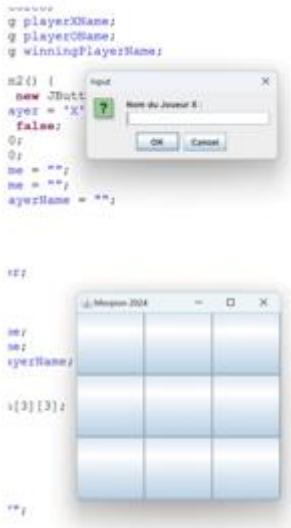
Morpion



Le jeu de morpion, également connu sous le nom de Tic-Tac-Toe, est un jeu de plateau classique pour deux joueurs. L'objectif du jeu est de placer trois symboles identiques (habituellement "X" ou "O") de manière consécutive dans une rangée horizontale, verticale ou diagonale sur un plateau de jeu 3x3. Les joueurs alternent pour placer leur symbole sur une case vide du plateau, et le premier joueur à aligner trois symboles de manière consécutive remporte la partie. Si toutes les cases sont remplies sans qu'aucun joueur n'aligne trois symboles, la partie se termine par un match nul.

Le langage utilisé est le C.




Morpion 2024
X
O
Message
Le joueur Lina gagne !
OK
X
O


Morpion 2024
X
O
Message
Le joueur Lina (Lina) gagne !
Score : Lina - 1, Jul - 0
OK
X
O

L'architecture du code du jeu de morpion en Java peut être décrite comme suit :

1. **Classe principale Morpion** :
    - Cette classe hérite de **JFrame**, ce qui en fait une fenêtre graphique.
    - Elle contient la logique principale du jeu et orchestre les interactions entre les différents composants.
    - Les principales fonctionnalités de cette classe incluent l'initialisation de l'interface utilisateur, la gestion des clics sur les boutons et la vérification de l'état du jeu pour détecter les victoires ou les matchs nuls.
2. **Attributs de la classe Morpion** :
    - **buttons** : Une grille de boutons représentant le plateau de jeu.
    - **currentPlayer** : Le joueur actuel qui joue, initialement défini comme 'X'.
    - **gameWon** : Un indicateur indiquant si la partie a été remportée.
    - **scoreX** et **scoreO** : Les scores des joueurs X et O.
    - **playerXName** et **playerOName** : Les noms des joueurs X et O.
3. **Méthodes de la classe Morpion** :
    - **initializePlayers()** : Initialise les noms des joueurs en demandant à l'utilisateur de les saisir.
    - **initializeUI()** : Initialise l'interface utilisateur en créant et disposant les boutons sur la fenêtre.
    - **checkWin(int row, int col)** : Vérifie s'il y a une victoire ou un match nul en examinant l'état actuel du plateau de jeu.
    - **ButtonClickListener** : Une classe interne qui écoute les clics sur les boutons et exécute une action en réponse à un clic.
    - **updateScore()** : Met à jour le score des joueurs et affiche un message indiquant le joueur gagnant.

- **getPlayerName(char player)** : Retourne le nom du joueur en fonction du symbole 'X' ou 'O'.
- **restartGame()** : Réinitialise le jeu pour commencer une nouvelle partie.

4. **Méthode main()** :
    - Cette méthode crée une instance de la classe **Morpion** et rend la fenêtre du jeu visible.

Les fonctionnalités essentielles du jeu de morpion incluent la gestion des clics sur les boutons, la vérification des conditions de victoire ou de match nul, la mise à jour du score des joueurs et la possibilité de redémarrer le jeu.

Bilan de compétences :

- Gérer le patrimoine informatique
- Travailler en mode projet *
-Mettre à disposition des utilisateurs un service informatique