

## Documentation Img2brick

Cette documentation présente les aspects fonctionnels et techniques de l'application de Morpion développée en Python.

### SOMMAIRE :

<b>1. Analyse Fonctionnelle : .....</b>	<b>2</b>
- <b>1.1. Présentation : .....</b>	<b>2</b>
- <b>1.2. Système de Jeu: .....</b>	<b>2</b>
<b>2. Analyse Technique : .....</b>	<b>3</b>
- <b>2.1. Architecture du code : .....</b>	<b>3</b>
- <b>2.2. Environnement de déploiement: .....</b>	<b>3</b>
- <b>2.3. Structure de données: .....</b>	<b>3</b>

## 1. Documentation Fonctionnelle

### Présentation

L'application est un jeu de Morpion classique se jouant à deux joueurs sur un même terminal. L'objectif est d'aligner trois symboles identiques (horizontalement, verticalement ou en diagonale) sur une grille de 3x3.

### Système de Jeu

- initialement : La partie commence avec une grille vide. Le joueur 1 possède le symbole X et le joueur 2 le symbole O.
- Déroulement d'un tour : Le programme demande successivement au joueur actif de saisir le numéro de la ligne et de la colonne (de 1 à 3) où il souhaite placer son symbole.
- Règles de validation : Un joueur ne peut pas jouer sur une case déjà occupée ou en dehors des limites de la grille. En cas d'erreur, le programme demande une nouvelle saisie.
- Conditions de fin :
  - Victoire : Un joueur aligne trois symboles.
  - Égalité : La grille est complète sans qu'aucun alignement ne soit réalisé.
- Rejouer : À la fin d'une partie, l'utilisateur a la possibilité de relancer une nouvelle partie ou de quitter l'application.

## 2. Documentation Technique

### Architecture du Code

Le code est structuré de manière procédurale dans le fichier main.py et utilise les fonctions suivantes :

- main() : Point d'entrée qui gère la boucle principale du jeu et l'alternance des tours.
- dessine\_grille(tab) : Gère l'affichage visuel de la grille dans la console après chaque action.
- jouer\_tour(tab, joueur) : Gère l'interaction avec l'utilisateur pour la saisie des coordonnées.
- verif\_tour(...) : Fonction de contrôle vérifiant la validité d'un coup (coordonnées et disponibilité de la case).
- condition\_fin\_partie\_... : Série de fonctions spécialisées (diagonales, lignes/colonnes, match nul) pour déterminer l'état final du jeu.

### Environnement et Déploiement

- Langage : Python 3.12.3.
- Environnement Virtuel : Le projet inclut un dossier venv pour isoler les dépendances.
- Scripts d'activation : Plusieurs scripts (Bash, Fish, PowerShell) sont fournis pour activer l'environnement selon le système d'exploitation utilisé.
- Compatibilité OS : Le code utilise os.name pour adapter la commande d'effacement de la console (cls pour Windows, clear pour Unix/Linux).

### Structure des données

La grille est représentée par une liste de listes (matrice) : [[ " ", " ", " "], ...], facilitant le parcours par indices pour les vérifications de victoire.

-