

# Cahier des charges – Projet Morpion en Python (Tkinter)

## I. Introduction

### A. Présentation du projet

Le projet consiste à développer un jeu de morpion (tic-tac-toe) en Python, utilisant la bibliothèque Tkinter pour l'interface graphique.

Le jeu permettra à deux joueurs de s'affronter sur une même machine, en alternant les coups et en affichant les résultats en temps réel.

### B. Objectifs du cahier des charges

Ce document a pour objectif de définir :

- Les fonctionnalités du jeu
  - Les règles de fonctionnement
  - Les objectifs pédagogiques et techniques
  - Les contraintes de développement
  - Les critères de validation
- 

## II. Contexte du projet

### A. Description du contexte

Le projet s'inscrit dans un cadre d'apprentissage de la matière "Qualité de développement".

Il vise à mettre en pratique la rigueur de développement d'un projet en respectant les normes de codage.

### B. Analyse des besoins

- Développer une interface simple, intuitive et agréable.
- Permettre une interaction fluide entre les joueurs.

- Assurer la détection automatique des conditions de victoire ou d'égalité.
  - Faciliter la réinitialisation d'une partie.
- 

## **III. Objectifs du projet**

### **A. Objectifs généraux**

- Développer un jeu complet et fonctionnel.
- Respecter les règles traditionnelles du morpion.
- Respecter les normes de codage et le concept de qualité de développement.

### **B. Objectifs spécifiques**

- Créer une grille de 3x3 cases interactives.
  - Alternner automatiquement le tour entre les deux joueurs ("X" et "O").
  - Déterminer automatiquement le gagnant ou le match nul.
  - Réinitialiser la partie après chaque fin de jeu.
- 

## **IV. Fonctionnalités requises**

### **A. Liste exhaustive des fonctionnalités**

1. Interface graphique avec une grille 3x3.
2. Affichage des symboles "X" et "O" au clic sur une case.
3. Vérification des conditions de victoire ou d'égalité.
4. Affichage d'une boîte de dialogue annonçant le résultat.
5. Réinitialisation automatique de la grille en fin de partie.

## B. Priorisation des fonctionnalités

- **Essentielles** : grille interactive, alternance des joueurs, détection victoire/nul.
- **Secondaires** : ajout d'un compteur de scores, personnalisation des symboles, amélioration du design.

## C. Interactions entre les fonctionnalités

- Chaque clic modifie la grille, ce qui déclenche une vérification des conditions de victoire.
  - La détection d'une fin de partie entraîne soit une victoire, soit un nul, suivi d'une réinitialisation.
- 

# V. Contraintes et limitations

## A. Contraintes de temps

- Le projet doit être développé en un délai assez court (durant les cours).

## B. Contraintes techniques

- Utilisation exclusive du langage **Python 3**
  - Interface graphique réalisée uniquement avec **Tkinter**.
  - Pas de dépendances autres que Tkinter.
  - Exécution locale (pas de mode en ligne multijoueur).
- 

# VI. Tests et validation

## A. Stratégie de test

- Tests unitaires sur la fonction de détection du gagnant.
- Tests fonctionnels sur l'interface (clics, alternance joueurs).

- Tests utilisateurs (jouer plusieurs parties).

## **B. Critères de réussite des tests**

- Les victoires doivent être correctement détectées.
- Les matchs nuls doivent être reconnus.
- La grille doit se réinitialiser sans erreurs.
- L'interface doit rester stable en toutes circonstances.

## **C. Procédure de validation du projet**

- Démonstration du bon fonctionnement après plusieurs test.
  - Vérification que toutes les fonctionnalités essentielles sont implémentées.
- 

## **X. Glossaire**

- **Tkinter** : Bibliothèque standard de Python pour la création d'interfaces graphiques.