



PROJET MORPION

Lewis CRANE

Table des matières

Présentation Générale	2
Besoin Initial	2
Solution Proposée	2
Plan et Maquette du Site	2
Dictionnaire de Données et MCD	2
Dictionnaire de Données :	2
Modèle Conceptuel de Données (MCD)	2
Descriptif Technique de l'Application	3
Technologies et Langages Utilisés	3
Organisation du Code	3
Procédure d'Installation	3
Guide d'Utilisation	3
Descriptif des Tâches et Attributions	3
Planning des Activités (Diagramme de Gantt)	3
Déroulement du Projet et Difficultés Rencontrées	4
Difficultés	4
Solutions	4
Conclusion	4

Présentation Générale

Besoin Initial

Le projet consiste en la création d'un jeu de morpion (tic-tac-toe) en utilisant la bibliothèque Pygame. Le but est de proposer une application interactive permettant à deux joueurs de s'affronter sur un plateau de jeu en affichant leurs coups sur une interface graphique.

Solution Proposée

Pour répondre à ce besoin, nous avons développé une application en Python utilisant Pygame. L'application affiche une grille de jeu, permet aux joueurs de placer leurs symboles ("X" et "O") et détecte les conditions de victoire.

Plan et Maquette du Site

Comme il s'agit d'une application de bureau et non d'un site web, la conception repose sur une interface graphique affichant :

- Une grille de 3x3 pour le jeu
- Les symboles « X » et « O »
- Un système de gestion des tours
- Une vérification automatique des conditions de victoire

Dictionnaire de Données et MCD

Dictionnaire de Données :

Nom de la variable	Type	Description
c11, c12, c13...	str	Cases de la grille
joueur	str	Joueur actif ("X" ou "O")
victoire	int	Indicateur de victoire (0 = en cours, 1 = gagné)
tour	int	Nombre de tours joués

Modèle Conceptuel de Données (MCD)

Un MCD n'est pas strictement nécessaire pour ce projet, car il ne repose pas sur une base de données.

Descriptif Technique de l'Application

Technologies et Langages Utilisés

- Python 3
- Bibliothèque Pygame pour l'affichage graphique

Organisation du Code

- Chargement des images
- Affichage de la grille de jeu
- Gestion des entrées utilisateur
- Vérification des conditions de victoire
- Mise à jour de l'affichage

Procédure d'Installation

1. Installer Python 3
2. Installer Pygame : ``pip install pygame``
3. Exécuter le script Python : ``python morpion.py``

Guide d'Utilisation

- Lancer l'application
- Entrer les coordonnées de jeu en console
- Observer l'affichage des symboles sur la grille
- Le jeu continue jusqu'à ce qu'un joueur gagne ou qu'il y ait match nul

Descriptif des Tâches et Attributions

Le projet étant réalisé en solo, toutes les tâches ont été effectuées par CRANE Lewis.

Planning des Activités (Diagramme de Gantt)

Tâche	Durée
Conception du projet	2 jours
Développement du code	3 jours
Tests et corrections	2 jours
Rédaction du rapport	1 jour

Déroulement du Projet et Difficultés Rencontrées

Difficultés

- Blocage du programme à cause d'une boucle infinie
- Gestion incorrecte des tours de jeu
- Détection de la victoire incorrecte

Solutions

- Implémentation d'une gestion événementielle de Pygame
- Correction des erreurs d'affichage et de logique
- Ajout de vérifications supplémentaires

Conclusion

Ce projet m'a permis de renforcer les compétences en programmation Python et en gestion des interfaces graphiques avec Pygame. Les difficultés rencontrées ont été une bonne occasion d'apprentissage et d'amélioration des compétences en débogage et en gestion des entrées utilisateur.