

TP blanc : Jeu **Puissance 4** en .NET MAUI

Objectifs

- Implémenter un jeu de **Puissance 4**, permettant à **deux joueurs** de s'affronter sur un même appareil.
 - Respecter l'architecture **MVVM** afin de séparer la logique métier de l'interface utilisateur.
 - Écrire des **tests unitaires** pour valider la logique métier (gestion des tours, détection de victoire, etc.).
-

Règles du jeu

Le jeu **Puissance 4** se joue sur une **grille de 7 colonnes et 6 lignes**.

1. Déroulement du jeu :

- Deux joueurs jouent **tour à tour**, en insérant un jeton de leur couleur dans l'une des colonnes de la grille.
- Le jeton tombe jusqu'à la **première case vide** de la colonne choisie.

2. Conditions de victoire :

Un joueur gagne s'il aligne **4 jetons de sa couleur** :

- **Horizontalement** ou **Verticalement** ou **En diagonale**

3. Cas d'égalité :

- La partie est déclarée **nulle** si la grille est **complètement remplie** sans qu'aucun joueur n'ait aligné 4 jetons.
-

Contraintes Techniques

Architecture

L'application doit respecter le **pattern MVVM** :

- **Model** : gestion des données et règles métier (grille, règles de victoire, etc.).
- **ViewModel** : liaison entre la logique métier et l'interface utilisateur.
- **View** : interface graphique en **XAML**.

Tests Unitaires

- La **logique métier** doit être couverte par des **tests unitaires** pour valider :
 - L'insertion des jetons.
 - La détection des alignements gagnants.
 - La gestion de l'égalité.

Interface Utilisateur

- Interface avec une grille dynamique

- Affichage du joueur en cours de jeu
 - Un joueur ne peut pas déposer un jeton dans une colonne pleine
 - Affichage du gagnant du jeu quand la partie est finie
 - Bouton pour recommencer une partie quand elle est finie
-