



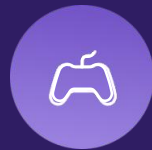
Puissance 4 Multijoueur en en Ligne

Un projet de développement web complet : du classique à l'innovation.

Appuyez sur une touche pour commencer

Présentation du Projet et Technologies

Objectif : Moderniser le jeu classique avec des modes innovants



Le Projet

Recréer un Puissance 4 en ligne, fonctionnel et compétitif.



Backend (Logique Serveur)

Go (Golang) pour la performance et la gestion des connexions simultanées.

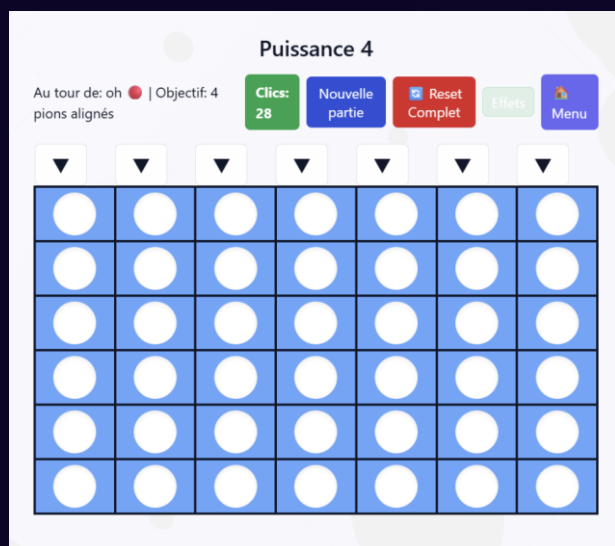


Frontend (Interface Utilisateur)

HTML, CSS, JavaScript pour une expérience utilisateur rapide et responsive.

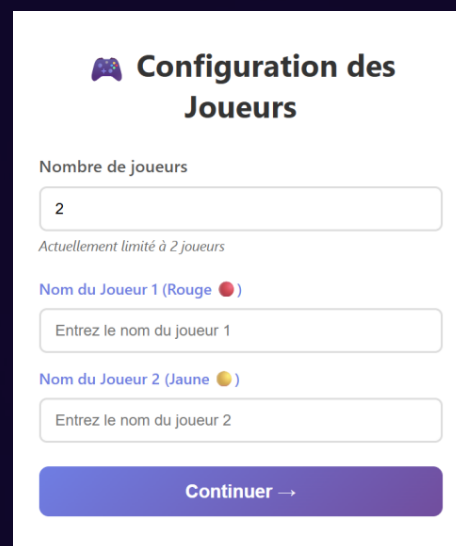
Description du Jeu : La Base Commune

Une reproduction numérique fidèle, pensée pour l'accessibilité.



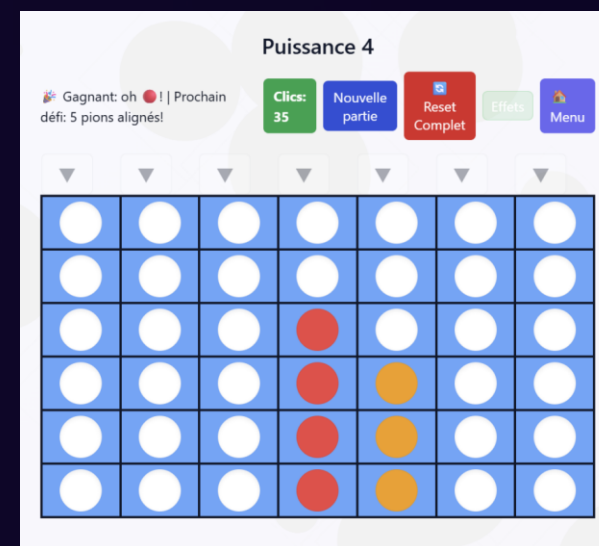
Grille Standard

Grille classique : 7 colonnes et 6 lignes.
La fondation du gameplay.



Interface Intuitive

Design clair, simple et entièrement
responsive sur tout appareil.



Règle de Victoire

L'alignement de 4 pions (vertical,
horizontal ou diagonal) reste la règle.

Mode Classique et L'Innovation des Variantes

Au-delà de la version classique, trois modes pour renouveler l'expérience.



Mode Classique

Version de base, fidèle et essentielle. Point de départ pour le développement.



Mode Exponentiel

Après la victoire d'un joueur la grille s'agrandis de plus en plus le puissance 4 deviens un puissance 5,6,7...

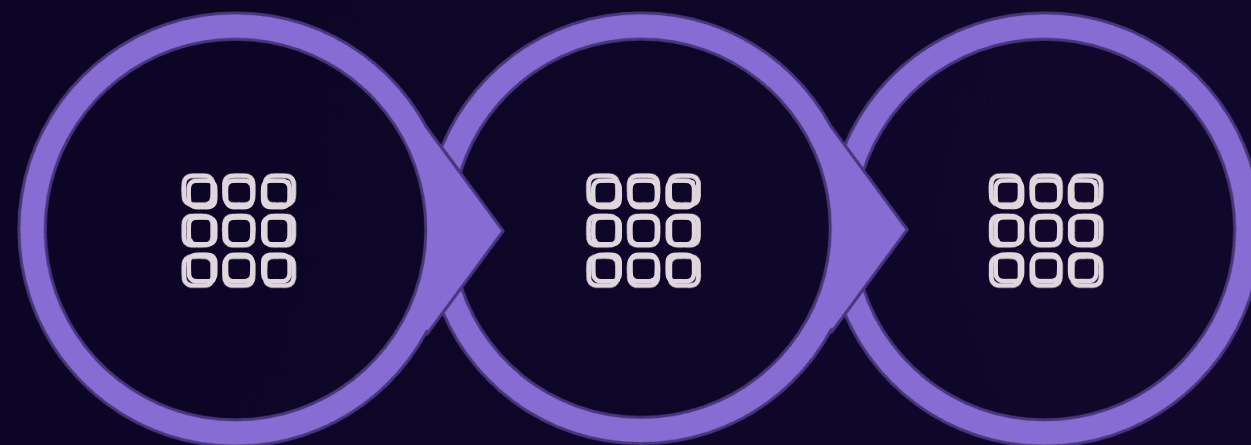


Mode Turbo

Un mode rapide avec des *power-ups*.

Zoom sur le Mode Exponentiel

Ce mode introduit une complexité de jeu et un défi technique.



Step 1: 7x6

Step 2: 8x7

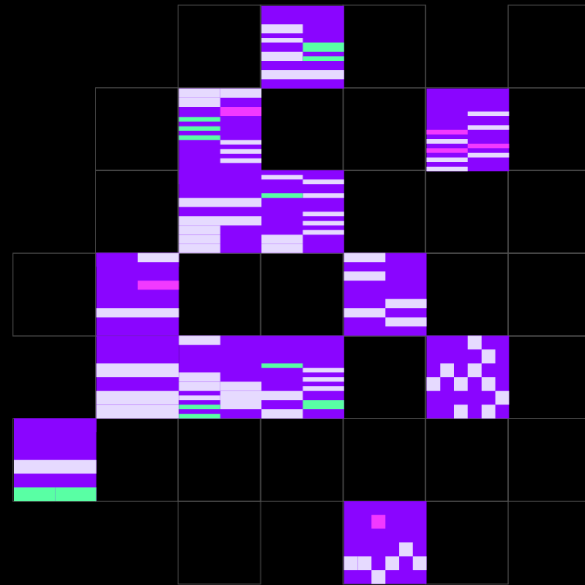
Step 3: 9x8

Les Défis Techniques

- Gestion de la logique de jeu sur une grille dynamique (taille variable).
- Mise à jour dynamique et synchronisée de l'affichage pour tous les joueurs.
- Calcul des victoires et des alignements sur des dimensions changeantes.

Essential MCP
Servers for
Developers

 Render



Le Cœur du Projet : Multijoueur en Temps Réel

Réel

Permettre aux joueurs de s'affronter partout, à tout moment.

Hébergement Distant

Jeu accessible depuis
n'importe quel navigateur
via un serveur Go distant.



Interaction Synchronisée

Actions et état du jeu mis à
jour en **temps réel** pour tous
les participants.

Technologie de Communication

Utilisation des **WebSockets** pour une communication
bidirectionnelle rapide et fiable.

Architecture Client–Serveur Détaillée

Séparation claire des responsabilités pour une robustesse maximale.



Client (HTML/JS)

Rôle : **Affichage et Interaction.**
Envoie les coups du joueur.



Serveur (Go)

Rôle : **Logique et Synchronisation.**
Gère les tours, valide les coups, et met à jour le jeu.

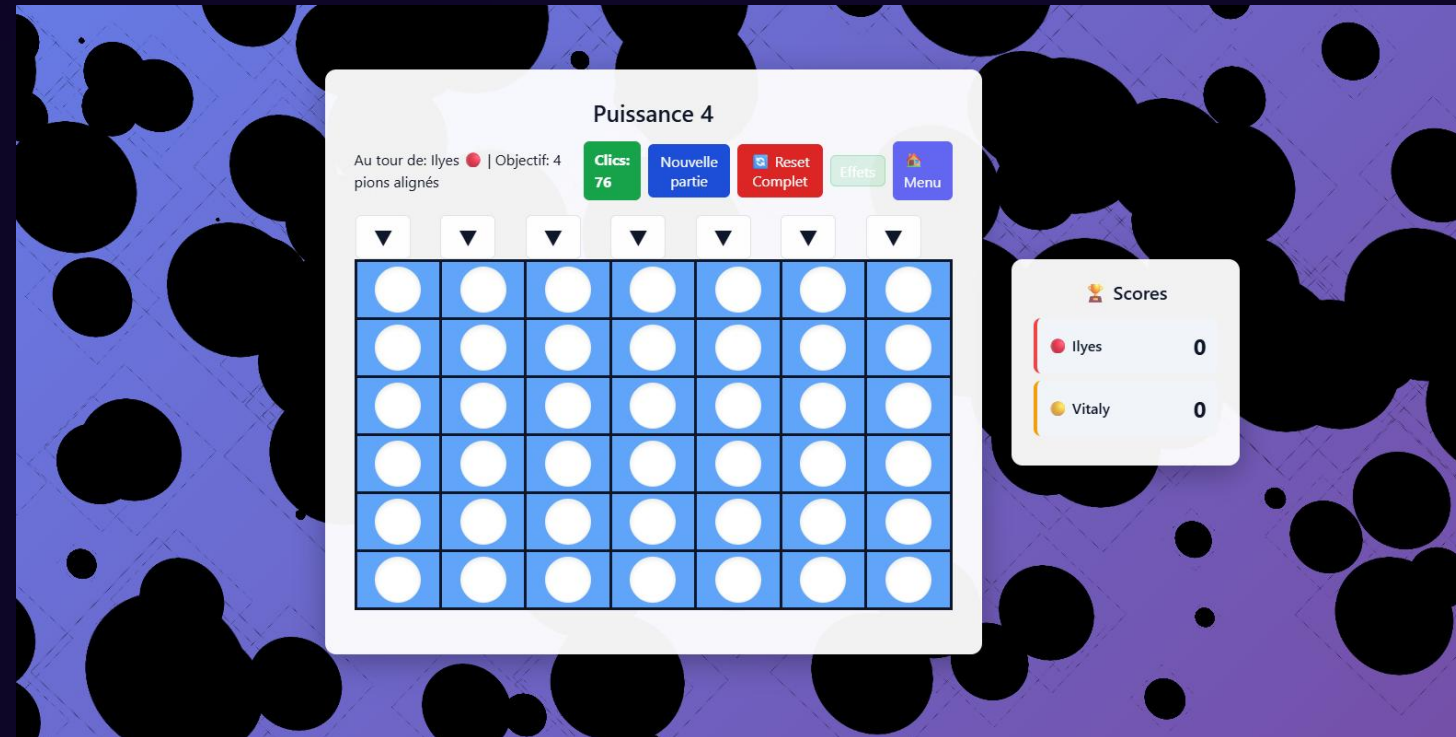


Canal de Données

WebSockets. Assure la fluidité des parties multijoueurs.

Interface Utilisateur et Expérience Visuelle

Un design épuré au service de la concentration et de la clarté.



Design Épuré

Simple et intuitif. Réduction du bruit visuel pour se concentrer sur le jeu.

Indicateurs Clairs

Le joueur actif et le mode de jeu choisi sont toujours affichés de manière visible.

Conclusion

1

Projet Complet

Maîtrise de l'ensemble de la chaîne : du développement serveur Go à l'interface web.

4

Compétences Acquises

Go, JavaScript, communication réseau, et hébergement de services en ligne.