

# Serveur de Matchmaking Puissance 4

# Serveur de Matchmaking Puissance 4

Un système de matchmaking pour le jeu Puissance 4, avec un serveur en Python et une interface client en Tkinter.

## Structure du projet

---

```
.
├── client/ # Code du client
│   ├── client.py # Client avec interface Tkinter
│   └── server/ # Code du serveur
│       ├── server.py # Serveur principal
│       ├── database.py # Gestion de la base de données
│       └── shared/ # Code partagé
│           ├── models.py # Modèles de données
│           └── requirements.txt # Dépendances Python
```

---

## Prérequis

- Python 3.7 ou supérieur

- pip (gestionnaire de paquets Python)

# Utilisation

## Démarrage du serveur

1. Ouvrez un terminal et naviguez vers le dossier du projet
2. Lancez le serveur :

```
bash
```

```
python server/server.py
```

Le serveur démarrera sur `127.0.0.1:5000` par défaut.

## Démarrage des clients

1. Ouvrez un nouveau terminal
2. Lancez le client :

```
bash
```

```
python client/client.py
```

3. Répétez l'étape 2 dans un autre terminal pour lancer un deuxième client

## Comment jouer

1. Entrez un nom d'utilisateur dans chaque client

2. Cliquez sur "Se connecter"
3. Attendez qu'un adversaire se connecte
4. Cliquez sur une colonne pour placer votre jeton
5. Le premier à aligner 4 jetons gagne !

## Fonctionnalités

- Matchmaking automatique
- Interface graphique intuitive
- Gestion des connexions et déconnexions
- Sauvegarde des parties dans une base de données SQLite
- Système de logging pour le débogage

## Logs

Les logs sont sauvegardés dans les fichiers suivants :

- `server.log` : Logs du serveur
- `client.log` : Logs du client
- `matchmaking.db` : Base de données SQLite

## Développement

### Structure du code

- ``shared/models.py`` : Définit les modèles de données (Player, Match, etc.)
- ``server/database.py`` : Gère la persistance des données
- ``server/server.py`` : Gère les connexions et la logique du jeu
- ``client/client.py`` : Interface utilisateur et communication avec le serveur

## Ajouter de nouvelles fonctionnalités

1. Modifiez les modèles dans ``shared/models.py`` si nécessaire
2. Mettez à jour la base de données dans ``server/database.py``
3. Implémentez la logique côté serveur dans ``server/server.py``
4. Ajoutez l'interface utilisateur dans ``client/client.py``