

Objectif du Module

Ce module permet de stocker de manière **sécurisée** et **immuable** les scores des joueurs d'un tournoi Pong sur une **blockchain**. Dans notre cas, nous utilisons le réseau **Ethereum (testnet Holesky)** pour déployer un **smart contract**. Cela garantit que les scores des joueurs sont enregistrés de manière transparente et accessible.

Description du Smart Contract

Le contrat intelligent est responsable de :

- Stocker les noms des joueurs et leurs scores associés
- Gérer les données du tournoi (nom, liste des joueurs, scores)
- Permettre la récupération des informations

Fonctionnement

Structure des Joueurs (Player)

- **name** : Nom du joueur (*string*)
- **score** : Score du joueur (*uint256*)

Tableau des Joueurs

- Tous les joueurs sont stockés dans un **tableau dynamique** (`players[]`)
- Chaque joueur est ajouté avec son **nom** et son **score** lors du déploiement du contrat

Variables Globales

1. **tournamentName** : Nom du tournoi, défini au moment du déploiement
2. **admin** : Adresse de l'administrateur du contrat (celui qui déploie)

Fonctions Principales

1. **constructor**
 - Appelé lors du déploiement
 - Initialise :
 - Le nom du tournoi
 - La liste des joueurs (noms)
 - Les scores associés
2. **getPlayer(uint256 index)**
 - Récupère les informations d'un joueur à un index donné
 - Retourne : **Nom** et **Score**
3. **getPlayersCount()**
 - Retourne le **nombre total de joueurs** enregistrés
4. **tournamentName()**
 - Retourne le **nom du tournoi**
5. **admin()**

- Retourne l'**adresse de l'administrateur** du contrat

Déploiement du Contrat

Le contrat est déployé sur le réseau de test Ethereum **Holesky** à l'aide d'un **script Python** utilisant la librairie **Web3py**

Étapes de Déploiement

1. Compilation du Contrat

- Utilisation de la librairie **solcx**
- Génère deux éléments :
 - **Bytecode** : Code binaire du contrat
 - **ABI (Application Binary Interface)** : Interface pour interagir avec le contrat

2. Envoi de la Transaction

- Le script Python envoie une transaction contenant :
 - Le bytecode du contrat
 - Les informations nécessaires (nom du tournoi, joueurs, scores)
- Une fois validée, le contrat est déployé sur la blockchain

3. Adresse du Contrat

- Une **adresse unique** est générée pour le contrat déployé
- Cette adresse est utilisée pour interagir avec le contrat (récupérer les scores, etc)

Récupération des Informations

Une fois le contrat déployé, les données peuvent être récupérées via les fonctions suivantes :

1. Nom du Tournoi

- Utilisez la fonction `tournamentName()` pour obtenir le nom du tournoi

2. Nombre de Joueurs

- Utilisez la fonction `getPlayersCount()` pour connaître le nombre de joueurs inscrits

3. Détails d'un Joueur

- Utilisez `getPlayer(index)` pour récupérer les informations d'un joueur à un index donné :
 - **Nom**
 - **Score**