

Sujet de Projet : Développement d'un Casino Décentralisé avec Roulette et Oracle Chainlink

Contexte et Objectif du Projet

Les casinos traditionnels reposent sur des systèmes centralisés où l'utilisateur doit faire **confiance** à l'opérateur. Avec la blockchain, nous pouvons créer un **casino transparent et décentralisé**, où **l'aléatoire est garanti par un Oracle Chainlink** et où les **transactions sont vérifiables** sur Ethereum.

Votre mission est de **développer un casino décentralisé** avec un jeu de **roulette basé sur un Oracle Chainlink pour générer des nombres aléatoires**. Un **frontend interactif en Typescript** permettra aux utilisateurs de **placer leurs mises** et de **recevoir leurs gains** de manière sécurisée.

Fonctionnalités Attendues

Contrat Smart Contract du Casino (Solidity)

- Un **contrat principal** gérant les **mises, le tirage et le paiement des gains**.
- **Gestion des dépôts et retraits** en ETH ou USDC.
- **Vérification des mises** pour éviter les abus (limites min/max).

Roulette avec Oracle Chainlink

- Utilisation d'un **VRF Chainlink (Verifiable Random Function)** pour garantir **l'aléatoire**.
- Génération d'un **numéro entre 0 et 36**, respectant les règles classiques de la roulette.
- Affichage **transparente des résultats** pour éviter toute triche.

Gestion des Gains et du Trésor

- Calcul automatique des **paiements en fonction des mises**.
- Les **pertes des joueurs** sont stockées dans un **trésor du casino**.
- Une **commission de 5%** sur les gains est envoyée vers un **Gnosis SAFE** pour la gestion des fonds du casino.

🔧 Frontend en Typescript avec WalletConnect

- Interface utilisateur permettant de :
 - ✅ **Se connecter avec WalletConnect** pour déposer et miser.
 - ✅ **Placer des mises sur différents types de paris** (numéro, couleur, pair/impair, etc.).
 - ✅ **Afficher en direct les résultats de la roulette avec un historique des parties.**
 - ✅ **Effectuer des retraits des gains gagnés.**
-

📌 Contraintes Techniques

- **Solidity (v0.8.x) avec Foundry**
 - **Chainlink VRF** pour l'aléatoire de la roulette.
 - **Gnosis SAFE** pour la gestion des fonds.
 - **Typescript + Next.js (ou React.ts)** pour le frontend.
 - **WalletConnect** pour les transactions et l'authentification des joueurs.
 - **Tests unitaires avec Foundry** pour garantir la sécurité du contrat.
-

📄 Évaluation du Projet

💯 Critères de notation

- **40% Smart Contracts** (sécurité, fonctionnement de la roulette, gestion des mises).
 - **30% Frontend** (expérience utilisateur, intégration WalletConnect).
 - **20% Tests et Déploiement** (tests unitaires avec Hardhat/Foundry).
 - **10% Documentation** (explication du code, guide utilisateur).
-

🚀 Bonus (optionnel) :

- ✅ Ajout d'autres jeux de casino (blackjack, poker).
- ✅ Intégration de **Chainlink Price Feeds** pour ajuster les paiements en stablecoins.
- ✅ Mise en place d'un **token de fidélité** pour récompenser les joueurs.

✅ Système de révélation de clé privée en cas de pertes au poker ou autres jeux, mais bonus quadruplé

📅 **Deadline : A définir**

100 **Notation : Note de suivi et soutenance final**

🎯 **Objectif final : Déployer un prototype fonctionnel sur un testnet Ethereum (Sepolia/Goerli).**

Bonne chance ! 🎲 🎮 🚀