

## Sujet : Scam Swap

### Contexte :

Dans le cadre de la finance décentralisée (DeFi), le développement d'un smart contract pour gérer un système d'usines de tokens, de pools de liquidité et de mécanismes d'échange est une étape clé pour permettre une expérience utilisateur fluide et un écosystème robuste.

Le projet nécessite la création d'une plateforme où les utilisateurs peuvent :

1. Générer leurs propres tokens ERC20 grâce à une **factory de tokens**.
2. Créer des pools de liquidité à l'aide d'une **factory de pools**.
3. Effectuer des swaps avec des **frais fixes de 2%**, qui sont partiellement redistribués aux fournisseurs de liquidité.
4. Investir dans les pools pour devenir fournisseur de liquidité et ainsi profiter d'une partie des frais de swap.
5. Calculer et mettre à jour dynamiquement le **ratio** de liquidité pour refléter les échanges et les investissements.

---

### Spécifications Fonctionnelles :

#### 1. Usine de Tokens (Token Factory) :

- Permettre aux utilisateurs de déployer un contrat ERC20 personnalisable (nom, symbole, supply initiale).
- Enregistrer automatiquement les tokens créés pour faciliter leur utilisation dans les pools de liquidité.

#### 2. Usine de Pools (Pool Factory) :

- Permettre la création de pools de liquidité entre deux tokens ERC20 (Token A et Token B).
- Initialiser chaque pool avec une liquidité minimale des deux tokens.
- Assurer le suivi des pools actifs et leurs ratios de liquidité.

#### 3. Mécanisme de Ratio :

- Calculer le **ratio** de liquidité de chaque pool en fonction des réserves actuelles (par exemple,  $\text{ratio} = \text{reserveA} / \text{reserveB}$ ).

- Mettre à jour dynamiquement ce ratio après chaque swap ou ajout/retrait de liquidité.

#### **4. Swap avec Frais (2%) :**

- Autoriser les échanges de Token A vers Token B (et inversement) avec un calcul précis des frais.
- Redistribuer un pourcentage des frais aux fournisseurs de liquidité proportionnellement à leur part dans la pool.

#### **5. Investissement dans la Pool :**

- Permettre aux utilisateurs d'investir dans une pool en y ajoutant des liquidités (Token A et Token B).
- Émettre des "Pool Tokens" représentant la part de l'utilisateur dans la pool.

#### **6. Redistribution des Frais :**

- Récolter 2% de chaque swap en tant que frais.
- Redistribuer 1% aux fournisseurs de liquidité, et 1% dans une réserve de trésorerie (ou un fonds de gouvernance).

#### **7. Retrait de Liquidité :**

- Permettre aux utilisateurs de retirer leurs parts de liquidité.
- Récupérer leur part proportionnelle des tokens ainsi qu'un pourcentage des fees accumulés.

#### **8. Création de Pools piégées avec des tokens piégés**

- S'assurer une réussite des fonctions précédentes pour ce type de token
- Trouver un mécanisme permettant de récupérer les fonds de n'importe quel investisseur

---

#### **Contraintes Techniques :**

- Respecter le standard ERC20 pour les tokens.
- Protéger les pools contre des attaques courantes (par exemple, protection contre les flash loans avec des ratios manipulés ou encore certaines failles permettant de voler les fonds d'une pool).
- Éviter les erreurs d'arrondi dans les calculs de ratio et de frais.

---

## **Étapes de Développement :**

### **1. Développement des Contrats :**

- Développer le contrat ERC20 standard pour les tokens personnalisés.
- Implémenter la Token Factory et la Pool Factory avec des mappings pour suivre les actifs créés.
- Implémenter le mécanisme de swap avec les calculs de ratio et de frais.

### **2. Développement des Fonctions Secondaires :**

- Ajout de la fonctionnalité d'investissement et de retrait de liquidité.
- Gestion de la distribution des frais.
- Toutes fonctionnalités supplémentaires ou curiosités techniques seront récompensées.

---

## **Objectif Final :**

Créer un smart contract DeFi complet, évolutif et sécurisé permettant une interaction fluide entre les utilisateurs, leurs tokens personnalisés, et des pools de liquidité dynamiques.