SUJET ARTEFHACK - BLOCKCHAIN ET MARKETING ETHIQUE

1. Blockchain et marketing éthique

Artefact explore l'avenir du digital marketing grâce à la technologie de la blockchain! La blockchain permet de rendre le web plus transparent sûr, et offre des opportunités d'application au digital marketing.

Le projet **ArtefHack** vise à établir un équilibre entre les différents acteurs du marché :

- l'**Advertiser** veut partager son message à un public le plus large possible
- le **Publisher** veut vendre ses contenus
- les **Users** veulent consommer des contenus qui leur plaisent

2. Fonctionnement de la plateforme

La plateforme **ArtefHack** fonctionne de la façon suivante:

- Un ensemble de contenus sont mis en vente par le **Publisher**
- Les **Users** visitent **ArtefHack** et reçoivent en retour un contenu et éventuellement un message publicitaire
- **ArtefHack** choisit les contenus à montrer aux **Users**, et peut choisir de leur montrer un message publicitaire
- **ArtefHack** peut acheter des nouveaux contenus à montrer aux **Users**, et rémunère le Publisher en conséquence
- L'**Advertiser** paye **ArtefHack** à chaque impression d'un message publicitaire
- **ArtefHack** apprend sur les préférences de l'utilisateur à chaque visite

Données du problème :

- **Users**
    - Préférence (uint): comprise entre 0 et 100 qui détermine le type de contenu qu'ils aiment
    - Appétence aux messages publicitaires (bool)
- **Contenus**:
    - Préférence: comprise en 0 et 100, qualifie le contenu
- **Balances**:
    - Gèrent les montants de tokens détenus par chaque address
    - Gèrent les transferts entre addresses avec la fonction **pay**
    - Initilisation des balances:
        - **ArtefHack** reçoit 2000 tokens
        - **Advertiser** reçoit 50000 tokens
- Prix des transactions:
    - Achat d'un contenu : 50 - **ArtefHack** paye le **Publisher**

- Affichage d'un message publicitaire : 10 - l'**Advertiser** paye **ArtefHack**

3. Implémentation des fonctions **visit** et **eval**

L'objectif de l'**ArtefHack** est de maximiser le nombre de visites sur la plateforme:
- les utilisateurs qui aiment contenus visités viendront plus souvent
- les utilisateurs qui n'aiment pas la publicité
- d'acheter des contenus aux publishers

Les fonctions à implémenter:
- **Visit**:
  - appelée par l'utilisateur
  - renvoie un contenu et éventuellement un message publicitaire
- **Eval**:
  - appelée par l'utilisateur
  - prend en argument entre autres l'évaluation qu'il a fait du contenu/message qui lui a été présenté auparavant

Les users se comportent de la manière suivante. Ils renvoient:
- une **évaluation négative** s'ils ont déjà vu le content qui leur est présenté
- une **évaluation négative** si ArtefHack leur renvoie un message publicitaire alors que cela les dérange
- une **évaluation positive** si |(préférence(contenu)-préférence(utilisateur)|<10, négative sinon

ArtefHack ne connaît pas les préférences et l'appétence aux messages publicitaires des users au début de la simulation.

4. Simulation et test de votre implémentation

Opérations à effectuer pour tester votre implémentation:

- lancer testrpc : testprc -a 504 -l 5000000
- Déployer les contrats :
  - déploiement sur testrpc déjà lancé : truffle migrate
  - ne pas oublier de rm -rf build/contracts/ si vous avez déjà déployé auparavant et de relancer testrpc
- Lancer la simulation :
  - depuis le dossier test/
  - activer le virtualenv python3 : source ~/.venv-py3/bin/activate
  - commande : python3 simulate.py --setup True --starting-users-count s --max-users-count m --contents-count c --days-count d

- attention à mettre des valeurs faibles dans un premier temps pour que la simulation ne dure pas trop longtemps

5. Soumission des résultats

Les candidats doivent fournir une implémentation des fonctions visit et eval du fichier artefHack.sol ainsi que les contrats nécessaires à leur fonctionnement:

- une archive zip contenant l'ensemble des fichiers
    - Les contrats dans un dossiers 'contracts'
    - Un fichier de déploiement truffle: '2_deploy_contracts.js'
- archive à envoyer sur le chat Discord à Diplomate (Ulysse)
- votre implémentation sera évaluée avec un jeu de données différent de celui dont vous disposez (mais suivant les mêmes règles de génération)
- paramètres d'évaluation : 3 scénarios seront testés et le score global sera la somme des scores sur chacun des scénarios.
    - contents-count : 50 / 100 / 400
    - starting-users-count : 10 / 20 / 50
    - max-users-count : 10 / 50 / 300
    - days : 5 / 20 / 50
    - timeout: 1 minute / 10 minutes / 30 minutes
- formule du score: nombre de visites + nombre de visites satisfaisantes + 1/10 * publisher_token