IFT-4100/7100 : Aspects pratiques de la chaîne de blocs

TP3 : L'application décentralisée du siècle !

Date d'échéance : 21 avril 2024

1. Objectifs

Ce travail a pour principal objectif de vous faire pratiquer le développement d'applications décentralisées (dApps) et de contrats intelligents sécurisés. Vous vous servirez du langage Solidity ainsi que de la grande variété d'outils de développement existants (Truffle, Ganache, Web3.js, etc).

Quelques points importants :

- Bien vrai que vous ayez la possibilité de faire ce TP de façon individuelle, il est **FORTEMENT** recommandé de travailler en **équipe de 2 personnes**.
- Laisser son coéquipier faire tout le travail (peu importe les raisons) est inacceptable. Vous passerez à côté des objectifs de ce cours. De la même manière, il ne faut pas non plus trop en faire. Il faut apprendre à travailler en équipe !
- Vous vous verrez attribuer la note de zéro si vous rendez comme travail un projet copié ou cloné via des plateformes comme GitHub, Bitbucket, etc.

2. Travail à faire

2.1. Application décentralisée

Vous devez concevoir une dApp respectant les critères suivants :

- Elle devra être composée d'au minimum un contrat intelligent.
- Elle devra disposer d'une interface utilisateur. Vous avez le choix entre une interface en ligne de commande, une interface graphique ou une interface Web.
- Votre contrat intelligent devra définir au minimum 2 variables d'état.
- Votre contrat intelligent devra définir et émettre au minimum un événement. L'évènement devra être écouté du côté du front-end et exécuter une certaine logique à la réception de l'évènement.
- Votre contrat intelligent devra faire usage au minimum des types `mapping` et `uint`.
- Votre contrat intelligent devra contenir au minimum une fonction marquée comme étant `payable`.
- Votre contrat intelligent devra faire usage de `msg.sender` et `msg.value`.

- Votre contrat intelligent devra définir et appliquer un `modifier` de fonction. Ex : `onlyOwner`.
- Votre contrat intelligent devra être déployé sur un réseau Ethereum de test.
- Votre contrat intelligent devra être « *upgradable* ».

Si vous décidez de concevoir une interface Web et que vous souhaitez la déployer, nous vous recommandons l'usage de plateformes comme Vercel, Fly.io, Render qui proposent des plans gratuits. Vous êtes toutefois libres d'utiliser n'importe quelle autre plateforme de déploiement. Rien ne vous oblige également à déployer votre application Web.

2.2. Rapport vidéo

Vous devez également fournir un rapport vidéo détaillant les points suivants :

- Justification des choix des différentes technologies utilisées au sein de votre application ;
- Justification des choix des différentes librairies utilisées au sein de votre application (si vous avez choisi d'utiliser des librairies) ;
- Énumération et justification des différentes étapes du cycle de développement logiciel sécuritaire que vous avez suivies. Par exemple, si vous n'avez pas effectué d'étape d'analyse, il faudra en quelques phrases en expliquer la raison ;
- Mention des vulnérabilités contre lesquelles votre application est protégée et les techniques utilisées afin de mettre en place ces protections. Votre application devra au minimum être protégée contre l'ensemble des vulnérabilités étudiées en classe. Des points supplémentaires seront attribués si votre application est protégée contre d'autres vulnérabilités que nous n'avons pas eu l'occasion d'étudier en classe. Il faudra donc les mentionner de manière explicite dans votre vidéo.

3. Modalités d'évaluation

Ce travail sera évalué sur 90 points, et la note sera ramenée sur 12. Voici la grille de correction :

Critère	**Pondération**
Rapport vidéo (Qualité des réponses, explications et justifications)	20 points
Interface utilisateur + Contrat intelligent	40 points
Déploiement du contrat intelligent sur un réseau Ethereum de test	20 points
Qualité du code (noms de variables, nom des fonctions, etc)	10 points

Notez qu'une application qui n'est pas fonctionnelle (qui ne s'exécute pas ou qui plante à l'exécution) pourrait recevoir une note de 0.

Vous devez remettre une archive Zip (fichier avec extension *.zip*) contenant le rapport vidéo, le code de votre application et le fichier de correction *fichier_de_correction.txt* rempli (voir la section « *Ce que vous devez rendre* » pour plus de détails). Si vous avez du mal à y arriver ou si vous êtes confus, nous vous invitons à exploiter les ressources qui sont mises à votre disposition pour vous aider : posez des questions sur le forum du cours, consultez l'enseignant lors de ses périodes de disponibilités, etc.

4. Remarques

Plagiat : Tel que décrit dans le plan de cours, le plagiat est strictement interdit. Ne partagez pas votre code source à quiconque. Une politique stricte de tolérance zéro est appliquée en tout temps et sous toute circonstance. Tous les cas détectés seront référés à la direction de la faculté. Des logiciels comparant chaque paire de TPs pourraient être utilisés pour détecter les cas de plagiat.

Retards : Tout travail remis en retard peut être envoyé par courriel à l'enseignant si le portail des cours n'accepte pas la remise. Une pénalité de 10% sera appliquée pour un travail remis le lendemain de la remise, puis une pénalité de 25% sera attribuée à un TP remis le surlendemain. Une note de 0 sera attribuée pour un plus long retard.

Remises multiples : Il vous est possible de remettre votre TP plusieurs fois sur le portail des cours. La dernière version sera considérée pour la correction.

Respect des normes de programmation : Nous vous demandons de prêter attention au respect des normes de programmation établies dans le langage Solidity, notamment de nommer vos variables et fonctions en utilisant la notation camelCase. Ex : `maMariable`, `maFonction()`, etc.

5. Ce que vous devez rendre

Vous devez remettre une archive **zip** d'un dossier, contenant **uniquement** les éléments suivants :

- Le rapport vidéo d'une durée maximale de 20 min. Pour le style de la vidéo, il serait plus intéressant que ce soit comme un échange naturel entre deux personnes travaillant dans la même entreprise. Pour enregistrer la vidéo et la voix, vous pouvez utiliser des logiciels comme QuickTime, OBS Studio, Zoom, etc. Il est important d'effectuer un test audio/vidéo préalable afin de vous assurer que les bons périphériques ont été sélectionnés.
- Un dossier ***code*** contenant le code de votre application décentralisée (Interface utilisateur + Contrat intelligent).
- Le fichier de correction ***correction.txt*** fourni avec l'énoncé, que vous devez compléter en y indiquant votre nom, votre NI, et le nombre d'heures que vous avez consacré à

compléter votre TP. Vous perdrez des points si ce fichier n'est pas fourni ou n'est pas complété.

Cette archive devra être remise via le site Web du cours. Une pénalité de 5% pourra être appliquée si vous remettez des fichiers inutiles.

PS : En lieu et place de mettre le code de votre application dans le dossier code, vous pouvez simplement rajouter un fichier texte dans le dossier code contenant un lien vers votre projet dans un référentiel de code distant sur GitHub ou encore Bitbucket. Il faudra toutefois vous assurer de donner accès à votre projet au correcteur, en utilisant son adresse courriel : olufemi-honore-etsmoberg.hounwanou.1@ulaval.ca.