BilleGateCoin - BGC

Scattered Corporation

réalisé par

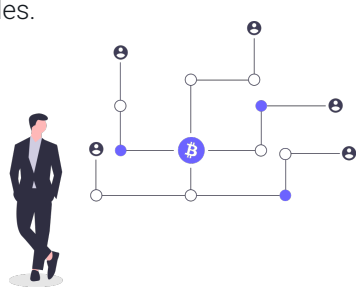
aurele.oules • leo.gervoson • raphael.brenn • frederic.dong

Projet EPITA 2020

Table des matières

1 Introduction

BilleGateCoin est une plateforme décentralisée pour un jeu de billes. Cette plateforme utilisera la technologie de la Blockchain pour sécuriser les parties. En 2008, Satoshi Nakamoto est le premier à résoudre le problème de la "double dépense" dans un système décentralisé : le Bitcoin. Cette technologie révolutionnaire et surprenante n'a cessé de prendre de plus en plus de place dans notre société. Nous avons décidé d'aborder cette technologie de la Blockchain de manière divertissante en l'appliquant à un jeu-vidéo dont les règles sont simples et les objets sont collectionnables : le jeu de billes.



1.1 Membres

1.1.1 Aurèle

Depuis que je suis tout petit, je suis passionné par la programmation et les jeux-vidéos. La Blockchain est une nouvelle technologie qui est encore en phase de développement et d'adoption. Ce projet de S2 à EPITA est l'occasion de montrer le potentiel de cette technologie émergente en l'appliquant au domaine du jeu vidéo. J'ai hâte de participer au développement de ce jeu original.

1.1.2 Léo

Ce projet m'intéresse d'une part car j'ai eu l'occasion d'en apprendre sur la blockchain lors des exposés d'anglais en présentant un exposé sur la blockchain, et d'autre part car je trouve l'utilisation de la blockchain pour la mise en place d'un jeu vraiment innovante et liant deux domaines habituellement séparés : les jeux et la blockchain.

1.1.3 Raphaël

Raphaël Brenn, 17 ans, j'ai une petite expérience en programmation (notamment pour des petits robots) et en modélisation 3D mais je n'avais jamais touché à Unity avant très récemment.

Je pense que créer ce type de jeu permet de montrer de façon concrète un autre secteur dans lequel il est possible d'utiliser la blockchain, par rapport à celui des crypto-monnaies, et plus largement expérimenter le développement d'un jeu vidéo selon un nouveau point de vue (ce qui permet à la fois de faire face à des problèmes inattendus, et contourner facilement d'autres problèmes qui auraient été plus gênants sans l'utilisation de cette technologie), ce que je trouve intéressant et potentiellement très enrichissant.

1.1.4 Frédéric

Depuis l'explosion médiatique qu'a apporté le Bitcoin, j'ai été très intéressé par la technologie du Bitcoin et de la Blockchain. Comment cette technologie permet de faire fonctionner plusieurs ordinateurs ensemble est tout simplement fascinant.

Lorsque j'ai entendu parler du projet, un jeu reposant sur ce principe, j'ai tout de suite été excité par le projet.

1.2 Exemples

Il existe plusieurs applications connues de la Blockchain :

- Bitcoin : système bancaire décentralisé
- Ethereum : machine virtuelle pour éxécuter du code de manière décentralisée
- CryptoKitties : jeu décentralisé fonctionnant sur Ethereum dont le but est de collectionner des chats virtuels

1.3 Implémentation

Le jeu de billes est un jeu classique dont le but est de collectionner des billes plus ou moins rares en combattant des adversaires ou en les échangeant. Une partie met en jeu les billes des joueurs : celui qui remporte la partie gagne la bille de son adversaire.

Chaque bille possède des propriétés physique différentes (adhérance, taille) et une rareté différente. La façon la plus classique de jouer au jeu de billes est de lancer sa bille sur celle de l'adversaire pour gagner.

Dans l'implémentation de ce jeu, le joueur crée la partie avec un adversaire sélectionné. Un terrain de jeu est généré avec des obstacles placés aléatoirement et les deux billes des joueurs sont placées de part et d'autre du terrain. Le joueur lance sa bille en direction de celle de l'adversaire. Si ce joueur touche sa bille, il la remporte. Sinon c'est au tour de l'adversaire de lancer sa bille.

Nous allons évoquer les raisons de la décentralisation de la plateforme.

2 Décentralisation

La décentralisation de la plateforme offre une multitude d'avantages en comparaison à un système centralisé "normal".

2.1 Confiance

Les joueurs n'ont pas à faire confiance à une autorité centrale. *Don't trust, verify.*

Le processus de création des billes ainsi que la définition de leur rareté est entièrement transparent et contrôlé par les joueurs mêmes, plutôt qu'une entreprise privée. Le taux de billes créées est fixé par les joueurs et personne ne peut le modifier sans le consensus général de la communauté. Personne ne peut confisquer les billes d'un joueur, même pas les développeurs.

2.2 Point of Failure

Dans un système décentralisé, il n'existe pas de point unique de défaillance (*single point of failure*).

Dans une plateforme centralisée, si le système est défaillant, tout le réseau est impacté. Contrairement à un système décentralisé où il ne suffit d'un seul ordinateur pour assurer la stabilité du réseau.

2.3 Ouvert au développement

N'importe quel individu avec des notions en programmation peut contribuer au développement de la plateforme. Si assez de joueurs (consensus) acceptent ces changements alors ce code est ajouté à la plateforme.

Rien n'empêche un joueur de cloner cette plateforme afin de créer un jeu avec des règles différentes : *Hard Fork*.

3 Fonctionnement

Afin de permettre la décentralisation du jeu, nous utiliserons la technologie de la Blockchain.

La *Blockchain* est un régistre distribué (*distributed ledger*) immuable et sécurisé par des milliers d'ordinateurs.

3.1 Utilisateurs

Sur une plateforme de jeu décentralisée, il n'y a pas de système de comptes avec une adresse email et un mot de passe. Chaque joueur doit alors se générer une clé privée ainsi qu'une clé publique correspondante, avec un algorithme de cryptographie tels que RSA ou *Elliptic Curve Digital Signature Algorithm* (ECDSA).

Cette clé privée doit être précieusement cachée car elle permet de signer toutes les transactions du joueur, c'est-à-dire créer une partie, lancer sa bille, échanger des billes, etc. L'utilisateur aura donc la possibilité de chiffrer sa clé privée sur son ordinateur et de la déchiffrer avec un mot de passe lors de la signature d'un contrat.

3.2 Lancement de la partie

Deux modèles sont envisageables pour la création d'une partie.

3.2.1 Deux étapes

Le joueur choisit la bille qu'il veut jouer, son adversaire (sa clé publique) et la bille de son adversaire (qui lui appartient). Le joueur signe l'invitation avec sa clé privée et la diffuse sur la blockchain. L'adversaire reçoit alors une invitation pour créer cette partie, s'il accepte la partie, il doit également signer le contrat avec sa clé privée sur la blockchain. La partie commence.

Ce modèle assume que les deux joueurs n'ont pas de moyen de communication direct et utilisent la blockchain comme plateforme pour trouver des joueurs.

Ce modèle est couteux car il demande deux contrats sur la blockchain (création de la partie, et acceptation de la partie).

3.2.2 Double signature

Le joueur choisit la bille qu'il veut jouer, son adversaire (sa clé publique) et la bille de son adversaire (qui lui appartient). Le joueur signe ce contrat avec sa clé privée et demande à l'adversaire de signer également ce contrat avec sa clé privée. Une fois le contrat signé par les deux joueurs, le contrat est diffusé sur la blockchain. La partie commence.

Ce modèle assume que les deux joueurs ont un moyen de communication direct.

Ce modèle est moins couteux pour le joueur car l'accord entre les deux joueurs est fait hors-blockchain (*off-chain*). La création d'une plateforme centralisée permettant aux joueurs de trouver un accord est possible. La centralisation de cette plateforme ne pose pas de problèmes sur l'aspect décentralisé du jeu en lui-même.

Lors de la création de la partie, les joueurs doivent donner une ou plusieurs billes aux *mineurs* afin de traiter la transaction (*transaction fee*). Cette taxe permet aux mineurs d'être rémunérés pour leur travail. Nous implémenterons ce contrat à double signature plutôt que le précedent.

3.3 Source d'entropie

Lorsque la partie est créée, des obstacles sont aléatoirement placés sur le terrain pour ajouter de la difficulté au jeu de billes. Comme dans la vraie vie, il peut y avoir des trous, des arbres, des obstacles. Comment pouvons-nous générer un nombre aléatoire dans une Blockchain ? Nous ne pouvons-pas générer de nombres aléatoires sur les nodes car le résultat serait différent pour tout le monde, alors nous devons utiliser une autre source d'entropie.

3.3.1 Solution

Lorsque le contrat de la création de la partie est signé, celui-ci est diffusé dans la *mempool* des ordinateurs qui assurent la sécurité du réseau : *nodes*. Les mineurs vont alors essayer de former un block avec les contrats et transactions de la mempool.

Pour se faire, les mineurs vont calculer un hash *SHA-256* en concatenant toutes les valeurs du block telles que la date du block, les données de chaque contrat et l'indice du dernier block. Si ce block contient un certain nombre *(mining difficulty)* de '0' en tête de hash, alors ce hash est valide et le block est dit "miné".

Ce hash n'est qu'un nombre hexadécimal aléatoire compris entre 1 et la *mining difficulty*. Ce nombre va permettre la génération des obstacles la partie, en fonctionnant comme un *seed*.

3.3.2 Limites

Toutes les parties créées dans ce block auront alors le même seed, c'est-à-dire l'exacte même répartition des obstacles.

Il suffit de régler le *block time* de la blockchain à un temps court pour permettre une génération de block plus rapide et donc des parties plus diversifiées.

3.4 Déroulement de la partie

Une fois la partie créée, si le seed de la partie est pair c'est au joueur 1 de commencer, sinon c'est au tour du joueur 2.

Pour lancer sa bille, le joueur doit cliquer sur sa bille, et la tirer avec une certaine force. Un début de trajectoire s'affichera pour que le joueur puisse voir approximativement la direction de sa bille. Une fois la bille lâchée, un vecteur force est créé. Le joueur doit alors signer ce vecteur force avec sa clé privée. Celui-ci est exécuté sur l'écran du joueur, il voit alors sa bille se déplacer. Le vecteur signé est également diffusé sur la Blockchain.

Les nodes et les mineurs vont vérifier la nouvelle position de la bille en fonction du vecteur signé par le joueur. Si la bille touche celle de l'adversaire alors la partie se termine et il remporte toutes les billes. Dans le cas contraire la partie continue et c'est au tour de l'autre joueur.

Pour éviter des problèmes de précision, la nouvelle position de la bille n'est jamais enregistrée dans la Blockchain. Ainsi pour déterminer l'état actuel de la partie (la position des deux billes) il faut recalculer la position de chaque bille grâce aux précédents vecteurs force signés par les joueurs respectifs.

Plus la partie est longue, plus le calcul de la position de la bille est coûteux en ressources, les taxes de transactions seront alors de plus en plus élevées. Les joueurs doivent se mettre d'accord en début de partie du nombre de tours maximum, si ce nombre est depassé pendant la partie, celle-ci se termine et les joueurs reprennent leur bille.

3.5 Fin de partie

Lorsqu'un joueur parvient à toucher la bille de l'autre, celui-ci gagne la partie. Pour récupérer ses gains (la bille de l'adversaire ainsi que sa propre bille), le joueur doit diffuser sur la blockchain une *claim transaction*. Tant que le gagnant n'a pas diffusé cette transaction, il ne peut pas récupérer les mises.

Ainsi pour vérifier l'appartenance d'une bille, les nodes doivent uniquement parcourir la liste des *claim transactions* au lieu de recalculer l'état de la partie et vérifier que les billes se touchent. Cela permet également de constituer rapidement l'historique d'appartenance de la bille.

Ce contrat peut être diffusé uniquement si la partie est terminée, et la clé publique de la signature correspond à la clé publique du gagnant.

3.6 Nodes

Une *node* est un ordinateur connecté à tout le réseau de la blockchain. Elle a un rôle majeur sur la sécurité du réseau.

3.6.1 Arbitre

La node possède le rôle d'arbitre. Lorsque une partie est créée ou un joueur lance sa bille, la node va vérifier si le joueur possède bien cette bille, si c'est bien au tour du joueur de jouer, s'il n'essaye pas de lancer sa bille plus fort que la limite, etc. Si le contrat est valide, celui-ci rentre dans la *memory pool* de la node, sinon il est rejeté.

3.6.2 Régistre

La node s'occupe de stocker l'historique entier de toutes les parties et transactions de tous les joueurs, c'est-à-dire la blockchain.

L'ordinateur réserve également un emplacement pour stocker les transactions non confirmées par les mineurs, c'est-à-dire les lancés de billes et créations de parties qui viennent d'être diffusées par les joueurs. Cet emplacement s'appelle la *memory pool*.

3.6.3 Distribution

Le rôle principal de la node est de partager la blockchain à toutes les autres nodes du réseau. La node qui possède la plus grande chaîne est celle qui possède la chaîne la plus valide, elle doit donc la partager au réseau.

3.7 Mineurs

Les mineurs assurent l'immuabilité de la blockchain. Un block contient des contrats et des lancés de billes. Pour ajouter un block à la blockchain, un puzzle constitué exclusivement des données du block doit être résolu. Ce puzzle n'est solvable uniquement en essayant des milliards de combinaisons par secondes.

3.7.1 Block hash

Un hash [1] SHA-256 est un nombre hexadécimal aléatoire compris entre 1 et $2^{256} - 1$.

La solution du puzzle que les mineurs doivent résoudre est en réalité un hash du block qui doit être compris entre 1 et $\frac{2^{256}-1}{mining\ difficulty}$.

1. Une fonction hash est une fonction qui prend une entrée telle que du texte ou un nombre, et retourne un nombre qui apparaît aléatoire mais est déterministe. Pour la même entrée, la fonction retournera toujours la même sortie. C'est une fonction à sens unique, le hash ne permet pas de retrouver l'entrée.

Les mineurs forment un block à partir des contrats de la *memory pool* des nodes.

Si le hash n'est pas compris dans cet intervalle, les mineurs calculent à nouveau un hash du block en concatenant un *nonce* différent. C'est un nombre arbitraire qui permet uniquement de générer un hash différent en conservant les valeurs du block.

Ce hash sécurise la blockchain car il est facile de le résoudre si tout le réseau travail sur ce problème, mais il est impossible de résoudre ce problème seul. Ce système assume que plus de 50% du réseau est honnête.

Si un utilisateur malveillant voulait essayer de falsifier une transaction en l'ajoutant soi-même à la blockchain, c'est-à-dire en minant le block soi-même, il devrait alors résoudre ce puzzle extrêmement compliqué seul. De plus, aucune node n'accepterai cette transaction, sauf si l'utilisateur possède plus de la moitié des ordinateurs du réseau (*51% attack*) car l'utilisateur pourrait ainsi corrompre le consensus et former une nouvelle blockchain. Cette attaque est extrêmement peu probable.

3.7.2 Ajustement de la difficulté

La difficulté de minage est une variable qui assure que les mineurs ajoutent des blocks à la blockchain de manière régulière même si le nombre de mineurs, donc de puissance de calcul, augmente.

Cette difficulté permet aux nodes de prévoir l'espace disque nécessaire pour stocker la blockchain et assure une économie stable (voir section suivante). Elle doit donc être ajustée régulièrement.

Le *block time* est une constante qui définit le temps qu'un mineur met pour ajouter un block à la chaîne.

Sur la plateforme de jeu de billes, le block time doit être assez court (entre 20s et une minute) afin de permettre des exécutions de contrats (créations de parties, et lancés de billes) rapides. Par exemple, le Bitcoin utilise un block time de 10 minutes, et Ethereum un block time de 20 secondes.

Un intervalle trop court peut causer des problèmes sur le délai de distribution des nouveaux blocks entre les nodes.

La difficulté de minage est au départ 1. Celle-ci est ajustée tous les X blocks. Chaque node va calculer le temps que X block prennent pour être minés en théorie, divisé par la moyenne de temps que les X derniers blocks ont pris pour être minés. La difficulté est ainsi multipliée par ce coefficient.

$$nouvelle\ difficulté = difficulté \times \frac{temps\ attendu}{temps\ réel}$$

3.7.3 Récompenses

Dans chaque block miné, le mineur inclut une transaction spéciale : la *coinbase transaction*. C'est la récompense du mineur. Cette transaction contient uniquement l'adresse publique du mineur pour recevoir sa récompense. Celle-ci contient de nouvelles billes créées, toutes plus ou moins rares telles que 1000 billes grises, 100 billes bleues, et 50 billes rouges. Les rouges seraient par définition plus rares que les autres car il y en aurait moins en circulation. En plus de ces nouvelles billes, le mineur récupère toutes les taxes des transactions de chaque joueur (sous la forme de billes également).

Tous les X blocks (210 000 blocks pour Bitcoin ≈ 4 ans), les récompenses des mineurs sont divisées par deux jusqu'à que la totalité des billes soient distribuées. La prochaine récompense serait de 500 billes grises, 50 billes bleues, et 25 billes rouges.

4 Contrats (ébauche)

Placement de billes

octets	1	1	8	...	1	8
données	# types	type	quantité	...	type	quantité

4.1 Création de la partie

octets	1	1	?	?	?	8	128
données	version	type contrat	taxe (billes)	billes J1	billes J2	nonce	signatures

4.2 Lancé de bille

octets	1	1	?	1	1	32	8	64
données	version	type contrat	taxe (billes)	second	Z	contrat ID	nonce	signature

4.3 Claim transaction

octets	1	1	?	32	20	8	64
données	version	type contrat	taxe (billes)	contrat ID	adresse	nonce	signature

4.4 Échange (double signature)

octets	1	1	?	?	?	8	128
données	version	type contrat	taxe (billes)	billes J1	billes J2	nonce	signatures

5 Jeu

5.1 Menus

Lors du lancement du jeu, un menu principal apparait. Des boutons apparaissent :
- Parties en cours
- Nouvelle partie
- Collection

5.1.1 Parties en cours

Une nouvelle page s'affiche avec les parties en cours de l'utilisateur (sa clé publique). Le joueur peut voir le nombre de coups déjà joués, les billes mises en jeu, la clé publique de son adversaire et l'état actuel de la partie.

Si c'est au joueur de jouer, il peut alors cliquer sur un bouton pour continuer la partie. Le terrain de la partie se génère et il peut lancer sa bille.

5.1.2 Nouvelle partie

Lorsque le joueur clique sur le bouton Nouvelle partie, une nouvelle page s'ouvre avec :
- La clé publique de l'adversaire
- Une boîte de séléction de la/les billes qu'il veut jouer
- Une boîte de sélection de la/les billes de l'adversaire
- Une boîte de sélection de la/les billes qu'il donne aux mineurs (transaction fee)

Le joueur devra signer ce contrat avec sa clé privée avec une boîte de dialogue lui demandant son mot de passe. Celui-ci devra faire signer ce contrat à l'adversaire. Une fois le contrat signé par les deux parties, ce contrat peut être diffusé sur la blockchain par un des deux joueurs. La partie commence.

5.1.3 Collection

Le joueur pourra voir sa collection de billes en cliquant sur le bouton Collection sur le menu d'accueil. Il pourra également examiner les propriétés physique de chaque billes et leur rareté.

5.2 Gameplay

Les terrains du jeu sont générés en fonction d'un *seed*. Pour le même seed, la répartition des billes et des obstacles seront sera toujours la même le même.

Une physique déterministe sera implémentée. Pour un lancé de bille quelconque, la nouvelle position de la bille doit toujours être la même peu importe la marque du processeur de l'utilisateur, sa rapidité, l'heure, etc. Cela permet aux *nodes* d'obtenir le même résultat peu importe le contexte, et donc d'arbitrer la partie avec une haute précision.

6 Outils et coûts

Nous allons utiliser de nombreux outils durant le développement de ce projet.

6.1 Développement global

Les principaux outils que nous utiliserons seront :
- C#
- Git
- GitHub
- Travis-CI
- LaTeX
- Discord
- Google

6.2 Développement de la blockchain

Nous utiliserons différents éditeurs de code tels que VIM, VS Code, ou Rider.
Nous utiliserons les technologies suivantes pour le développement de la blockchain :
- LevelDB : base de données clé/valeur
- ECDSA : algorithme de signature digitale
- SHA-256 : algorithme de hash

6.3 Développement du jeu

Les logiciels suivants nous aideront durant le développement du jeu :
- Unity
- Adobe Photoshop
- Blender

6.4 Developpement du site web

Les technologies suivantes nous aideront durant le développement du site web :
- Netlify
- React.js
- Sass
- MongoDB / MySQL

— Google Chrome

6.5 Coûts

Élément	Prix	Quantité	Description
Hébergement des serveurs	0.00€	∞	C'est pas décentralisé pour rien
Nom de domaine	0.00€	1	Offert par GitHub Student Pack
Discord	0.00€	-	-
GitHub	0.00€	-	-
Électricité	0.1452€	/ kWh	-

7 Tâches

7.1 Répartition des tâches

Tâche	Aurèle	Léo	Raphaël
Génération clés privées (ECDSA)	second	**principal**	
Création de porte-feuille	**principal**	second	
Mining (Proof of Work)	**principal**	second	
Merkle Tree	**principal**	second	
Récompenses des mineurs	second	**principal**	
Encodage/décodage contrats	**principal**	second	
Signature contrats	second	**principal**	
Command Line Interface (CLI)	**principal**		second
Module de persistance (LevelDB)	**principal**	second	
Communication tcp/ip	second	**principal**	
Memory Pool	second	**principal**	
Application des règles du jeu (consensus)	second		**principal**
Définition des 256 billes (nom, atoûts, rareté)		second	**principal**
Design des billes	second		**principal**
Menus du jeu		second	**principal**
Interface de selection de billes	**principal**		second
Modélisation 3D des obstacles	second		**principal**
Création du plateau de jeu (avec seed)		second	**principal**
Lancement de la bille	**principal**		second
Physique deterministe (Unity DPhysics)	**principal**	second	
Détection collision de victoire	**principal**	second	
Site web	**principal**		

7.2 Objectifs

Tâche	1ère soutenance	2ème soutenance	3ème soutenance
Génération d'un porte-feuille fonctionnel	**100%**	/	/
Encodage / Décodage des contrats	**100%**	/	/
Signature contrats	**100%**	/	/
Mining	33%	66%	**100%**
Memory Pool	50%	**100%**	/
Récompenses	0%	50%	**100%**
CLI	33%	66%	**100%**
Module de persistance	50%	**100%**	/
Communication TCP/IP	50%	80%	**100%**
Définition des billes	33%	66%	**100%**
Design des billes	33%	66%	**100%**
Menus du jeu	50%	**100%**	/
Modélisation 3D	0%	50%	**100%**
Terrain de jeu	33%	66%	**100%**
Lancés et trajectoires	33%	66%	**100%**
Site web	0%	50%	**100%**

8 Conclusion

BilleGateCoin est un projet ambitieux et un défi pour tous les membres de **Scattered Corp**. Malgré l'aspect simple du jeu de billes, le but de ce projet est de créer un jeu vidéo entièrement transparent afin de complètement réduire le niveau de confiance que les joueurs doivent donner aux développeurs de la plupart des jeux vidéos d'aujourd'hui.