INTERACTIVE GAME DESIGN BY SMART CONTRACTS
Supervised by John Yuen

LI Kin Yeung
UID: 3035326013
Table of Contents

1 INTRODUCTION .................................................................................................................. 3
2 BACKGROUND ....................................................................................................................... 3
3 OBJECTIVES .......................................................................................................................... 6
4 RELATED WORKS .................................................................................................................. 7
4.1 BLOCKCHAIN GAMES ...................................................................................................... 7
4.1.1 CRYPTOKITTIES ......................................................................................................... 7
4.1.2 FISHBANK .................................................................................................................... 7
4.1.3 TOTAL POKER .............................................................................................................. 8
4.2 LAYER 2 SOLUTIONS ....................................................................................................... 8
4.2.1 LIGHTNING NETWORK ............................................................................................... 8
4.2.2 PLASMA NETWORK ..................................................................................................... 9
4.2.3 LOOM NETWORK ......................................................................................................... 9
5 SCOPE .................................................................................................................................. 10
5.1 DELIVERABLES ................................................................................................................. 10
5.2 JUSTIFICATIONS ............................................................................................................... 10
6 METHODOLOGIES ............................................................................................................. 11
6.1 DESIGN ............................................................................................................................. 11
6.1.1 UI DESIGN .................................................................................................................. 11
6.1.2 SECURITY DESIGN ..................................................................................................... 12
6.1.3 GAME DESIGN ............................................................................................................ 13
6.1.4 GRAPHICS DESIGN ................................................................................................... 13
6.2 IMPLEMENTATION .......................................................................................................... 14
6.3 TESTING ............................................................................................................................ 14
7 RISK AND MITIGATION ..................................................................................................... 15
7.1 TECHNICAL RISK .......................................................................................................... 15
7.2 SCHEDULE RISK ............................................................................................................. 15
8 PROJECT MANAGEMENT .................................................................................................... 15
9 PROJECT SCHEDULE AND MILESTONES ........................................................................ 16
10 REFERENCES ....................................................................................................................... 17
1 Introduction

Due to the rise of new technology, the game world has been revolutionizing in recent years, giving players a brand new experience and interaction throughout the gameplay. Some games utilize blockchain technology to build a secure trading system which allows players to trade their unique assets. However, public blockchain suffers from latency problem, each transaction takes a long time for validation which is undesirable for gaming experience. In light of this phenomenon, some layer 2 solutions have been developed on top of blockchain to support swift transaction and scalable network. The project focuses on the realization of layer 2 solutions by building an interactive game using smart contracts.

2 Background

In the history of game development, the gaming experience is changing tremendously by integrating with new elements and technologies, blockchain is no exception.

Blockchain is a distributed ledger technology that allows each stakeholder in the blockchain network holds a record of the transaction history. As for public blockchain, each transaction is validated via a consensus algorithm like proof-of-work used by Bitcoin. Proof-of-work requires mining, the use of enormous computational power to validate trust-less transactions. Each block in the blockchain corresponds to a transaction which is cryptographically hashed with a mathematical puzzle. People who try to solve the puzzle are called miners. The miner gets some rewards like cryptocurrencies which are newly generated once the puzzle is solved. Then, the transaction is said to be validated and the block is attached to the current blockchain.
As shown in Fig. 1 above, each block contains the hash of the previous block to avoid block violation. Therefore, the transaction is immutable and the entire network is tamper resistant. In general, the more the computation power, the higher the possibility of getting the rewards. Thus, the concept of mining pools is quickly recognized where companies gather large numbers of mining machines and share the rewards equally. However, the complexity of the puzzle depends on the number of users and network load. This ensures a moderate speed of transaction validation and cryptocurrency generation.

Besides greater transparency of the blockchain network, one of the famous applications is tokenization. Blockchain technology has been gaining currency these days due to the invention of Bitcoin, which is the first cryptocurrency launched in the world. The founder of Bitcoin, Satoshi Nakamoto proposed a peer-to-peer electronic cash system, promoting anonymous transfer of values securely.

Apart from Bitcoin, another blockchain platform called Ethereum focuses on decentralized applications that run smart contracts. Smart contracts are sets of well-defined functions and instructions deployed in the blockchain. Every participant in the blockchain network can perform some tasks like initiating a transaction via making calls to smart contracts. Ether, which is the cryptocurrency in Ethereum is used in those transactions. Each transaction costs gas which is the price for running the smart contracts. They are self-executing once deployed in
the network, thus a good smart contract design is of paramount importance in blockchain applications, preventing vulnerabilities and attacks.

The beauty of blockchain and cryptocurrency realizes transparent value transfer and asset tokenization, facilitating the new era of game development. Blockchain game is not a new concept. The first blockchain game called HunterCoin was launched in 2014, where players compete with each other and collect coins back to respawn point [2]. It turned a traditional mining concept into gaming experience, players can move around, combat with other players, as well as collecting resources. Many blockchain games are currently being developed, stimulating the integration of blockchain and game.

However, public blockchain has latency problem which hinders scaling and flexibility in game development. Some layer 2 solutions are built on top of layer 1 root blockchain like bitcoin and Ethereum, to tackle latency and scalability problems. It allows blockchain games with layer 2 solutions to have a faster transaction throughput, improving the gaming experience. Players can interact with other players as usual, with higher security and flexibility throughout the gameplay.
3 Objectives

The goal of this project is to apply blockchain technology in game design, and evaluate existing layer 2 solutions to solve the latency problem of public blockchain. Besides, security is also a main focus in blockchain game, preventing fraudulent transactions and exploitations which destroy the game economy.

From the traditional view of public blockchain, transactions are validated via a consensus algorithm called proof-of-work, requiring huge power consumption. In order to improve the transaction speed of peer-to-peer network, layer 2 solutions are introduced to provide a secure, scalable infrastructure on top of the root blockchain layer. The project aims to utilize suitable layer 2 solutions for game development to reduce transaction latency.

Besides latency issues, the project aims to safeguard in-game transaction during the gameplay. For every turn in the gameplay, identification of real players is crucial to avoid perpetrators to take advantage of it.

Moreover, a good game design caters for all audiences. Striking a balance between veteran players and new players is crucial, especially for games that involve scarce and powerful items. For a game to be sustainable, it shall be free to play without superfluous advantages offered to the paid players. The project aims to ensure fairness of the gameplay, regardless of nature of the players.
4 Related Works

4.1 Blockchain Games

There are different types of blockchain games in the current market, focusing on different features and values.

4.1.1 CryptoKitties

CryptoKitties is a crypto-collectible game where people can buy virtual cats using cryptocurrencies. There are infinite combinations of cats since they are generated by non-fungible tokens. Each cat is unique and is tied to a token with specialized appearance, pattern and accessories. Players can trade with the others using cryptocurrencies and breed a new cat using two cats they owned.

4.1.2 Fishbank

Fishbank is a multiplayer fish battling game. Players can collect different fishes with various abilities. Besides collecting fishes, they can battle with other players and hunt other fishes to growth.
their weights. There are also items to protect the fishes from attacks for certain period of time.

4.1.3 Total Poker

Total Poker is a mobile poker game based in blockchain architecture, featuring secure and fair gameplay. Deck operations are decentralized and cards are randomly distributed. Players can create or join a game room anonymously. The latest update of the game features free gameplay to attract more players to join the poker community.

4.2 Layer 2 Solutions

Layer 2 solutions are used to scale a public blockchain network. The base of the blockchain is a layer 1 protocol that relies on itself for security via proof-of-work. With large hash power generated by miners, it prevents fraudulent transactions and double-spending attacks. Layer 2 solutions sacrifice security in exchange for faster transaction throughput, and rely on layer 1 for security. Participants in the network need to periodically check relevant transactions on layer 2. If there are suspicious transactions, participants can request for auditing by giving evidences to the smart contracts residing in layer 1[4]. After processing the request, the system can then punish the cheaters in layer 2 by returning their funds back to the owner in layer 1. Therefore, layer 2 solutions allow faster transaction throughput with a mechanism to safeguard participants from exploitation. The layer 2 solutions listed below are locomotives in different blockchain platforms.

4.2.1 Lightning Network

Lightning is a layer 2 solution for Bitcoin. It supports off-chain instant payments which is secured by blockchain smart contracts. By creating a state channel between two participants, they can transact many times until timeout is reached. The final result after
subsequent transactions will be sent to the blockchain for validation, freeing up some blockchain bandwidth. It can handle large amount of transactions instantly with small transaction fee, facilitating the scalability of Bitcoin transactions.

4.2.2 Plasma Network

Plasma relies on Ethereum for security. It keeps some smart contracts in the network to performs off-chain transactions. It also allows spawning of child-chains which are attached to the root Ethereum blockchain. Each child-chain can also spawn its child-chain, having a tree hierarchy of side chains that periodically communicate with the root blockchain [5]. This is called the Plasma-chain which handles transaction at a faster speed without replicating the transactions across the Ethereum blockchain.

4.2.3 Loom Network

Loom network aims to provide a scalable infrastructure for Ethereum blockchain applications. The Loom Standard Development Kit (SDK) produces DAppChains which are layer 2 blockchains based in Ethereum. The Loom SDK has a large spectrum of support to various environments such as JavaScript, Phaser, Go and Unity. Different from proof-of-work in the root blockchain, pluggable consensus and smart contracts can be applied to DAppChains to optimize network scalability [6]. It specifically supports a consensus called delegated proof-of-stake (DPoS), enabling flexible scaling of online games and social applications. The DAppChains utilizes Plasma-based relays for secure value transfer in the underlying Ethereum platform.
5 Scope

There are many deciding factors in designing a blockchain game. The following sections provide an overview of the game and architectural decision. They are subject to changes during the first stage of the development cycle.

5.1 Deliverables

The proposed blockchain game is a turn-based battling game with collectibles on mobile platform. Each player chooses three characters for a battle with the opponent. There are different types of collectibles, for example, weapons, armours, resources, etc. Each collectible is generated using Ethereum Requests for Comments 721 (ERC721) tokens. These tokens are non-fungible, tied to a unique collectible. Players can buy collectibles in shop, using weapons and armours to strengthen the character’s ability.

Besides the blockchain game, interim and final reports will be completed during different stages of the development cycle, keeping track of the progress and milestones of the project. An exhibition poster will also be available at the end of the project to showcase the project result.

5.2 Justifications

In this project, Ethereum is used as the root blockchain layer rather than Bitcoin. Bitcoin blockchain is token-centric, and the layer 2 solutions for it focus on peer-to-peer instant payment without third party authority. Ethereum focuses on decentralized applications with smart contracts. Besides, the average transaction speed of Ethereum is 6 minutes where as that of Bitcoin is 78 minutes, thus Ethereum is more efficient [7].

As for the selection of layer 2 solution, LOOM network is used in this project. It is tailor-made for blockchain game development as it supports different platforms like Unity and Phaser. Also, it uses DAppChains for in-game transaction without costing any Ether. This is the main reason of using Loom
because it is unacceptable for players to pay a certain amount of fees whenever they make a move in the gameplay.

6 Methodologies

6.1 Design

The design phase starts in early October. It includes user interface (UI) design, security design, game design and graphics design. Due to time limit constraint, some design elements and resources are found in online assets store.

6.1.1 UI Design

The following part highlights some high-level structure in important scenes: main battle scene and inventory.

![Main Battle Scene Diagram]

Fig. 3: The main battle scene

The game is in landscape mode, providing more space for battling. As show in Fig. 3, the main battle scene is divided into three regions, the upper part shows the player information, including username, level and battling information. The middle part shows players’ characters, the red square indicates the arbitrary places for the characters. The lower part shows character status, for example, skills and species.
The inventory is divided into three parts. The lower part indicates a list of characters the player owned. Each grid shows the image of that character. The upper right part shows position and structure of the battle team. Player can drag and drop character to that area to form a battle team. The upper left part lists some character information, such as health points (HP), ability and skills, with an image of the character.

6.1.2 Security Design

The Loom network is secured by the underlying Ethereum network. The game logic is deployed in the DAppChains to support fast and interactive gameplay. The idea of Loom network is that attackers are disincentivized, since they cannot benefit at all by hacking in the DAppChains.

Besides, the Unity SDK developed by Loom network provides identity management. It validates players’ identity in the gameplay. Since the game logic is running on DAppChains, there is no need
for players to digitally sign on every turn which improves the gaming experience.

Apart from security guaranteed by Loom network, smart contract design is vital to prevent fraudulent transactions. Simple methods like input field checking, are used to prevent unexpected behaviors from happening.

6.1.3 Game Design

The game is a turn-based game where players strategically attack the opponents or avoid the characters from being attacked. For each turn, the player can move a character within certain range or attack opponent’s character. Different characters have different ranges of attacks. Each character gain energy from normal attack. Once the energy gauge is full, the character can use special ability or skill to deal huge damage to the opponents. Some special characters or skills brings a status effect to the opponents after the main attack, like paralyzed, poisoned and confused.

6.1.4 Graphics Design

The style of the landscape, background and characters is cartoonish. The proposed theme is animal, with customized and dissected body parts. Some characters are drawn using Blender. Depending on the time allocation, some graphics resources are borrowed from the asset store.
6.2 Implementation

The implementation phase starts in mid-October. The project is divided into 5 sub-tasks, each individual part is self-complete for easier integration. These divisions are subject to changes.

- Design character models, game assets and user interfaces
- Implement ERC721 tokens to distribute game assets
- Design main game logic, battling and in-game transaction
- Implement inventory management system for characters
- Implement in-game asset store and trade system

During the implementation phase of the project, each sub-task may be further divided into smaller sub-tasks for clarity and change management. Finally, a fully functional blockchain game will be implemented.

6.3 Testing

The testing phase will start right after the implementation phase starts. The project is divided into several modules to simplify the testing procedures. Unit testing is crucial to avoid unexpected situation from happening. It is easier to track the problems and bugs by testing each individual module separately.

Besides unit testing, testing of smart contracts is vital to avoid unnecessary behavior. Once the smart contracts are deployed, it is difficult to make changes and is subject to loss and failure. A tool called Truffle is used for smart contract development, testing and deployment.

Besides unit testing, user acceptance test is carried out. During the initial release of the game, it is important to know whether the user interfaces and gaming experience are acceptable for the players. After collecting some feedbacks from beta testers, improvements can be made during the final stage of the project.
Risk and Mitigation

7.1 Technical Risk

The Loom network SDK is newly released several months ago. The implementation may be complex and there are not many blockchain game projects using it. There may be bugs and integration problems since it is still in beta release.

Fortunately, the Loom network provides a platform called DelegateCall. It allows developers to ask questions regarding blockchain development, especially Loom DAppChain development.

7.2 Schedule Risk

Since it is a one-man project, time management is important for project completion and success. The completion time of each module is estimated, and wrong timing estimation may lead to project failure. Also, some complex functionalities may be not identified at an early stage, requiring extra time for research and development.

In order to tackle schedule risk, keeping a checklist of functionalities is important to keep track on the development progress. Also, the project focuses on interactive game design using smart contracts, extra functionalities shall be omitted if time is limited. Moreover, ranking different modules according to substantiality is useful during the early stage of development.

Project Management

The project follows agile development cycle because it allows change management. The proposed modules are developed incrementally to avoid project failure. The Unified Process also suggests phase gates during the entire development process. During the implementation phase, project objectives and methodologies are reviewed to understand the development progress and decide whether each module can be completed on time.
## Project Schedule and Milestones

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Milestones</th>
</tr>
</thead>
<tbody>
<tr>
<td>September – Early October</td>
<td>1. Preliminary research on blockchain game and layer 2 solutions.</td>
</tr>
<tr>
<td></td>
<td>2. Design graphics and motions of characters, assets and user interfaces.</td>
</tr>
<tr>
<td>Early October – Mid-October</td>
<td>1. Test demo game from Loom network to familiarize the architecture and data flow</td>
</tr>
<tr>
<td></td>
<td>2. Implement ERC721 tokens and test in a test network</td>
</tr>
<tr>
<td>Mid-October – Mid-November</td>
<td>1. Design main game logic and battling turns</td>
</tr>
<tr>
<td></td>
<td>2. Refine artifacts of previous design</td>
</tr>
<tr>
<td>Mid-November – Late December</td>
<td>1. Integrate ERC721 tokens with assets</td>
</tr>
<tr>
<td></td>
<td>2. Design in-game transaction</td>
</tr>
<tr>
<td></td>
<td>3. Implement characters</td>
</tr>
<tr>
<td>Late December – Mid-January</td>
<td>1. Prepare some demonstrations for presentation</td>
</tr>
<tr>
<td></td>
<td>2. Design inventory management system for characters</td>
</tr>
<tr>
<td>Mid-January – Mid-February</td>
<td>1. Implement inventory management system for characters</td>
</tr>
<tr>
<td></td>
<td>2. Design in-game asset store and trade system</td>
</tr>
<tr>
<td></td>
<td>3. Integrate inventory management system with characters and assets</td>
</tr>
<tr>
<td>Mid-February – Mid-March</td>
<td>1. Implement in-game asset store and trade system</td>
</tr>
<tr>
<td></td>
<td>2. Test battling components</td>
</tr>
<tr>
<td>Mid-March – Mid-April</td>
<td>1. Final testing on all components</td>
</tr>
<tr>
<td></td>
<td>2. Integrate all components and design game flow</td>
</tr>
<tr>
<td></td>
<td>3. Prepare demo sections for final presentation</td>
</tr>
<tr>
<td></td>
<td>4. Design exhibition poster</td>
</tr>
<tr>
<td>Mid-April – Late May</td>
<td>1. Prepare for project exhibition</td>
</tr>
<tr>
<td></td>
<td>2. Prepare demo sections for final presentation</td>
</tr>
</tbody>
</table>
10 References


