JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 6 No. 2 (2025) 1-14
https://publisher.uthm.edu.my/ojs/index.php/jscdm

# Blockchain-Based Cheat Detection System for Multiplayer Online Games

**Syamsul Erisandy Arief[1], Reyhan Fajar Pamenang[1], Riri Fitri Sari[1]***

[1] *Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, 16424, INDONESIA*

*Corresponding Author: riri@ui.ac.id
DOI: https://doi.org/10.30880/jscdm.2025.06.02.001

**Article Info**

Received: 8 June 2025
Accepted: 15 November 2025
Available online: 18 December 2025

**Keywords**

Competitive gaming, Ethereum Blockchain, Pong game

**Abstract**

The presence of competitive gaming in the video game industry requires a system that could promote fairness in the gameplay aspect. Many players have utilized networking attacks such as Distributed Denial of Service (DDoS) to win a competitive game. This action will enable players to gain an unfair advantage during gameplay. Attempts to cheat using DDoS attacks in competitive gaming created a significant need for a prevention mechanism. To satisfy this, we have designed a cheat detection system by leveraging Godot DotNet capabilities to connect a game client to the Ethereum Blockchain environment via Nethereum Web3 capabilities. Blockchain is used because it can keep records in an untampered state. We tested our system on the classic Pong game by capturing the positional data of all moving gameplay elements and sending them into the blockchain network. The location coordinate of each central gameplay element in the game is stored in the blockchain. The gameplay evaluation shows that 64-bit hex data of gameplay elements' coordinates have been transmitted and stored successfully. The performance evaluation indicates that the game runs at 180 FPS using 6% of the GPU workload and 11% of the CPU workload, resulting in a time difference of under 200 ms for each transaction.

## 1. Introduction

The video game industry is emerging in the digital landscape. As of 2024, there are over a billion active Internet users worldwide, with an average of 84.1% having played video games through the Internet, and the top three countries with the highest-rated video game reach are the Philippines, Indonesia, and Vietnam, respectively [1]. Being connected to the Internet, video games are no exception to cyber-attacks, which could disrupt gameplay and harm the privacy of its users and developers.

Many video game matches and competitions have been held across numerous countries. These games use a server that synchronizes players' data while managing the game environment. The security aspect of these video games is highly imperative to secure players' and servers' data integrity and privacy while promoting in-game fairness [2].

In video games, cheating is considered to be modifying or manipulating digital game files to gain unfair advantages against other players [3]. Common video game cheats exist in different varieties, such as aimbots, wall hacks, and scripting [2] [4]. These types of video game cheats originate from the client's side. Another standard video game cheat originating from the server's side is lag switching, commonly known as Distributed Denial of Service (DDoS) attacks. In contrast to the client's side cheats, the server's side cheats are more impactful towards

altering gameplay for competitive multiplayer online games, as they could disrupt gameplay for more than one gaming session, therefore impacting more players.

In order to keep players from cheating, video game developers would implement anti-cheat measures. However, many anti-cheat systems in the industry have not succeeded in keeping their game from cheating [5] [6] [7]. Additionally, current anti-cheat measures are prone to DDoS attacks, which slow down and disrupt game servers while disabling players from playing their games online. This kind of attack is possible due to the centralized nature of the service provided by the game developers, in which attackers would target specific game servers and conduct their attack. These attacks impact the game's playability and people's interest. Many cases of DDoS cheating attempts have been made in the past, disrupting the game and its fanbase as it renders the game unfair and completely unplayable [5] [6] [7].

Another case of current anti-cheat measures on video games that can be studied is the invasive and intrusive nature of one anti-cheat measure. In this case, an anti-cheat system would run at the kernel level of a computing system, leaving the computer prone to any malicious attack through the anti-cheat system. If the anti-cheat system security measure has a flaw, it could endanger the computer user's privacy and the data stored in the system [8].

To prevent these problems, we have developed a cheat detection system by leveraging new technological leaps in Blockchain Technology. Blockchain Technology's immutability could be significantly beneficial in keeping the integrity of video game data while mitigating DDoS attack disruptions from interrupting the gameplay [9] [10] [11]. Therefore, this research aims to ensure the integrity and immutability of vital gameplay object data values by storing them in a blockchain. Furthermore, adding smart contracts could automate security maintenance [12].

This paper begins by describing the rising cheating problems in online gaming. After that, it represents the fundamental aspects of Blockchain Technology and Multiplayer Online Games, along with some related works done by other researchers in this field. Our proposed approach to this topic will then be presented, and the performance will be compared in detail. Finally, the results of our approach will be analyzed and discussed for possible future work.

## 2. Background

This section describes the fundamentals of Multiplayer Online Games and Blockchain Technology and discusses some of the works done by other researchers related to blockchain implementations in gaming.

### 2.1 Multiplayer Online Game

A Multiplayer Online Game (MOG) is a digital video game that can connect players to the Internet to engage with other players worldwide. A game client, or a video game player system owned by players, establishes a connection using a network, such as the Internet. This client ranges in type from personal computers to company-released video game consoles. The general network model used in the design of MOGs is client-server networking due to its scalability, which is crucial in a large player base MOG [13].

In a client-server network (see Fig.1), game clients establish a connection in a game server somewhere in the world, where game inputs made by the player are sent to t.1he server, creating a centralized environment. In this network model, the quality of service for each gaming session depends on the server's integrity. Therefore, video game developers using this network model would spend more of their resources on securing a high-quality server provider.

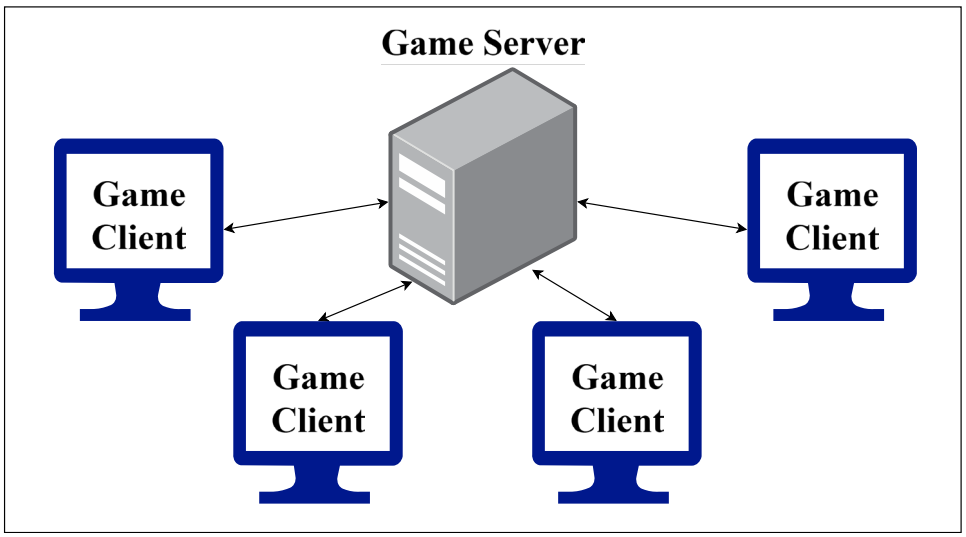



**Fig. 1** *Client-Server networking model*

J. of Soft Computing and Data Mining Vol. 6 No. 2 (2025) p. 1-14

This network model is known to be susceptible to DDoS attacks. A DDoS attack happens when a server is overloaded with information from an external party in a short time. Since game clients need to connect to a server to engage in gameplay, an attack targeted towards a game server would render the game completely unplayable and risk other game clients connected to it [14].

## 2.2 Blockchain Technology

Blockchain Technology is a distributed and decentralized ledger system connected to nodes across the network. Each node shares access to the ledger and its records. The records stored are immutable. Therefore, no one can change or tamper with a record after it enters the ledger. Four types of blockchain networks are available: public, private, permissioned, and consortium. Users could choose which network to use based on their needs [9] [10] [11].

As its name would suggest, records are stored in chained blocks, where each block contains records alongside reference data for the block before it, creating a chain. In blockchain, each record is combined into other records simultaneously, forming a transaction. A transaction could contain records from many clients connected to the blockchain network. These records would be stored in the form of 64-bit hex data. Further data encryption could be done at the developer's request.

The networking architecture used in Blockchain Technology is a Peer-to-Peer (P2P) networking model (See Fig.2) [10]. In this decentralized model, each client connected to the same blockchain network would act as a peer, and each peer would share the same ledger. Since each peer shares the same ledger, any modifications made to the ledger would require verification from each peer, establishing a strong immutability aspect of the ledger.

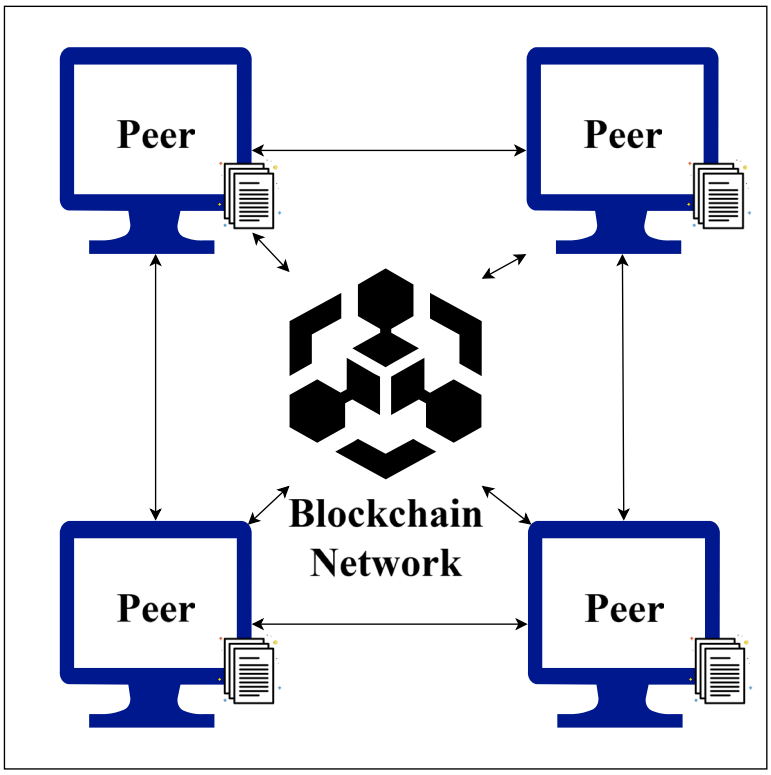



**Fig. 2** *Blockchain technology peer-to-peer architecture model*

One of the key features of Blockchain Technology that distinguishes it from other database models is the implementation of smart contracts. A smart contract is a digital ruleset that predetermines an action once all the rules have been met [10] [12]. In blockchain networks, smart contracts are used to set the rules, terms, and conditions that must be met before a transaction is made valid. This is imperative to the immutability and distributive aspects of the ledger. In addition, being a digital ruleset, smart contracts remove the need for physical contracts and replace an intermediary between two parties.

## 2.3 Related Works

We have found some studies related to the implementation of Blockchain Technology in video games to mitigate cheating [15] [16] [17] [18] [19]. One scientific study has tackled this problem by proposing a decentralized online gaming platform [15]. This study implements blockchain into the game platform itself, creating a Decentralized Application (DApp) for probability games that run on the Ethereum blockchain network. The proposed approach achieved a gaming platform that could run online with a multiplayer aspect that allows players to connect and share tokens with each other to participate in the games. However, we found some key points that could be improved.

One key point of improvement lies in the compatibility aspect of the proposed blockchain implementation with other MOGs. The proposed approach focuses on implementing blockchain for probability games on a single platform without implementing it on other MOGs. Therefore, it would be troublesome if other games were to be implemented on the platform, since it may trigger some compatibility errors that could result in the system being unable to run.

A key improvement point also lies in the choice of tools used to develop the platform. Other tools might give the platform more flexibility and allow for advanced features and functionality. Additionally, developers would not have to compensate for the security of the platform should they elect to add more features for the game in future developments.

Moreover, another scientific study on this topic explored methods of blockchain-based anti-cheat measures [16]. This study uses blockchain technology to monitor gameplay inputs the player provides and detect anomalies. The study found several benefits and drawbacks to using blockchain as an anti-cheat measure. The most prominent benefit is its ability to moderate gameplay based on game state validation, compared to conventional anti-cheat systems that inject themselves into the users' kernel. However, a considerable drawback in this case would be the lack of regulations in the industry to standardize the approach to blockchain-based anti-cheat measures.

An additional study proposed a novel approach to address problems in Peer-to-Peer (P2P) games [17]. This study leverages bright contract-based data packet transfer schemes for game data. The results of this study found that the proposed approach is more effective in terms of latency, scalability, packet loss percentage, performance, and data storage than conventional methods. A key point is that future improvements using advanced learning algorithms may be possible.

A survey on this topic views the perspective of blockchain technology implementation in gaming environments [18]. The aspects analyzed in this study are security advantages, digital identities, emerging trends, and potential research endeavors. In addition, this survey encourages the use of virtual reality (VR), non-fungible tokens (NFT), and other emerging technologies in future works.

## 3. Methodology

This section describes the proposed method for developing the blockchain-integrated tools used in the development process and the overall development process of the blockchain integration.

### 3.1 Proposed Method

The primary purpose of this project is to integrate Blockchain Technology into the game to securely log gameplay telemetries, such as the player's movement data, the ball's movement data, and scores for each player. The players' data consists of Y coordinates to the possible placement for each player in-game. Since the player's position significantly affects the gameplay, it is a central component of the game mechanic. Thus, the slightest modification could completely alter the authenticity of each game. Meanwhile, the ball data consists of X and Y coordinates about the game's playable surface area. In addition, the scores for both players will also be logged using Blockchain Technology to avoid any unwanted modifications.

We implemented this method in a local server environment for this research since this type of environment is also prominently used in the video gaming industry, as many tournaments are held using dedicated servers provided by the tournament holder. Moreover, implementing our method on a public server would require further agreements from server providers, and regulations are yet to be implemented.

In addition, we have also implemented a User Interface (UI) to display gameplay statistics consisting of players and ball positional data and scores for each player. Additionally, this game and console terminal feature will display the number of blocks currently used in the blockchain. The significance of displaying these data in real time is that it refers to players and legitimacy authorities during gameplay.

To connect our game client with a simulated blockchain network, we have run the base game on our system and plan to use several blockchain integration tools that seamlessly connect to the developed game client. The connection will be made possible by lines of code written in the game's program. The connection between the game client and the blockchain network was validated by cross-referencing the data from the game client's UI and the transaction logs found in the blockchain network. Furthermore, our proposed blockchain-based solution can be seen in Fig 3.

J. of Soft Computing and Data Mining Vol. 6 No. 2 (2025) p. 1-14

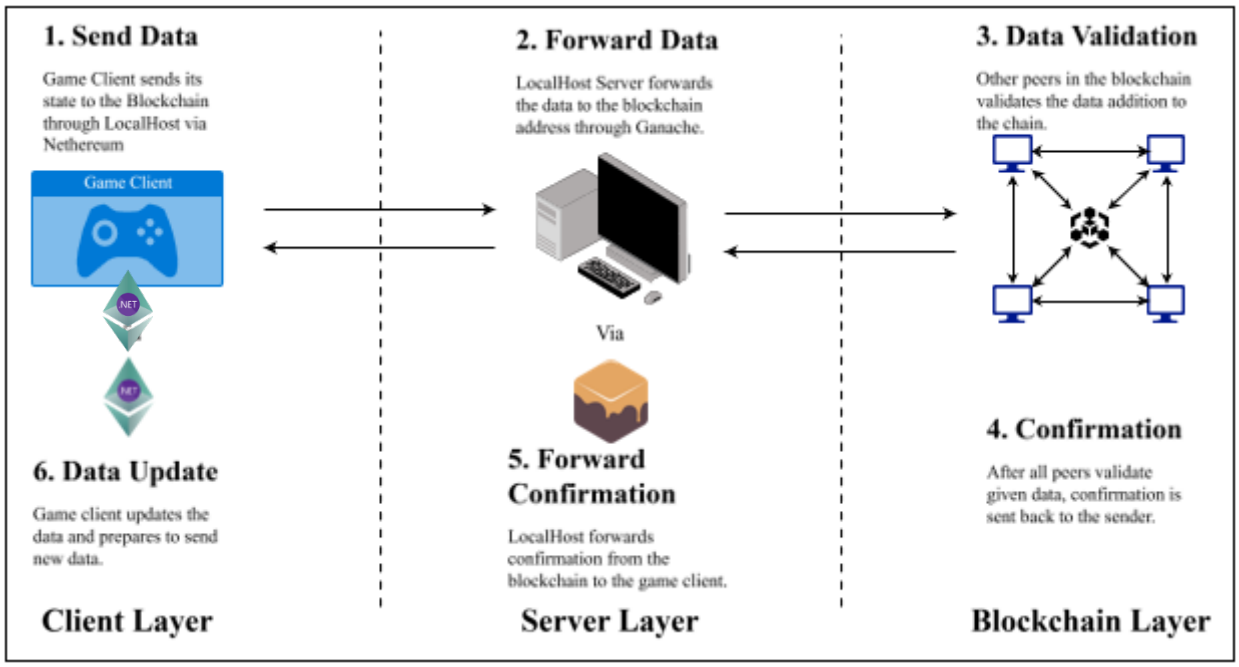



**Fig. 3** *Proposed blockchain-based solution*

## 3.2 Development Tools

We used several tools to create this system. The first tool was the gaming engine itself. We chose Godot Game Engine with DotNet (.NET) integration for this project. Godot is an open-source game engine software that allows users to create free games with no strings attached and no royalties. The users' games are theirs, down to the last line of programming code. Godot's development is fully independent and community-driven, empowering developers to help shape their game to match their expectations. Godot Engine is a feature-packed, cross-platform game engine to create 2D and 3D games from a unified interface. It provides a comprehensive set of standard tools so developers can focus on making games without reinventing the wheel. Games can be exported with one click to some platforms, including the major desktop platforms (Linux, macOS, Windows), mobile platforms (Android, iOS), as well as Web-based platforms and gaming consoles [20].

The use of Godot version 3.4.4 .NET enabled the possibility of C# scripting and Web3 integration into the game engine itself. This could help game developers create a video game that allows the game client to connect to various blockchain networks and perform data sending and receiving. Godot was set as the main game engine in this project to develop the Ping Pong game with Ethereum Blockchain integration for the cheat detection system. Ultimately, the .NET integration, Godot's simple workflow, and its powerful yet open-source engine became the reasons why we chose Godot as the game engine for this research.

The blockchain development tool used is an Ethereum Blockchain connection's programming library called Nethereum. Nethereum is a .NET integration library for Ethereum, simplifying innovative contract management and interaction with Ethereum nodes, which is suitable because the Godot engine uses the .NET framework. Nethereum provides the library to implement Web3 services inside a Godot .NET engine-made game [21]. In this project, Nethereum allows blockchain integration on the game engine side, enabling the game developer to send variables into the blockchain network and customize variables that would be sent into the blockchain network. Furthermore, using Ethereum over other blockchain networks offers better scalability, flexibility, and diversity while supporting the connection of a blockchain network to the game itself [22].

We also used Ganache, a personal blockchain for Ethereum development that can be used to deploy contracts, develop applications, and run tests. It is part of the Truffle Suite, a popular set of tools used in Ethereum development. Ganache provides a local blockchain instance that mimics the Ethereum network, allowing developers to test their smart contracts and applications in a controlled environment without using the live Ethereum network [23]. In this project, Ganache is used to simulate an Ethereum blockchain network that would be used to store the variables sent from the game client.

The last tool that we used was Remix IDE. Remix IDE is an Integrated Development Environment (IDE) for Ethereum's innovative contract development. It is a powerful IDE that provides a comprehensive set of features for writing, testing, debugging, and deploying smart contracts written in Solidity, the primary programming

language for Ethereum [24]. This project uses Remix to create and monitor smart contracts alongside the game engine.

### 3.3 Game Development Process

The game development process spread across the game engine, the smart contract monitoring software, and the blockchain network simulator. However, most of the developments were focused on the game engine. To visualize the development process, Fig .4 shows a flow diagram that elaborates the step-by-step development process of the game.

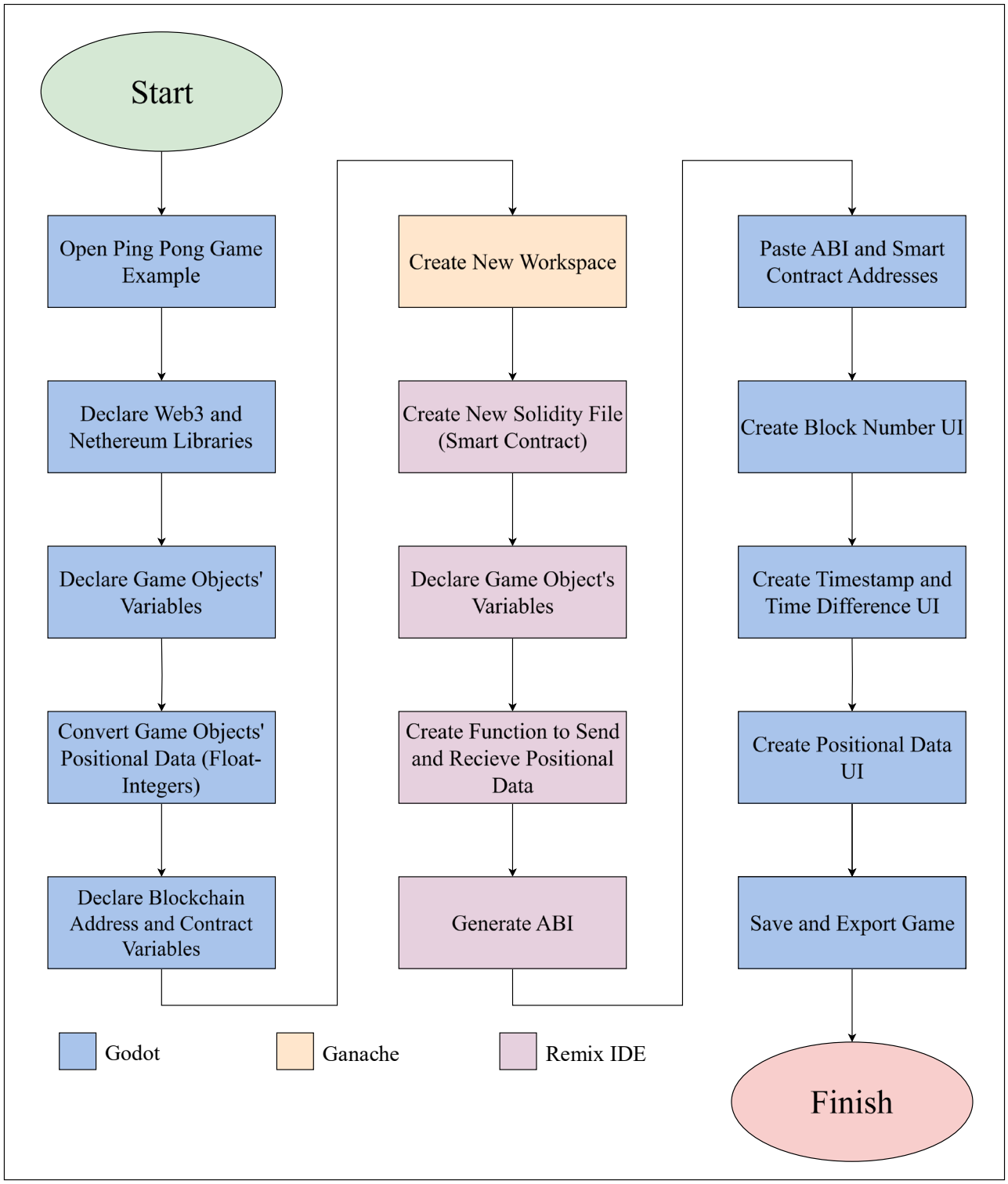



**Fig. 4** *Blockchain integrated game development process flow diagram*

J. of Soft Computing and Data Mining Vol. 6 No. 2 (2025) p. 1-14

As seen on Fig. 4, the game development process began with the Ping Pong game assets the game engine provides as a programming example [25]. Three fundamental objects in the Ping Pong game support major gameplay elements: two player-controlled paddles and a moving ball. The two paddles move at a constant speed on a single axis, the y-axis that moves the paddle up and down. Meanwhile, the ball itself moves on two axes: the x-axis, which moves the ball sideways, and the y-axis, which moves the ball up and down. Other than these fundamental game objects, other game objects support the gameplay aspect of the game, such as the walls and the ceiling. These two objects are static and do not need to be tracked.

After the game objects have been initialized, development of the newly created programming code for this project started. As mentioned before, there are three fundamental game objects. Each fundamental game object was declared in the programming code. Then, each fundamental game object's positional data was extracted from its properties. Since the positional data came in a float format, including decimals, a conversion process was added to convert the data format from float to whole integers. After the conversion process, the data gathered from the conversion process was set as the content of the text displayed on the game client as the game is played.

The blockchain development process began with the creation of a new workspace in Ganache. The workspace was named, then the creation of the workspace was done (with all parameter values set to default). Then, 10 accounts appeared with a unique address, index, and private key.

The innovative contract development process began by creating a file inside a workspace for smart contracts in the IDE. The file was created using the default workspace. After that, we filled in the function to store and retrieve data to make a smart contract. In this case, the data storage function was implemented as part of best practice. The three fundamental game objects discussed earlier got initialized here, alongside variables for player scores, resulting in the initialization of 6 variables. After that, the contract was deployed.

The blockchain integration process occurred in the game engine. This integration process sent positional data from each fundamental game object into the blockchain network. The first code snippet for the blockchain integration process was the code to connect to the blockchain. This code snippet connected the game client to the blockchain network.

A label entity in the program sent the block number variable in order to be displayed in the game client to indicate that the client had successfully connected to the blockchain network. After this, the smart contract's Application Binary Interface (ABI) was declared on the programming code. Then, the address for the smart contract alongside the function to store the fundamental object's position in the smart contract was proclaimed on the programming code. After this, the code to retrieve positional data from each fundamental game object got declared again, and then the transaction process code block to send the data into the blockchain was declared. This code block contained codes for gas limit declaration, the smart contract address declaration, and the transaction function itself.

In addition, we added some lines of code that write the block number, each of the fundamental game object's positional data, a timestamp, and the time difference for each block into the console terminal. The purpose of this code is to check whether the data sent into the smart contract is identical to the data processed in the game itself and ensure the game's performance when sending data to the blockchain.

To visualize the game's operational workflow in this research, Fig.5 shows the sequence diagram of the operational workflow of the blockchain technology implementation within the developed game client.

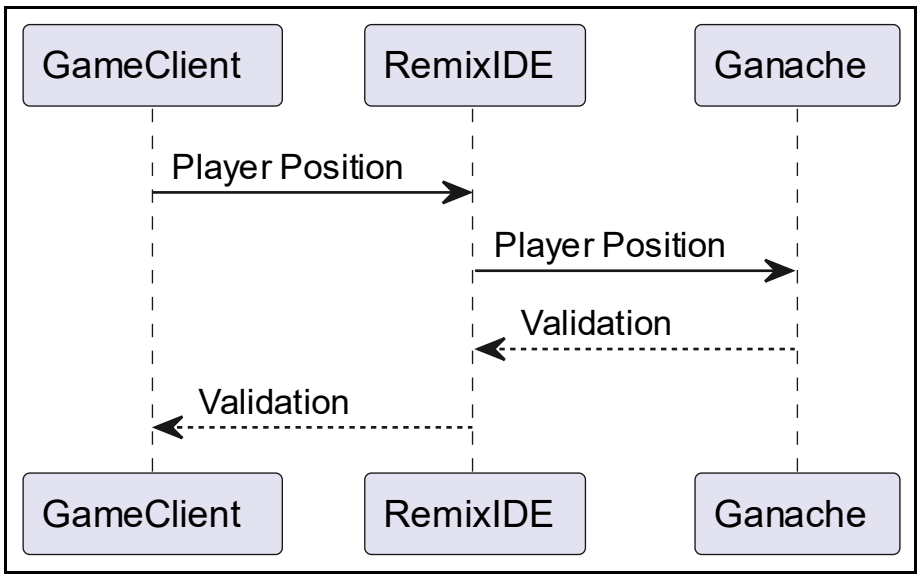


**Fig. 5** *Blockchain integrated game operational workflow sequence diagram*

## 4. Results and Discussion

This section describes the overview, results, and comparison of the tests conducted in this research.

### 4.1 Testing Overview

Two tests were conducted to see if the game and the blockchain network have a good interconnection, send or acquire the proper parameters, and impact the gaming experience overall. The first test conducted was a behavioral test aimed at checking the fundamental features of the blockchain connection to the game. The objective of this test is to satisfy an evaluation question following a feature. If one of the questions is not satisfied, the test will be suspended. The features and their corresponding evaluation question set for this test could be seen in Table 1.

**Table 1** *Behavioral testing features*

| No. | Feature | Description | Evaluation Question |
|---|---|---|---|
| 1. | Stability | The game sends data from the game to the blockchain every second. | Can the game client send data to the blockchain every second? |
| 2. | Integrity | The data sent and stored within the blockchain network matched with the game shown. | Are the data shown in the game consistent with the data in the blockchain network? |
| 3. | Display | The data in the game are shown correctly. | Can users see the positional data on the screen while playing the game? |

Secondly, a performance test was conducted to evaluate the gameplay performance by adding the implemented blockchain data ledger system and how it affects the data-sending process to the blockchain. This test used three metrics of evaluation that were computed during the test. These metrics are elaborated in Table 2. Moreover, the test was executed on a computer system with the device specification listed in Table 3.

**Table 2** *Performance testing metrics of evaluation*

| Performance Metric | Description |
|---|---|
| Frame Per Second (FPS) | The number of images shown on the computer's display monitor for a single second. Higher is better. |
| Graphics Processing Unit (GPU) Usage | The percentage of processing power utilized by the GPU. Higher is better. |
| Central Processing Unit (CPU) Usage | The percentage of processing power utilized by the CPU. Higher is better. |

**Table 3** *Testing device specification*

| Component | Specification |
|---|---|
| Operating System | Windows 11 Pro 64-bit (10.0 Build 22631) |
| Motherboard | Gigabyte B550M GAMING |
| BIOS | F14 (UEFI) |
| CPU | AMD Ryzen 7 5800X 8-Core Processor (16 CPUs), ~3.8 GHz |
| GPU | AMD Radeon RX 6800 XT (16 GB VRAM) |
| RAM | 32 GB |

After conducting both tests, a performance comparison was conducted by playing both the original game without the added blockchain implementation and the modified game with the blockchain implementation for about a minute, and both games were compared in terms of their performances. In addition, the AMD Radeon

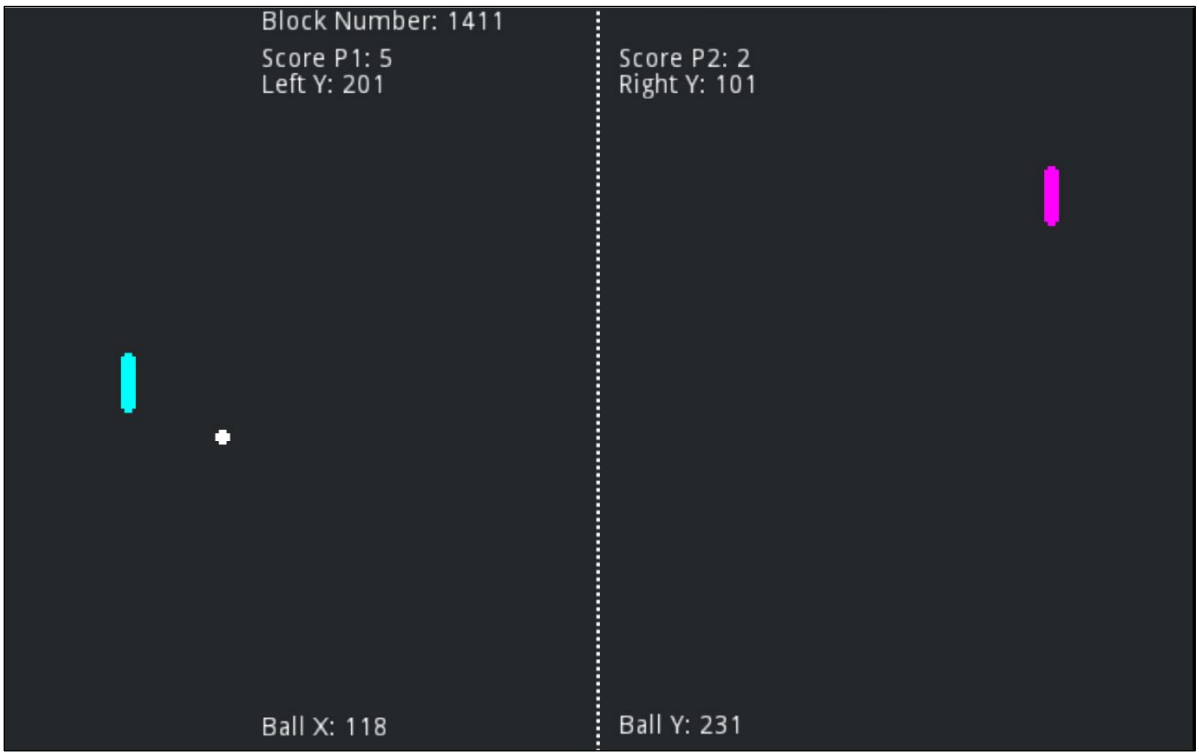
J. of Soft Computing and Data Mining Vol. 6 No. 2 (2025) p. 1-14

Software was used as the performance monitoring software for this test. Both performances were compared and evaluated after the test has concluded.

## 4.2 Behavioral Test

The behavioral test process started by running the simulated Ethereum network. Once the blockchain is up and running, the Pong game was started through the game engine after the correct address of the simulated blockchain has been stated. After that, the Pong game was played for a minute, resulting in 60 blocks of transactional data, while checking whether the positional data are displayed accordingly in the game's display. After a minute, the data produced in the blockchain could then be checked in the simulated blockchain network.

Furthermore, data tracing was done through Remix IDE. This was done by leveraging the retrieve function from the developed smart contract. The data was displayed in the order determined during the implementation process.




**(a)**

```
Transaction Hash: 0x6567acd728b55a297b08abb25b002f5e8ed24c008136156ba4c55c8adb1a5e3b
Block Number: 1411
201
101
91
207
5
2
Timestamp: 2025-05-12 20:14:45.366
Time difference (rounded): 61 ms
------------------
Transaction Hash: 0x72fc85dc416d61774e64024d5491b1f691f534ba95965a35a843a37d88051d71
Block Number: 1412
201
101
171
278
5
2
Timestamp: 2025-05-12 20:14:46.372
Time difference (rounded): 61 ms
------------------
```

**(b)**

**Fig. 6** *Blockchain integrated game (a) Game window; (b) Console terminal window*

Figure 6 shows the game window (a) and the game's console terminal window (b). As seen on Figure 6, we have successfully implemented the UI and console terminal displays that show correct data for each transaction made during gameplay. Comparing the data from the game's console terminal with the received transaction details in Ganache and Remix proves that the integrity of the data transmitted is untampered.

```
Raw data:
0x577032c3000000000000000000000000000000000000000000000000000000000000000a8000000000000000000000000
0000000000000000000000000000000000000000610000000000000000000000000000000000000000000000000000000000000000
0000015e000000000000000000000000000000000000000000000000000000000000009d000000000000000000000000
0000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000
000001

Header: 0x577032c3
P1 Y Coordinate: 00000000000000000000000000000000000000000000000000000000000000a8 (0a8 = 168)
P2 Y Coordinate: 0000000000000000000000000000000000000000000000000000000000000061 (061 = 97)
Ball X Coordinate: 000000000000000000000000000000000000000000000000000000000000015e (15e = 350)
Ball Y Coordinate: 000000000000000000000000000000000000000000000000000000000000009d (09d = 157)
P1 Score: 0000000000000000000000000000000000000000000000000000000000000001 (001 = 1)
P2 Score: 0000000000000000000000000000000000000000000000000000000000000001 (001 = 1)
```

**Fig. 7** *Transaction data breakdown for sample block 186*

The data breakdown shown in Figure 7 describes that the raw data for a sample block numbered 186 stored in the blockchain consists of a 64-bit hex string of gameplay data compiled by the game client. By taking the header out and separating 64 characters, it could be observed that each gameplay data was stored in the form of hex numbers. Converting those hex numbers into decimals shows their actual value.

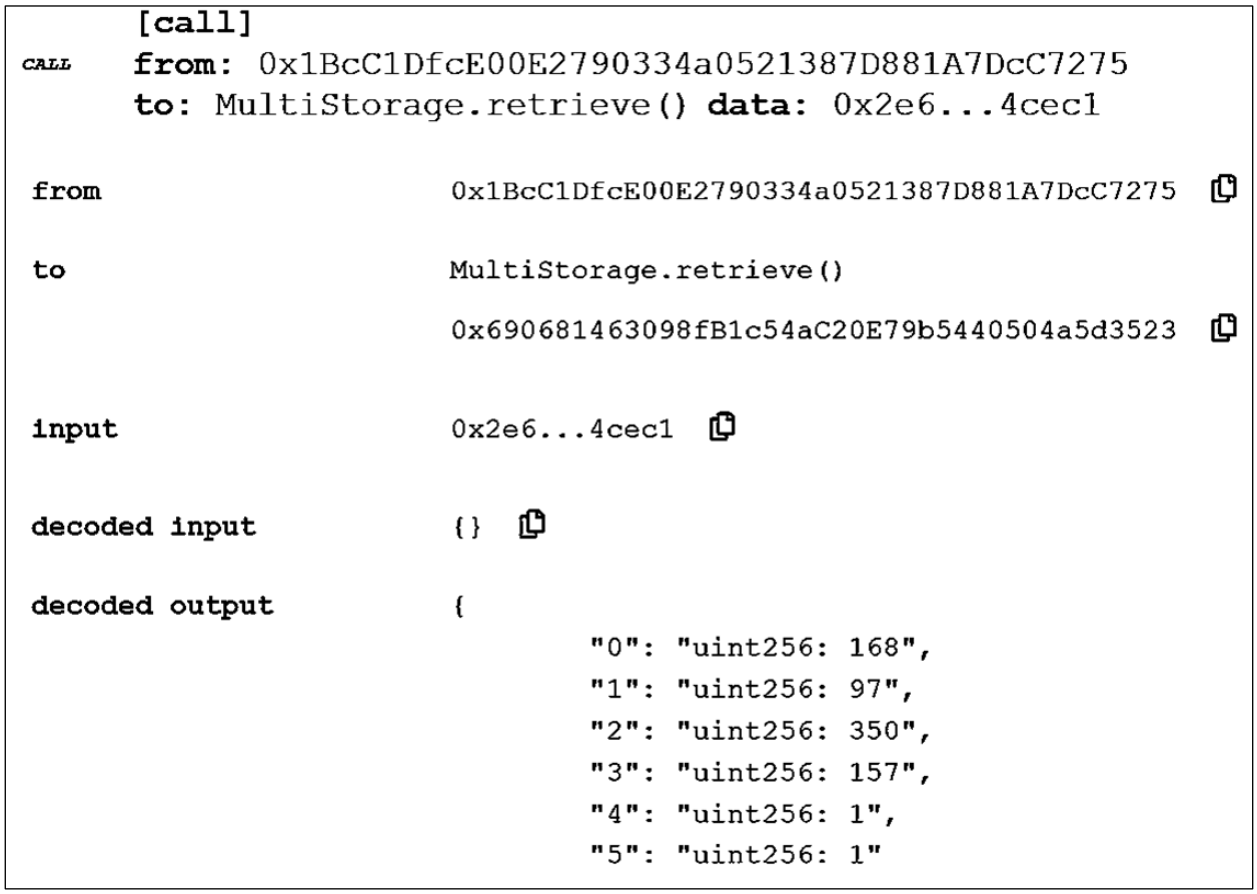



**Fig. 8** *Transaction data in remix IDE for sample block 186*

Meanwhile, Figure 8 shows the gameplay data obtained using Remix IDE. This gameplay data was obtained by using the retrieve function programmed in the smart contract. From 0 to 5 on the decoded output, the data consists of player 1 Y coordinate, player 2 Y coordinate, ball X coordinate, ball Y coordinate, the score for P1, and the score for P2, respectively. Comparing the data shown in Remix IDE to the data conversion shown in Figure 7, it could be seen that the data shown in Remix IDE are identical to the data found in Ganache. Furthermore, the testing results for each feature can be seen in Table 4. As for the security of the scoring data, it can be seen from the gameplay shown in Figure 6 that the positional data for the players' paddle and the ball are stored in each block's transaction data for every second of gameplay. Therefore, these data are considered a record in the blockchain and adhere to the immutability aspect of Blockchain Technology [10] [11] [12].

**Table 4** *Behavioral testing results for blockchain implemented game*

| No. | Feature | Evaluation Question | Result |
|---|---|---|---|
| 1. | Stability | Can the game client send data to the blockchain every second? | Yes |
| 2. | Integrity | Are the data shown in the game consistent with the data in the blockchain network? | Yes |
| 3. | Display | Can users see the positional data on the screen while playing the game? | Yes |

Overall, the behavioral test concludes that the addition of blockchain technology to store gameplay data does not significantly impact the gaming experience. Therefore, it has been proven that blockchain technology could be seamlessly integrated into the game without sacrificing any gameplay aspects. Moreover, the consistency of the gameplay data generated and transmitted from the game client as well as the data stored and viewed from the blockchain network has shown that blockchain technology is capable of securing the immutability aspect of the game data.

## 4.3 Performance Test

The performance test process started by launching the performance monitoring software and selecting Godot as the game. The software will then begin its monitoring process and show the game performance telemetries, such as the Frame Per Second (FPS), GPU usage, GPU memory usage, CPU usage, and other metrics. For this research, we will focus on the metrics detailed in Table 2. Table 5 shows the performance test results for the blockchain-implemented game.

**Table 5** *Performance testing results for blockchain implemented game*

| Metric | Value |
|---|---|
| FPS | 180 FPS |
| GPU Usage | 6% |
| CPU Usage | 11% |

Table 5 shows that the game runs smoothly on the testing device. On average, the game was running at 180 FPS, conveying that the game generates 180 frames during a single second of gameplay. Further inspection of the performance monitoring software indicates that around 6% of the GPU and 11% of the CPU power was used in the gameplay, which could be considered a light workload for the computer system. This stipulates that the game was running stably. Moreover, the time difference from the game's console terminal shown in Figure 6 shows that some time was taken to send the gameplay data to the blockchain. The time difference for each transaction does not seem to exceed 200 milliseconds for each transaction.

## 4.4 Performance Comparison of Original Game to the Blockchain-based Game

The performance of both games was compared by downloading the original game project files from the Godot Asset Library beforehand [25]. The original game was set to be the baseline method of comparison for this research. After the downloading process has been concluded and the project file has been opened, the game was compiled and run with the Godot 3.4.4 .NET game engine. The performance monitoring process was started by playing the game while the performance monitoring software runs on the side of the game window. When the game's performance values have been acquired, the performance metrics would then be compared to the blockchain-based game. Table 6 shows the results of both games' monitored performance.

**Table 6** *Performance testing comparison for original game and blockchain implemented game*

| Metric | Value (Original Game) | Value (Blockchain Implemented Game) |
|---|---|---|
| FPS | 180 FPS | 180 FPS |
| GPU Usage | 9% | 6% |
| CPU Usage | 8% | 11% |

Comparing the performance of the original game and the blockchain-implemented game from the data shown in Table 6, it could be observed that both games run on a similar performance. This indicates that the modification of blockchain technology to the game does not significantly impact the game's performance. Moreover, it could also be stated that implementing blockchain technology in the original game does not alter the overall gaming experience.

The conducted performance test concludes that the game was able to be played without any disruption caused by the game client, the system, or the blockchain network. This has proven that adding blockchain technology to store game data does not alter any major performance changes. Therefore, game developers could leverage blockchain technology to secure their game data from being altered during gameplay without having to worry about any performance related issues.

## 5. Conclusion

This research aims to implement Blockchain Technology as an anti-cheat method in the Ping Pong game. We used Godot as the main gaming engine, Nethereum as the blockchain .NET integration library, Remix IDE as the smart contract manager, and Ganache as the blockchain network simulator. We have successfully connected the game to the blockchain, sent and received positional data from the blockchain, and displayed positional data from the

game client. Additionally, we have evaluated the behavioral and performance aspects of the blockchain implemented game, as well as comparing its performance to the original game.

During this research, we came across several limitations when it comes to implementing blockchain technology into a multiplayer video game. Three main limitations that we thought could be addressed in the future are the scarcity of organized up-to-date documentation and guide on the use of Nethereum in Godot Game Engine, blockchain implementation testing on other multiplayer games, and the lack of transparency in competitive gaming digital environments. These limitations should illustrate the points of improvement if this research is continued.

For future work, the game's GUI and miscellaneous functionalities could be added or modified to suit the player or developer's needs. Furthermore, the technology used in this study may be updated or replaced as developments in blockchain technology arise. Should this technology be implemented, further regulations should be set to ease blockchain implementation in the gaming industry.

## Acknowledgement

This research was funded by Universitas Indonesia under Seed Funding Grant Professor with grant number NKB-2623/UN2.F4.D/PPM.00.00/2023. Furthermore, a grammar-checking tool that utilizes AI was used solely to check and conduct minor grammatical revisions to the writing of this paper, without heavily modifying the contents. Moreover, all authors have reviewed and approved the final content of this paper.

## Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

## Author Contribution

*The authors confirm contribution to the paper as follows:* ***study conception, design, and data collection:*** *Syamsul Erisandy Arief, Reyhan Fajar Pamenang;* ***analysis, interpretation of results, and draft manuscript preparation:*** *Syamsul Erisandy Arief, Riri Fitri Sari. All authors reviewed the results and approved the final version of the manuscript.*

## References

[1] Statista. (n.d.). Global Gaming Penetration Q3 2024, by country. Retrieved January 4, 2025, from https://www.statista.com/statistics/195768/global-gaming-reach-by-country

[2] Szatmáry, K. S. (2024). Cybersecurity of the Gaming Industry. In 2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY) (pp. 000441-000446). IEEE, https://doi:10.1109/sisy62279.2024.10737510.

[3] Consalvo, M. (2008). Cheating: gaining advantage in videogames. Choice Reviews Online, 46(01), 46–0099. https://doi.org/10.5860/choice.46-0099

[4] Adonis, S., & Vadlamudi, S. (2022). Ensuring Privacy and Cyber Safety in the Online Gaming World for Children. In 2022 International Conference on Cyber Resilience (ICCR) (pp. 1-6). IEEE., https://doi:10.1109/iccr56254.2022.10024690.

[5] Hatmaker, T. (2022, October 5). Overwatch 2 launch marred by ongoing DDoS attacks. TechCrunch. https://techcrunch.com/2022/10/05/overwatch-2-launch-ddos-attacks

[6] Eliaçık, E. (2024, January 29). Tekken 8 DDoS attacks prevent players from getting a clean start - Dataconomy. Dataconomy. https://dataconomy.com/2024/01/29/tekken-8-ddos-attacks

[7] Middler, J. (2021, October 19). 'GTA Online' Twitch streamers are being targeted with DDOS attacks. NME. https://www.nme.com/news/gaming-news/gta-online-twitch-streamers-%20are-being-targeted-with-ddos-attacks-3073077

[8] Park, S., Ahmad, A., & Lee, B. (2020, October). Blackmirror: Preventing wallhacks in 3d online fps games. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (pp. 987-1000), https://doi:10.1145/3372297.3417890.

[9] V. Rishiwal, U. Agarwal, M. Yadav, A. Alotaibi, P. Yadav, and S. Tanwar, "Blockchain-Secure Gaming Environments: A Comprehensive survey," IEEE Access, vol. 12, pp. 183466–183488, Jan. 2024, https://doi:10.1109/access.2024.3510467

[10] Elrom, E. (2019). The Blockchain Developer: A practical guide for designing, implementing, publishing, testing, and securing distributed blockchain-based projects. https://link.springer.com/content/pdf/10.1007/978-1-4842-4847-8.pdf

[11] D. Vujicic, D. Jagodic, and S. Randic, "Blockchain technology, bitcoin, and Ethereum: A brief overview," 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), pp. 1–6, Mar. 2018, https://doi:10.1109/infoteh.2018.8345547.

[12] Upadhyay, K., Dantu, R., He, Y., Salau, A., & Badruddoja, S. (2021, December). Paradigm shift from paper contracts to smart contracts. In 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) (pp. 261-268). IEEE, https://doi:10.1109/tpsisa52974.2021.00029.

[13] J. Oster and C. DeLozier, "Preventing Client-Side Exploits in Games with Capability Architectures," 2025 IEEE/ACM 9th International Workshop on Games and Software Engineering (GAS), pp. 41–42, Apr. 2025, https://doi:10.1109/gas66647.2025.00011

[14] Chattopadhyay, S., Tiwari, M., Tiwari, T., & Kapila, D. (2023, May). Role of Cyber security in Gaming technology. In 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 446-451). IEEE, https://doi:10.1109/icacite57410.2023.10182589

[15] Alefs, K., Hartl, F., Newman, L., Özdeveci, B., & Uriawan, W. (2022, September). Secure decentralized online gaming with lending functionalities. In 2022 Fourth International Conference on Blockchain Computing and Applications (BCCA) (pp. 27-32). IEEE, https://doi:10.1109/bcca55292.2022.9921994

[16] Krug, Z. (n.d.). Detecting Cheating in Online Multiplayer Video Games. University of Minnesota Morris Computer Science Senior Seminar Fall 2022. https://umm-csci.github.io/senior-seminar/seminars/fall2022/krug.pdf

[17] Patel, N., Shukla, A., Tanwar, S., Kumar, N., & Rodrigues, J. J. P. C. (2021). GiNA: A Blockchain-based Gaming scheme towards Ethereum 2.0. ICC 2022 - IEEE International Conference on Communications, 1–6. https://doi.org/10.1109/icc42927.2021.9500723

[18] Rishiwal, V., Agarwal, U., Yadav, M., Alotaibi, A., Yadav, P., & Tanwar, S. (2024). Blockchain-Secure Gaming Environments: A Comprehensive survey. IEEE Access, 12, 183466–183488. https://doi.org/10.1109/access.2024.3510467

[19] Wu, F., Yuen, H. Y., Chan, H., Leung, V. C. M., & Cai, W. (2022). Facilitating Serverless Match-based Online Games with Novel Blockchain Technologies. ACM Transactions on Internet Technology, 23(1), 1–26. https://doi.org/10.1145/3565884

[20] E. Fransson, J. Hermansson, and Y. Hu, "A Comparison of Performance on WebGPU and WebGL in the Godot Game Engine," 2024 IEEE Gaming, Entertainment, and Media Conference (GEM), pp. 1–6, Jun. 2024, https://doi:10.1109/gem61861.2024.10585437.

[21] D. P. Bauer, Getting Started with Ethereum. 2022. https://doi:10.1007/978-1-4842-8045-4.

[22] S. Ali, B. Robinson, S. Solomon, S. Poudel, A. Sharma, and K. Upadhyay, "Chain Your Loot: Implementing Blockchain into Gaming Loot Box Markets," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pp. 00861–00867, Jan. 2025, https://doi:10.1109/ccwc62904.2025.10903803.

[23] P. Salire, "Blockchain for online video game integrity," 2023. https://doi:10.31979/etd.xf5d-suct.

[24] Y. M. Arif, M. N. Firdaus, and H. Nurhayati, "A Scoring System For Multiplayer Game Base On Blockchain Technology," The 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), pp. 200–205, Apr. 2021, https://doi:10.1109/apwimob51111.2021.9435249.

[25] Pong with GDScript - Godot Asset Library. (n.d.). 2024. https://github.com/godotengine/godot-demo-projects/tree/master/mono/pong