Innovating Board Games-Integrating Blockchain-Based Random Number Generation in Monopoly

Bohan Zhang

Computer Science and Technology, Century College, Beijing University of Posts and Telecommunications, Beijing, 100096, China
wangjieqiong_001@tzc.edu.cn

Abstract. Blockchain technology is rapidly evolving and widely adopted across various domains, owing to its fundamental characteristics of decentralization, transparency, autonomy, and immutability. This study introduces a blockchain-based pseudo-random number generation method applied to the popular game Monopoly. The proposed approach ensures the uniqueness and immutability of each generated random number, thereby safeguarding the integrity of the game data. Furthermore, it guarantees fairness and transparency, allowing players to validate both the game data and the random number generation process, effectively preventing any instances of cheating or manipulation. Through the integration of blockchain technology, the transparency and reliability of both the game data and the random number generation process are significantly enhanced, thereby enhancing the overall gaming experience for players. Additionally, the decentralized transaction feature of this method enables players to conduct transactions and transfer ownership without being constrained by centralized gaming platforms. This facilitates the unrestricted circulation of assets within the game, thereby augmenting the game's playability and enjoyment.

Keywords: Blockchain, Fairness, transparency, Monopoly Game.

1 Introduction

In the contemporary digital landscape, random number generation is pivotal for a variety of sectors, including cryptography, secure communications, gaming, and simulations [1]. Traditional methods of pseudorandom number generation, while widely used, face challenges in verifying the authenticity and randomness of the numbers produced, leaving them susceptible to malicious interference and attacks. Blockchain technology, recognized for its decentralized, transparent, and immutable properties, introduces novel solutions for enhancing trust and security in the random number generation process [2]. The intrinsic attributes of blockchain allow for the traceability of each random number back to its source, ensuring both its uniqueness and immutability. This capability is crucial in preventing falsification and tampering with the generated numbers. Monopoly, a well-known board game, depends on random numbers to simulate dice rolls. However, traditional methods of random number generation are prone to manipulation, which could compromise the fairness and

Y. Wang (ed.), *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, Advances in Computer Science Research 115,
https://doi.org/10.2991/978-94-6463-540-9_25

integrity of the game [3]. This study investigates the integration of blockchain technology into the Monopoly game, with the aim of developing a secure and reliable pseudorandom number generation method that utilizes the difficulty and timestamp attributes of blockchain network blocks. This approach seeks to ensure the fairness and enhance the player experience by safeguarding the game against potential manipulations.

2 Relevant Theories

2.1 Blockchain Technology

Blockchain technology is a revolutionary innovation characterized by its decentralized, transparent, and immutable nature. It operates as a distributed ledger system, ensuring the security and integrity of data across a network of nodes through a consensus mechanism [4]. At its core, blockchain relies on sophisticated linking and encryption algorithms to organize data into blocks. Each block contains a cryptographic hash of the previous block, creating a chain-like structure. This design ensures that once data is recorded on the blockchain, it becomes virtually tamper-proof and irreversible. The decentralized nature of blockchain means that no single entity has control over the network, making it resistant to censorship and manipulation. Transactions and data stored on the blockchain are transparent, allowing for a high level of trust among participants [5]. Additionally, the immutability of blockchain data provides a reliable foundation for various applications, including random number generation. By leveraging the inherent properties of blockchain, such as its cryptographic hashing and consensus mechanisms, random number generation on the blockchain becomes highly secure and verifiable. The transparency and immutability of blockchain data ensure that generated random numbers are tamper-resistant and unbiased, providing a solid foundation for applications requiring randomness, such as gaming, lotteries, and cryptographic protocols. In essence, blockchain technology not only revolutionizes data management but also serves as a robust platform for ensuring the integrity and fairness of random number generation in various domains.

2.2 Pseudorandom Number Generation

Pseudo-random number generation (PRNG) is of paramount importance in computer science for simulating randomness. However, traditional methods encounter challenges related to predictability and security. In the context of blockchain technology, where security and verifiability are critical, the need for robust PRNG approaches becomes even more pronounced [6]. Solutions encompass leveraging blockchain characteristics, such as decentralization and cryptographic hashing, integrating submission-revelation schemes to ensure fairness, employing verifiable random functions (VRFs) to yield cryptographic-grade randomness, and incorporating on-chain random number generation protocols to guarantee transparency and tamper-resistance through consensus algorithms. The comprehensive application of these techniques enhances the

security and reliability of pseudo-random number generation within blockchain systems, thereby safeguarding the integrity and fairness of various operations.

2.3 Difficulty and Timestamps in Blockchain

In blockchain technology, difficulty adjustment and timestamping play a key role in ensuring the security and integrity of the network. First, the difficulty adjustment mechanism plays an important role in the blockchain protocol, which maintains a consistent rate of block generation [7]. This means that the time it takes to find a valid block remains relatively constant, regardless of the miner's computing power. In the case of Bitcoin, the difficulty is adjusted approximately every two weeks to keep the average block time around 10 minutes. The increased difficulty makes it computationally expensive for attackers to modify past transactions, enhancing the security of the network. Second, timestamps help establish the chronological order of transactions in the blockchain. Each block contains a timestamp indicating when it was created, which ensures that transactions are processed in the correct order, thus maintaining the integrity of the ledger. In addition, timestamps help prevent tampering with the order of transactions [8]. Nodes reject blocks with timestamps that differ significantly from the local time to prevent interference from malicious actors. Overall, difficulty adjustment and timestamping complement each other, and together provide a solid guarantee for the security and consistency of the blockchain network, ensuring the reliability and immutability of the distributed ledger system.

2.4 Application in Monopoly Game

The application of the blockchain-based pseudo-random number generation method in the Monopoly game has brought a significant improvement to the fairness and transparency of the game. In traditional Monopoly games, random number generation is often controlled by the game developer or a central server, which can be a risk of cheating or manipulation, resulting in a decrease in player trust in the outcome of the game. By introducing a blockchain-based pseudo-random number generation method, this problem can be effectively solved and players can be provided with a more reliable gaming experience [9]. Applying the blockchain-based pseudo-random number generation method to the Monopoly game can improve the fairness and transparency of the game. Players can verify the random number generation process to ensure that no cheating is happening. With the immutability and traceability of blockchain technology, the integrity of game data is guaranteed, and players can enjoy a more trustworthy gaming experience.

3 Module Design and Operating Procedures

3.1 Modular Design

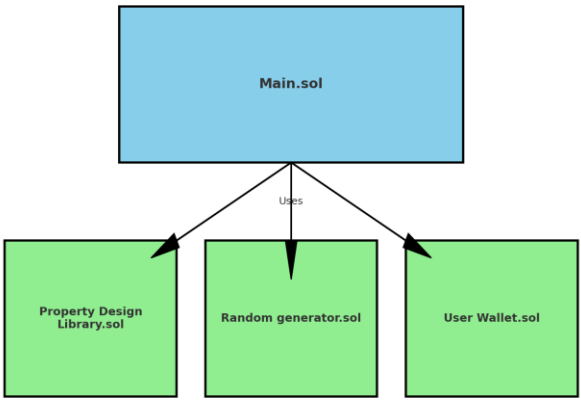

Fig. 1. The module design.

In Main. sol, the core logic of the game is implemented. First, the User_ Init function is used to initialize the player's wallet and provide them with the necessary information, including the initial capital [10]. Players then interact with the game by rolling the dice through the Throw_ Dice function, which determines the number of moves they take on the game board. In this way, they can participate in various transactions, such as paying tolls, buying property, or triggering events. Draw_ Event functions are used to simulate various events that affect a player's financial balance, potentially leading to unexpected payouts or revenues. When a player decides to purchase a property, the property_ buy function facilitates the transaction and the transfer of ownership, ensuring the validity and security of the transaction. In addition, Main. sol includes useful functions such as to String and address To String for converting uint256 and address types to strings, making the user interface more usable and user-friendly. Through the integration of these features, Main. sol implements a complete game logic that enables players to interact with smart contracts. As show in Fig.1.

In ensuring equitable gameplay and mitigating the influence of randomness within a blockchain-enabled Monopoly game, a meticulous approach is introduced. At the core of this strategy lies a prescribed algorithmic formula designed to regulate random outcomes. This formula is instrumental in fostering an environment of fairness and impartiality among participating players：

$$RandomNumber = [SHA-256](timestamp + difficulty) \backslash mod\ N \quad (1)$$

The proposed formula leverages cryptographic techniques, notably employing the SHA-256 hashing algorithm, renowned for its robustness and unpredictability. By integrating this cryptographic mechanism, the inherent randomness intrinsic to the

game's mechanics is effectively harnessed and controlled. Each pivotal game event necessitating a random outcome, such as dice rolls or card selections, is intricately linked to a specific timestamp embedded within the blockchain ledger. This timestamp serves as a unique input parameter for the SHA-256 hashing process, thus ensuring the deterministic generation of seemingly random numbers. To augment the unpredictability of the generated random numbers, a predetermined constant value is systematically incorporated into the calculation. This augmentation serves to enhance the cryptographic strength of the algorithm, thereby fortifying its resistance to potential manipulation or bias. Upon execution, the SHA-256 algorithm meticulously processes the concatenated input parameters, yielding a cryptographically secure hash. This hash is subsequently transformed into a random output, meticulously mapped to the requisite range essential for game mechanics, such as determining movement on the game board or selecting cards from the deck. Through the adoption of this meticulously crafted formula, the Monopoly game achieves an unprecedented level of fairness and transparency. The blockchain infrastructure underpinning the game ensures the integrity and audibility of the random number generation process, instilling confidence and trust among all stakeholders. As show in Fig. 2.

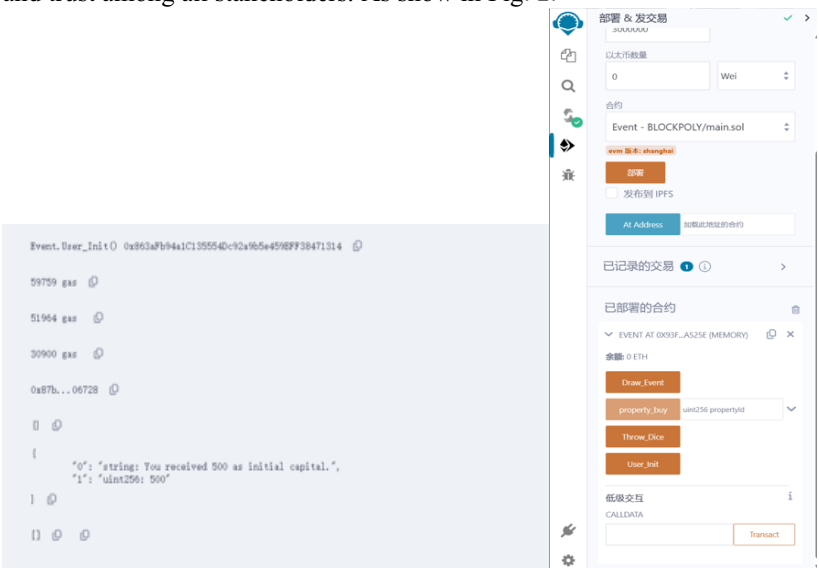

Fig. 2. Deployment and operating environment.

In Property Design. sol, non-fungible tokens (NFTs) representing real estate in the game are managed. First, create_ Property functions are used to assign new property NFTs to specific users. With this feature, each property in the game can be dynamically created and assigned to the player, ensuring the uniqueness and personalization of the property. Next, the transfer Property function is used to facilitate the transfer of ownership of the property. This means that players can freely trade real estate in the game, forming a real estate market. Real estate upgrade is an important feature in the game, and the upgrade Property function can be used to upgrade real estate and increase

the income from tolls. This allows property owners to be more flexible in managing and optimizing their assets. The get User Property function is used to retrieve a list of properties for a specific user, allowing players to know the status of their assets at any time. The is Property Owner Exists function is used to verify the ownership of the property and ensure the legitimacy and security of the transaction. Finally, through the get Data function, players can get detailed property information, including property name, owner, level, etc. The combination of these features makes Property Design. sol a feature-rich and flexible real estate management system that provides players with a rich gaming experience and management options.

The random_ generator. sol contract assumes a pivotal role within the game architecture by overseeing the critical task of random number generation. Its functionality is encapsulated within two key functions: R_ Dice and R_ Event.

The R_ Dice function is tasked with simulating the rolling of dice, a fundamental mechanic essential to the gameplay dynamics. By invoking this function, players trigger the generation of random numbers that effectively emulate the outcomes of dice rolls. This ensures an element of unpredictability and chance within the game, enriching the overall gaming experience and enhancing immersion for players.

As show in Fig.3. In parallel, the R_ Event function serves as the catalyst for a diverse array of in-game events, each imbued with its unique impact on gameplay progression. Whether it's the occurrence of unforeseen challenges, fortuitous opportunities, or unpredictable twists, this function leverages random number generation to inject dynamism and spontaneity into the gameplay narrative. Through its execution, players are confronted with a tapestry of experiences that transcend the mundane, fostering engagement and fostering a sense of exploration within the game world.

PROPERTY	PRICE	TOLL	Toll Increase	Building Expense
1. Tokyo	200	20	10	50
2. Seoul	60	6	5	20
3. Beijing	60	6	5	20
5. Manila	160	16	15	80
7. Hong Kong	200	20	10	100
8. Copenhagen	180	18	15	90
9. Sydney	300	30	20	150
10. Taipei	100	10	5	50
11. Bangkok	100	10	5	50
13. Kuala Lumpur	260	26	15	130
14. Singapore	300	30	20	150
15. Jakarta	180	18	10	90
16. New Delhi	220	22	10	110
18. Dubai	240	24	15	120
19. Istanbul	260	26	15	130
20. Athens	140	14	10	70
21. Rome	160	16	10	80
22. Paris	180	18	15	90
24. London	200	20	15	100
26. Berlin	220	22	15	110
27. Moscow	240	24	20	120
28. Madrid	260	26	20	130
30. Lisbon	120	26	20	130
31. Oslo	140	12	10	60

```
constructor() {
    chanceEvents.push("You've been offered a promotion for your outstanding performance, earning a bonus of
\u00A5200.");
    chanceEvents.push("You unexpectedly received a bonus, gaining \u00A5100 income.");
    chanceEvents.push("You unexpectedly incurred a financial loss, losing \u00A5100.");
    chanceEvents.push("Your diligent work has been recognized by the company, receiving a bonus of
\u00A5150.");
    chanceEvents.push("Your stock investments incurred losses, resulting in a deduction of \u00A5250 from
your funds.");
    chanceEvents.push("Your property has appreciated in value, earning you an income of \u00A5100.");
    chanceEvents.push("Your company's financial downturn led to a deduction of \u00A5150 from your income.");
    chanceEvents.push("Your excellent investment has earned you an additional income of \u00A5200.");
    chanceEvents.push("You made an unsuccessful investment, resulting in a loss of \u00A5400.");
    chanceEvents.push("You've won a competition, earning you a bonus of \u00A5100.");
```

Fig. 3. Board design.

User Wallet. sol manages the functionality related to user wallets within the game system. It includes functions such as deposit, withdraw, get Balance, get Position, and update Position. The deposit function allows users to add funds to their wallet. This function is essential for users to acquire in-game currency or tokens necessary for

various activities within the game. By depositing funds into their wallet, players can participate in transactions, purchase items, or engage in other game-related activities that require currency. Conversely, the withdraw function enables users to withdraw funds from their wallet. This functionality is crucial for players who wish to convert their in-game earnings or assets back into real-world currency or to transfer funds to other accounts. Withdrawal provides flexibility and liquidity to players, allowing them to manage their resources efficiently. To retrieve the balance of a user's wallet, the get Balance function is utilized. This function provides transparency regarding the amount of currency or assets held by the player at any given time. It enables users to monitor their financial status within the game and make informed decisions regarding their expenditures and investments. Apart from managing currency-related operations, User Wallet. sol also includes functions related to the player's position on the game board. The get Position function retrieves the player's current position, which is essential for various game mechanics and interactions. It allows the game system to track the player's progress, trigger events based on their location, and determine their eligibility for certain actions or rewards. Furthermore, the update Position function facilitates player movement on the game board, considering boundary conditions and any restrictions imposed by the game rules. This function ensures that players can navigate the game world seamlessly, avoiding obstacles or constraints while progressing through the game. It enhances the overall gaming experience by providing smooth and intuitive navigation mechanics.

3.2 Operating Procedures

The game board consists of 32 squares strategically designed to offer a diverse and engaging gameplay experience. These squares include the starting point, providing players with a foundational position to begin their journey, as well as 24 property grids representing unique pieces of real estate within the game world. These properties serve as valuable assets that players can acquire, manage, and trade to accumulate wealth and gain advantages over their opponents. Additionally, there are 7 event grids scattered across the board, injecting an element of unpredictability and excitement into the game.

To enhance gameplay dynamics and introduce surprise elements, 10 random events have been pre-defined for players to encounter during their journey. These events are seamlessly integrated into gameplay mechanics by storing them within the array of properties, simplifying the implementation process and ensuring smooth execution.

The game begins with the initial_ user () function call, signaling the commencement of the game session. From there, players take turns rolling the dice by pressing the throw_ dice button to determine their movement across the board. Upon landing on a property grid, players have the opportunity to purchase the corresponding property using the property_ buy function. This decision-making process requires strategic thinking and financial planning, as players must weigh the benefits of property ownership against their available resources.

In contrast, landing on an event grid triggers the draw_ event function, allowing players to draw event cards that can lead to either favorable or unfavorable outcomes. These events simulate real-life scenarios and introduce unpredictability and strategy to

the game. Whether gaining a financial windfall or encountering an unexpected setback, players must adapt their tactics and decision-making to navigate the ever-changing landscape.

Another crucial aspect of gameplay involves interactions between players, particularly when landing on each other's properties. In such instances, players are required to pay a toll fee to the property owner, simulating rent or property taxes. This mechanic incentivizes strategic positioning and property acquisition while fostering competition and negotiation between players.

As the game progresses iteratively, players are rewarded with 100 each time they pass the starting point. This incentive mechanism encourages continued engagement and strategic play as players strive to maximize their wealth and outmaneuver their opponents on the path to victory.

4 Challenges

Blockchain-based Monopoly games represent a pioneering leap in gaming innovation, yet they simultaneously unveil a spectrum of potential flaws and areas ripe for refinement. This section embarks on a meticulous analysis and discussion of these shortcomings, setting the stage for a comprehensive understanding of the landscape.

Foremost among these challenges looms the formidable issue of gas fees. In the intricate ecosystem of the Ethereum network, gas fees stand as tolls exacted for every transaction, exerting a palpable impact on the feasibility of microtransactions within the game. The sheer volume of transactions and data storage operations intrinsic to a Monopoly game exacerbates this concern, rendering it cost-prohibitive for players and impeding their enthusiasm for active participation. To surmount this hurdle, strategic interventions beckon, such as the adoption of sidechain or Layer 2 solutions to mitigate gas fees, or the judicious optimization of smart contract logic to curtail transaction volume and data overhead.

Simultaneously, the assurance of robust security in random number generation emerges as a pivotal imperative. Conventional methodologies for generating random numbers harbor vulnerabilities susceptible to manipulation, thus imperiling the integrity and equitability of gameplay. It becomes imperative, therefore, to embed within smart contracts airtight mechanisms for generating random numbers securely. The exploration of blockchain-driven randomness services or analogous innovations assumes paramount significance, furnishing a guarantee of randomness and equity integral to the game's ethos.

Furthermore, the exigency of enhancing user experience and interface design looms large on the horizon. Presently tethered to the rudimentary confines of Remix IDE for interaction, the game contends with a glaring deficit in user-friendliness. The absence of an intuitive interface and facile avenues for engagement poses a formidable barrier, potentially constricting the game's reach and immersive appeal. A pivotal task thus unfolds: the conception and implementation of a frontend interface that marries security with user-centric design, harmoniously interfacing with smart contracts to deliver an immersive and accessible gameplay experience.

Concomitantly, the fortification of smart contract logic and exception handling emerges as an imperative mandate. While smart contracts embody the quintessence of game logic and data management, the exigencies of anomaly resolution, ranging from player malfeasance to contract security breaches, demand robust mechanisms for resolution. Ensuring the impregnability and reliability of smart contracts crystallizes into a pressing challenge, underscoring the imperative of fortifying the game's foundational framework.

Lastly, the delicate equilibrium of game balance emerges as an arena warranting meticulous scrutiny and optimization. The calculus of property acquisition costs, event dynamics, and player interactions mandates exhaustive testing and recalibration to engender a gameplay experience that is both engaging and equitable.

5 Conclusion

This research culminates not merely in a prototype but stands as a beacon of potential within the evolving domain of blockchain-enhanced gaming. By integrating the timeless appeal of Monopoly with the transformative capabilities of blockchain technology, a prototype has been crafted that extends beyond simple entertainment, offering a preview of the future landscape of interactive gaming. The journey has navigated a series of implementation challenges, each serving as a crucible for innovation and refinement. The persistent issue of gas fees, which threatens the viability of microtransactions and player engagement, prompted the devising of strategies to mitigate these economic burdens. Beyond financial concerns, a myriad of complexities were encountered, from ensuring the integrity of random number generation to enhancing user experience and optimizing the logic of smart contracts, all while maintaining a balance within the game dynamics. Amid these challenges, a wealth of invaluable lessons and insights have been gleaned, shedding light on the path forward for future pioneers in blockchain gaming. The efforts mark not just the conclusion of a project but the beginning of a new era, where the convergence of blockchain and gaming redefines the boundaries of entertainment. Looking ahead, a future free from the constraints of gas fees is envisioned, where security is unassailable, user experiences are limitless, and game dynamics are meticulously tuned.

However, the journey is far from complete. It beckons exploration further, to decipher the complexities of cross-chain interoperability, to harness the potential of NFT asset management, and to venture into uncharted territories that stretch beyond current imagination. It is within these unexplored frontiers that the true essence of innovation resides—not merely in refining what is, but in shaping what could be.

In conclusion, the endeavors stand as a testament to the relentless advance of progress, driving ever closer to realizing a visionary future where blockchain-enabled Monopoly games transcend mere recreation to become exemplars of interactive entertainment. With steadfast dedication and relentless innovation, academia is poised at the dawn of a new era, where blockchain and gaming merge to forge a future replete with endless possibilities.

References

1. Du, M., Chen, Q., Liu, L., & Ma, X.: A Blockchain-based Random Number Generation Algorithm and the Application in Blockchain Games. IEEE International Conference on Systems, Man, and Cybernetics (SMC), Bari, Italy, 3498–3503 (2019).
2. Ehara, Y., & Tada, M.: How to generate transparent random numbers using blockchain. International Symposium on Information Theory and Its Applications (ISITA), Singapore, 169–173 (2018).
3. Bradić, S., Delija, D., Sirovatka, G., & Žagar, M.: Creating own NFT token using erc721 standard and solidity programming language. 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 1053–1056 (2022).
4. Sarathy, R.: Enterprise Strategy for Blockchain: Lessons in Disruption from Fintech, Supply Chains, and Consumer Industries. MIT Press (2022).
5. Makridakis, S., & Christodoulou, K.: Blockchain: Current challenges and future prospects/applications. Future Internet, 11(12), 258 (2019).
6. Zhu, X., Xu, H., Zhao, Z., et al.: An Environmental Intrusion Detection Technology Based on WiFi. Wireless Personal Communications, 119(2), 1425–1436 (2021).
7. Amate, R., Indulkar, S., Pedgoankar, A., Waghmare, L., & Deshpande, S.: Implementation of Blockchain by Gamification. 2023 6th International Conference on Advances in Science and Technology (ICAST), 262–266 (2023, December).
8. Laneve, C., & Ershov, I.: Blockchain gaming: An analysis of the use of blockchain technology in the video gaming industry. (n.d.).
9. Shi, X., Yao, S., & Luo, S.: Innovative platform operations with the use of technologies in the blockchain era. International Journal of Production Research, 61(11), 3651–3669 (2023).
10. Mollah, M. B., Zhao, J., Niyato, D., Lam, K. Y., Zhang, X., Ghias, A. M., et al.: Blockchain for future smart grid: A comprehensive survey. IEEE Internet of Things Journal, 8(1), 18–43 (2020).