Beyond Centralized Casinos: A Transparent and Efficient Protocol for On-Chain Blackjack

Arjun Shetty[1], Brian Zhou[2], Kaden Kram[3], Karen Wu[4], Sandy Zhang[4]

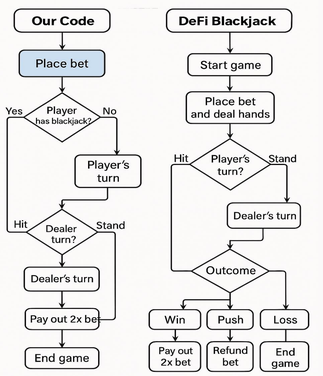
[1] Department of Mechanical Engineering, [2] Department of Electrical and Computer Engineering, [3] Department of Operations Research and Financial Engineering, [4] Department of Computer Science

Abstract

This work presents a decentralized on-chain implementation of blackjack that delivers a transparent and simple gaming experience. Our protocol removes traditional casino intermediaries by leveraging blockchain smart contracts and integrating cryptographic ways to generate a random number to ensure fair card distribution. By implementing a stateless 1v1 (player vs. dealer) model with simplified card mechanics, the system reduces on-chain complexity and gas consumption while maintaining the familiar gameplay structure. Overall, the platform offers real-time interaction, verifiable outcomes, and trustworthy economic arrangements that empower users with both fairness and efficiency.

Related Work

- A key feature of our work is its decentralized nature, which differentiates it from the extensive online gambling platforms that lack the transparency we are trying to achieve
- Decentralized gambling interfaces, specifically for roulette and blackjack, have been implemented with configurable ERC20 tokens
- However, the current work done with a decentralized blackjack model (A Decentralized Blackjack Game on Arbritrum) introduces elements like token burns and Uniswap that make the code vulnerable and complex
- Our work builds on the foundation of this previous work and provides a secure, simple and readable contract to improve the decentralized gambling experience
- A comparison between the game flow of related work is seen below, with our game simplifying on-chain logic for lower gas prices and better error handling



Motivation

Traditional online gambling often suffers from a lack of transparency and centralized control, which can lead to distrust amongst players. Existing decentralized solutions have attempted to address these issues but frequently introduce unnecessary complexity and potential security vulnerabilities—especially when integrating advanced features such as token burns or liquidity protocols. Our project is motivated by the need to create a simplified yet robust blackjack protocol that emphasizes fairness, security, and efficiency. By removing extraneous elements and focusing on a stateless design, we aim to lower risk, reduce gas fees, and provide a straightforward, verifiable gaming experience accessible to players worldwide.

Implementation

Technical Stack Summary

Smart Contract Layer
- Language: Solidity (v.0.8+)
- Testnet: Sepolia Testnet (Ethereum-compatible)

Frontend
- Framework: React.js
- Wallet Integration
- Metamask secured

Security & Reliability
- Minimal surface area
- Minimal external dependencies
- Metamask secured

Development Tools
- Compiler: Remix
- Coding Language: JavaScript
- Deployment: Netlify
- Version Control: Git

- The game is implemented as a 1v1 (Player vs. Dealer) model to avoid logic handling for concurrent sessions and the need for state synchronization
- Our code is publicly available to ensure secure, unbiased card assignment for each user- initiated betting session
- Players can wager a fixed amount (e.g. 100 LUSD), where winning outcomes result in a 2x payout and blackjack hands return a 1.5x profit. Losses are retained by the contract, where an initial bankroll is implemented to minimize early volatility and add liquidity
- A web interface was developed using React, HTML/CSS, and Ethers.js to facilitate wallet connectivity, bet placement, and real-time gameplay visualization

Gameplay

Orange & BlackJack is designed as a solo game where a player competes against a virtual dealer. Upon connecting their Metamask wallet and placing a bet (using LUSD), the smart contract automatically initiates the game by dealing two cards to both the player and dealer—with one of the dealer's cards revealed as per the rules. Cards are assigned values from 1 to 11 (with face cards counting as 10) to mimic traditional blackjack scoring, albeit in a simplified form. The player may then choose to "hit" for an additional card or "stand" to keep their current hand, aiming to reach as near to 21 as possible without busting. After the player concludes their turn, the dealer reveals the hidden card and draws additional cards until reaching at least 17. Standard payout rules apply: victories pay out 2× the original bet, blackjack wins return 1.5×, ties refund the original wager, and losses are retained by the contract. The gameplay logic is executed entirely on-chain, ensuring that every card distribution and outcome is secure, unbiased, and fully auditable via the underlying smart contract.

Leaderboard

Rank	Player	Net Profit
🥇	Sandy	10000.00 LUSD
🥈	The Joker	2680.00 LUSD
🥉	The actual Brian	-99900.00 LUSD

Analysis

- The broad goal of this project was achieved: creating an efficient and trustworthy model of on-chain Blackjack
- Making our betting sessions stateless reduced much of the complexity of the game and improved performance, but this was a trade off as improved on-chain efficiency meant less realism (no suits, multiplayer logic)
- The simple frontend ensured smooth wallet connection and gameplay but investment in a more advanced web platform would be beneficial for improved player engagement
- Limits on the contract's external dependencies as well as transparent code/testnet only usage resulted in a significant reduction in risk compared to traditional online gambling sites

Future Work

DAO & Community Governance
Establish a DAO funded by a treasury with a small house edge. Token holders vote on upgrades and rules

NFT-Based Achievements
Mint NFTs for rare in-game player feats, usable in gameplay or tradable

Real-Economy Deployment
Deploy to Ethereum Mainnet or Layer 2 solutions. Enable real-value wagering

Mobile Optimization
Improve UI for responsiveness and mobile browser support

References

- moses-0x. (2022). *DeFI-BlackJack: A decentralized Blackjack game on Arbitrum* [Computer software]. GitHub. https://github.com/moses-0x/DeFI-BlackJack
- Viswanath, P. (n.d.). *Elements of Decentralized Finance* [Lecture slides]. ECE 437: Decentralized Finance. Princeton University. https://web3.princeton.edu/elements-of-defi/
- jaslevy, HKnoll42, & dsukha. (2024). *Sepolia Roulette: A decentralized casino game on Sepolia* [Computer software]. GitHub. https://github.com/[REPO_PATH]
- Chainlink. (n.d.). *Data Streams*. Chainlink Documentation. https://docs.chain.link/data-streams

Acknowledgements

We want to thank Professor Viswanath for his time and teaching of this course and the lab TAs —Viraj Nadkarni, Mahsa Bastankhah, Niusha Moshrefi— for their help and guidance this semester.