Smart Contract

Security Assessment

For UpDown
22 Oct 2025

Ascendant
@ascendantfi
www.ascendant.finance

Table of Contents

DISCLAIMER

This independent audit has been conducted to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by the auditor.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and the auditor is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will the auditor or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

Auditor retains full rights over all intellectual property (including expertise and new attack or exploit vectors) discovered during the audit process. Auditor is therefore allowed and expected to re-use this knowledge in subsequent audits and to inform existing projects that may have similar vulnerabilities. The auditor may, at its discretion, claim bug bounties from third-parties while doing so.

Executive Summary

The smart contracts reviewed in this audit were found to be **Well Secured**, meaning they contain no critical severity issues that would render them too unsafe to launch. However, it is recommended that the remaining issues found within this report be resolved or mitigated to ensure best user experience.

Security Level	
Well Secured	
Secured	✓
Poorly Secured	
Insecure	

We performed an independent technical audit to identify Smart Contracts uncertainties. This shall protect the code from illegitimate authorization attempts or external & internal threats of any type. This also ensures end-to-end proofing of the contract from frauds. The audit was performed semi-manually. We analyzed the Smart Contracts code line-by-line and used an automation tool to report any suspicious code.

The following tools were used:

- Truffle
- Hardhat
- Remix IDE
- Slither
- Sol2UML

Overview

This report has been prepared for UpDown for the Base Network. This audit provides a user-centered examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

Summary

Project Name	UpDown
Platform	Base
Language	Solidity

Contracts Assessed

Name	Location
PredictionGame.sol	Not Published

Findings Summary

Severity	Found
High	0
Medium	1
Low	2
Informational	0
Total	3

Classification of Issues

High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
Medium	Bugs or issues that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
Informational	Consistency, syntax or style best practices, Generally pose a negligible level of risk, if any.

Manual Review

Issues Checking Status

Issue Description	Checking Status
Compiler errors	PASS
Race conditions and Reentrancy. Cross-function race conditions.	PASS
Possible delays in data delivery.	PASS
Oracle calls.	PASS
Front running.	PASS
Timestamp dependence.	PASS
Integer Overflow and Underflow.	PASS
DoS with Revert.	PASS
DoS with block gas limit.	PASS
Methods execution permissions.	PASS
Economy model of the contract.	PASS
The impact of the exchange rate on the logic.	PASS
Private user data leaks.	PASS
Malicious Event log.	PASS
Scoping and Declarations.	PASS
Uninitialized storage pointers.	PASS

Arithmetic accuracy.	PASS
Design Logic.	PASS
Cross-function race conditions.	PASS
Safe Open Zeppelin contracts implementation and usage.	PASS
Fallback function security.	PASS

Severity	**Low**
Contract	PredictionGame.sol
Description	Code redundancy ln:405, 411, 420
Code Snippet	PredictionGame claim function;
Recommendation	in claim() Function we can do some optimization by caching the round rounds[paymentTokenAddress] [gameTokenAddress][epochs[i]] cause accessing it on the top line 432 go to line 404and we modify line 405 411 420
Status	

Severity	**Low**
Contract	PredictionGame.sol
Description	OPTIMIZATION for statement
Code Snippet	require(epoch == currentEpoch[paymentTokenAddress][gameTokenAddress], "Bet is too early/late"); require(bettable(paymentTokenAddress, gameTokenAddress, epoch), "Round not bettable");
Recommendation	Using require with string is good but using custom errors with it instead of string and revert would be more gas-efficient.
Status	

Severity	**Medium**
Contract	PredictionGame.sol
Description	Function should be nonReentrant

Code Snippet	function _safeTransferBNB(address to, uint256 value) internal {
	(bool success,) = to.call{value: value}("");
	require(success, "TransferHelper: BNB_TRANSFER_FAILED");
	}

Recommendation	_safeTransferBNB Function should be nonReentrantsince that we call it in claimTreasury which is not nonReentrant
Status	

Functional Test Status

Function Name	Type/Return Type	Score
PredictionGame		
betBull	external	PASS
betBear	external	PASS
claim	external	PASS
genesisStartRound	external	PASS
_initializeTokenGenesis	internal	PASS
executeRound	external	PASS
pause	external	PASS
unpause	external	PASS
setMinBetAmount	external	PASS
setTreasuryFee	external	PASS
setOperator	external	PASS
ReetrancyGuard		
_nonReentrantAfter	private	PASS
_nonReentrantBefore	private	PASS
_reentrancyGuardEntered	internal	PASS
Hashes		
_efficientKeccak256	private	PASS
communtativeKeccak256	internal	PASS

MerkleProof		
multiProofVerify	internal	PASS
multiProofVerifyCallData	internal	PASS
processMultiProof	internal	PASS
processProof	internal	PASS
processProofCallData	internal	PASS
verify	internal	PASS
verifyCalldata	internal	PASS
IERC20		
allowance	external	PASS
approve	external	PASS
balanceOf	external	PASS
totalSupply	external	PASS
transfer	external	PASS
transfeFrom	external	PASS

Omitted Results

Note: Any issues that have been omitted from this report have been deemed by the reviewing team as irrelevant, inapplicable, and/or negligible to the proper functioning of this contract. Thus, any omitted issues can be safely ignored.

Automated Review

Conclusion

The smart contracts reviewed in this audit contain no critical severity issues and all Medium to Low issues have either been corrected or acknowledged.

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.

Ascendant

Ascendant

@ascendantfi

www.ascendant.finance

Ascendant