Solana Smart Contract Audit Report

Project: Win-2-Earn FPS Game (Solana) **Date:** September 24, 2025 **Auditor:** Saber

---

Table of Contents

Executive Summary

This audit covers the core Solana smart contracts for a Win-2-Earn FPS game, focusing on player matching, escrow, payouts, and anti-abuse. The codebase is modular, leverages Anchor best practices, and is generally secure. All previously identified issues have been addressed, including arithmetic safety, account ordering, player uniqueness, session ID validation, and additional defensive checks. The codebase now includes comprehensive documentation and robust test coverage for edge cases.

Scope

- Full review of all Rust smart contract code in `programs/wager-program/`
- Identification of vulnerabilities (re-entrancy, privilege escalation, logic flaws, etc.)
- Suggestions for compute/gas optimization
- Test coverage review and new test cases
- Code improvements for critical issues

Findings

Vulnerabilities

Issue	Severity	Details	Recommendation
Unchecked arithmetic	Resolved	All calculations now use checked math.	Continue to use `.checked_*` methods and Anchor's `require!`.
Remaining accounts order	Resolved	Order is now documented and runtime-checked in payout/refund handlers.	Maintain documentation and tests.
Vault Account (SOL)	Low	The vault is a raw `AccountInfo` (for SOL), but not used for SPL transfers.	If not used, remove or clarify its purpose.
Player uniqueness	Resolved	Duplicate player joins are now prevented.	Maintain uniqueness checks.
Session ID collision	Resolved	Session IDs are validated for length and allowed characters.	Maintain validation logic.

Logic Flaws

- State transitions, player uniqueness, and session ID validation are now robustly enforced.
- No critical logic flaws found in escrow, payout, or anti-abuse mechanisms.

Compute/Gas Optimization

- Account data is minimized and arrays are efficiently used.
- Payouts are batched per team; further batching may be considered for very large games.

- Anchor's `require!` is used for error handling.

Severity Ratings

- **Critical:** None found
- **High:** None found
- **Medium:** None (all previously medium issues resolved)
- **Low:** Vault account (if unused)

Recommended Fixes

All major recommendations have been implemented:

1. Checked math for all calculations involving user funds.
2. Player uniqueness checks in join logic.
3. Session ID format and uniqueness validation.
4. Documentation and runtime checks for `remaining_accounts` order in payouts/refunds.
5. Additional defensive checks for zero addresses, duplicate winners/players, and vault balance sufficiency.

Test Coverage & Results

- All core flows (joining, payouts, refunds) are covered by automated tests.
- New edge case tests added:
  - Duplicate player joins
  - Session ID collisions
  - Arithmetic overflow
  - Incorrect account ordering in payouts/refunds
  - Invalid team selection
- All tests pass as of the latest commit, confirming the effectiveness of the implemented protections.

Suggested Improvements

- Continue to maintain and expand test coverage as new features are added.
- Periodically review for new Solana/Anchor best practices.
- Remove or clarify the purpose of the unused vault account if not needed.

Walkthrough Summary

- The codebase is secure, modular, and well-documented.
- All previously identified issues have been addressed.
- No critical vulnerabilities found.
- The contract is ready for production deployment, pending ongoing review as the project evolves.

Prepared by Saber, September 2025