PIXELWRIGHT DIGITAL

Code Audit Triage

A fast read on every urgent risk in your codebase, plus a sample of the backlog.

DeckScore Public Sample Audit

DeckScore public sample – implementation details redacted

Date May 20, 2026

Files scanned 68

Total findings 33 (4 high / 22 medium / 7 low)

Section A — urgent 4 findings (release blockers + before launch)

Section B — backlog sample 10 shown of 15 non-urgent

SECTION 1

Executive Summary

ELEVATED RISK — ACTION REQUIRED

We found 4 high-severity issues that warrant attention before your next release. The top items below are ranked by severity and urgency; address the release blockers first.

33	4	22	7
TOTAL	HIGH	MEDIUM	LOW

Section A — urgent findings

1. **Revenue entitlement state trusted before verification**
   HIGH · Release blocker · — · Location redacted in public sample

2. **Launch-time purchase state could be read before verification completed**
   HIGH · Before launch · — · Location redacted in public sample

3. **Completed-session statistics could be recorded twice across lifecycle paths**
   HIGH · Before launch · — · Location redacted in public sample

4. **Completed-session resume path could re-credit statistics**
   HIGH · Before launch · — · Location redacted in public sample

HIGH · Release blocker

Revenue entitlement state trusted before verification

HIGH RELEASE BLOCKER CANDIDATE

LOCATION Location redacted in public sample

WHAT WE FOUND

The audit found that purchase-state UI could consult unverified local state before the authoritative entitlement check completed. This public sample redacts the exact storage mechanism, keys, file paths, method names, and code snippets; paid client reports include those details.

WHY IT MATTERS

Under certain local-device conditions, the app could temporarily present a paid or unlocked state before verification completed. For a revenue path, that creates monetization leakage and misleading entitlement state.

RECOMMENDED NEXT STEP

Make authoritative entitlement verification the gate before enabling paid-state UI. If a local hint is kept for launch speed, treat it as untrusted until verification completes.

HIGH · Before launch

Launch-time purchase state could be read before verification completed

HIGH BEFORE LAUNCH CANDIDATE

LOCATION Location redacted in public sample

WHAT WE FOUND

The launch path consulted cached purchase state synchronously, then started the authoritative entitlement verification asynchronously. During that window, UI that depends on paid state could trust stale local state. Implementation names and exact source locations are redacted in this public sample.

WHY IT MATTERS

The app could briefly honor an outdated paid state on launch. In a revenue path, even a short verification gap matters because monetization UI often mounts during app startup.

RECOMMENDED NEXT STEP

Gate paid-state UI on a separate entitlement-verified flag, or complete the authoritative transaction lookup before publishing paid state to the interface.

HIGH · Before launch

Completed-session statistics could be recorded twice across lifecycle paths

HIGH BEFORE LAUNCH CANDIDATE

LOCATION Location redacted in public sample

WHAT WE FOUND

Two completion paths were able to record the same finished session, while the runtime guard could reset during lifecycle transitions. The public sample redacts exact view-model names and code paths.

WHY IT MATTERS

Lifetime statistics, achievements, and streak-style counters could be inflated by replaying a completed state through resume or relaunch behavior.

RECOMMENDED NEXT STEP

Persist a per-session recorded flag and make statistics recording idempotent. Before recording, check whether the completed session already exists in durable history.

HIGH · Before launch

Completed-session resume path could re-credit statistics

HIGH BEFORE LAUNCH CANDIDATE

LOCATION Location redacted in public sample

WHAT WE FOUND

A resume path reset the runtime recording guard without first proving that the session was still eligible for scoring. Exact method names and model fields are redacted in this public sample.

WHY IT MATTERS

A completed saved state could be loaded again and credited again, inflating achievement and history data beyond what the player actually earned.

RECOMMENDED NEXT STEP

Refuse to resume sessions that are already present in completed history. Store the recorded state with the session and deduplicate writes by stable session identifier.

SECTION B

Backlog Sample

A 10-finding sample of the non-urgent items found in this codebase. The full backlog is tracked but not detailed in this deliverable — see the upgrade pitch on the next page.

#	FINDING	LOCATION	SEVERITY
1	Force unwrap crash risk in secondary view	Location redacted in public sample	MEDIUM
2	Reference-type collection warning in supporting test helper	Location redacted in public sample	MEDIUM
3	Optional initialization cleanup in game model	Location redacted in public sample	MEDIUM
4	Optional initialization cleanup in scoring model	Location redacted in public sample	MEDIUM
5	Oversized function body flagged for maintainability	Location redacted in public sample	MEDIUM
6	Optional initialization cleanup in score-entry model	Location redacted in public sample	MEDIUM
7	Optional initialization cleanup in saved-state model	Location redacted in public sample	MEDIUM
8	Optional initialization cleanup in app-state model	Location redacted in public sample	MEDIUM
9	Optional initialization cleanup in settings model	Location redacted in public sample	MEDIUM
10	Optional initialization cleanup in statistics model	Location redacted in public sample	MEDIUM

Showing 10 of 15 non-urgent findings. 5 additional non-urgent findings tracked but not detailed in this deliverable.
14 low-signal findings excluded from both sections during scoring.

SECTION 3

Urgency & Next Steps

Urgency classification

BAND	DEFINITION	IN SECTION A
Release blocker	Fix before your next App Store submission.	1
Before launch	Address in the current sprint; not a hard blocker but should not ship.	3

STEP 1 — BLOCK THE RELEASE

1 Section A finding is tagged release blockers. Fix before submitting your next build to App Store Review.

STEP 2 — SPRINT WORK

3 Section A findings should be addressed in the current sprint. Not hard blockers, but should not ship as-is.

STEP 3 — PLAN THE BACKLOG

Section B shows a 10-finding sample of non-urgent items. 5 additional non-urgent findings are tracked but not detailed in this deliverable.

STEP 4 — WANT THE FULL PICTURE?

Triage gives you every urgent finding plus a backlog sample. Surface and Standard add per-finding human verification, full backlog detail, and fix-ready remediation packets. Upgrading within 14 days of this deliverable's generated date credits the full Triage fee against the upgrade. Reply to your delivery email to discuss.

SECTION 4

Methodology & AI Disclosure

Triage is an automated, AI-curated deliverable. It uses a subset of the same scanning toolchain and Claude-based review that powers Pixelwright Digital's Surface and Standard audits. Section A surfaces every urgent finding; Section B shows a sample of the non-urgent backlog.

STATIC ANALYSIS

scc (lines of code), gitleaks (secrets in worktree & history), semgrep (built-in iOS & Swift rule packs), trivy (dependency vulnerabilities), swiftlint (style & common defects).

AI REVIEW

Claude (Anthropic) reviews tool findings and the source tree, contributing additional risks the tools miss. Authenticated via paid API key with training opted out. Surface and Standard tiers add Gemini and OpenAI for adversarial cross-validation.

SELECTION

All findings are scored, deduplicated, and ranked by severity and urgency. Section A surfaces every release-blocker and before-launch finding; Section B shows a 10-finding sample of the non-urgent backlog. 14 low-signal findings were excluded during scoring.

STANDARDS MAPPING

Paid findings reference CWE (MITRE) and OWASP MASVS where applicable. Public samples may redact identifiers that would expose private implementation details.

SECTION 5

Refund, Scope & Ownership

REFUND & CREDIT

For paid Triage clients, if any Section A finding is demonstrably contradicted by the source code as it existed at the audit commit, Pixelwright Digital will issue a credit equal to the full Triage fee, redeemable against a Surface or Standard upgrade. The credit must be claimed within 14 days of the "Generated" timestamp shown at the bottom of this page; one credit per engagement.

Excluded from credit: disagreements over severity rating, urgency band, or deduplication; disagreements over scope ("you should also have looked at X"); claims based on coverage not promised at this tier; findings whose underlying issue is real but whose remediation guidance is non-optimal.

SCOPE LIMITATIONS

Triage is a static-analysis deliverable. It does **not** include runtime testing, exploit verification, business-logic auditing, architectural review, App Store / App Review compliance certification, or any guarantee of exhaustive coverage.

A passing Triage does not imply the code is secure or shippable — it means the automated toolchain plus Claude review did not surface additional urgent issues within the scoped commit.

OWNERSHIP & IP

The report template, methodology, layout, prose, and trade dress are Pixelwright Digital intellectual property and may not be reproduced or repurposed without written permission.

The findings themselves — to the extent they concern the client's own code — are licensed to the client without restriction for internal use, redistribution to the client's contractors or counsel, and inclusion in compliance evidence packages.

Audit ID: deckscore-public-sample-20260520
Pipeline commit: macmini-rc-20260404-7-g91b10a0
Generated: 2026-05-21 01:05 UTC