

AstroStreet AR

Snapshot Summary Document

Group 2: Royce Jamerson, Khalid Jamil, Michael Lieng, Ricardo Ibanez

December 2025

1 Introduction

This document summarizes the evolution of *AstroStreet AR* across four project snapshots. Each snapshot corresponds to a different stage in the imagined 6–12 month development timeline and shows how the scope, design, and tooling matured over time.

Snapshot	Focus
Start (Snapshot 1)	Define core concept, basic AR gameplay loop, and initial documentation.
Checkpoint 1 (Snapshot 2)	Add major features (high scores, settings), refine architecture and interfaces.
Checkpoint 2 (Snapshot 3)	Deepen design (player stats, difficulty, accessibility, performance considerations).
Final / Due Date (Snapshot 4)	Polish implementation, stabilize features, and document future work.

2 Snapshot 1: Project Start

2.1 Snapshot Objective

The objective of Snapshot 1 was to establish the foundation of *AstroStreet AR*:

- Define the core game concept: a mobile AR shooter where players aim and shoot virtual asteroids and alien ships overlaid on the real world.
- Implement a minimal vertical slice:
 - Camera feed as background.

- Basic AR tracking using the chosen framework (e.g., ARCore/ARKit via Unity).
- Simple enemy objects appearing in front of the player.
- Basic projectile firing using tap input.
- Create initial project documentation and tool setup.

2.2 Features Implemented

At this stage, the following features were in place:

- Basic AR gameplay loop: camera feed, simple enemies, tap-to-shoot.
- Initial HUD text displaying at least a score counter placeholder.
- Single-player, local-only gameplay (no persistence yet).

2.3 Documentation and Tooling

- **SDD v1.0:** Introduced the high-level architecture and main components for *AstroStreet AR*.
- **SRS v1.0:** Captured the initial functional requirements and external interface expectations.
- **README / User Manual (initial):**
 - Brief project description.
 - Very basic instructions on running or building the project.
- **Jira:** Project and board created with initial tasks for core AR setup and minimal gameplay.

2.4 Risks and Open Issues

- AR tracking behavior and performance on different devices were not fully understood.
- No persistence of high scores or settings.
- No TestRail test runs defined yet; testing was informal.

3 Snapshot 2: First Checkpoint

3.1 Snapshot Objective

The objective of Snapshot 2 was to add a new major feature set and refine the requirements and design:

- Introduce a basic high-score system and dedicated High Scores screen.
- Add a Settings screen for simple configuration options (e.g., sound, sensitivity).
- Improve handling of AR session lifecycle and error conditions.
- Begin formal testing using TestRail.

3.2 Features Implemented

- **High scores:** Ability to save and display multiple high-score entries locally.
- **Settings:** Simple toggles or values for sound and sensitivity.
- **AR lifecycle handling:**
 - Basic detection of tracking loss.
 - User messages when AR is not ready or permissions are denied.

3.3 Documentation and Tooling Updates

- **SDD v2.0:**
 - Added detailed component responsibilities.
 - Documented AR session lifecycle and error handling.
 - Described key runtime scenarios (e.g., starting a session, firing and scoring).
 - Refined database design with entities such as `HighScore`, `PlayerStats`, and `Settings`.
- **SRS v2.0:**
 - Added assumptions and dependencies.
 - Added user characteristics.
 - Expanded hardware and software interface requirements.

- Extended legal and ethical considerations and glossary entries.
- **README:**
 - Updated to mention high scores and settings.
 - Included clearer instructions on basic gameplay.
- **TestRail:**
 - First Test Run created for core flows (start game, basic shooting, basic AR initialization).
 - Snapshot 2 summary exported and stored in the repository.

3.4 Risks and Open Issues

- High-score persistence needed more thorough testing across sessions.
- The difficulty curve and enemy spawn patterns were still simple and might not be engaging for longer-term play.

4 Snapshot 3: Second Checkpoint

4.1 Snapshot Objective

The objective of Snapshot 3 was to deepen the design and improve game feel, especially around difficulty and player feedback:

- Introduce aggregate player statistics to support future tuning.
- Refine difficulty progression and spawn limits.
- Incorporate accessibility and performance considerations into both requirements and design.

4.2 Features Implemented

- **Player statistics:** Tracking of total games played, best score, shots fired, hits, and total play time.
- **Improved enemy and spawn behavior:** More structured spawn limits and simple difficulty scaling.

- **Basic performance tuning:** Caps on enemies and projectiles, consideration for object reuse.

4.3 Documentation and Tooling Updates

- **SDD v3.0 (design-focused changes introduced by this stage):**
 - Game state management model (MainMenu, InitializingAR, Playing, Paused, GameOver).
 - Performance and resource considerations for mobile AR.
 - UI navigation flow between core screens.
- **SRS v3.0 (requirements-focused changes introduced by this stage):**
 - Non-functional requirements for performance, usability, reliability, and safety.
 - Future requirements and explicit out-of-scope features.
- **README:**
 - Updated to reflect more polished gameplay and design.
 - Documented the evolution of SDD/SRS versions.
- **TestRail:**
 - Second Test Run added for new behavior (player stats, improved difficulty, updated UI flow).
 - Snapshot 3 summary exported and stored.

4.4 Risks and Open Issues

- Some accessibility enhancements (e.g., fully configurable HUD sizes) remained future work rather than implemented features.
- Online or networked features were still intentionally out of scope.

5 Snapshot 4: Final / Due Date

5.1 Snapshot Objective

The objective of Snapshot 4 was to stabilize the project for final submission and clearly document remaining future work:

- Polish core gameplay loop, UI readability, and AR feedback.
- Ensure that SDD and SRS were consistent with the implemented features.
- Finalize non-functional requirements, legal/ethical considerations, and future enhancements.

5.2 Features and Polish Completed

- Small UI refinements (text clarity, button placement, HUD polish).
- Better messaging when AR tracking is unstable or unavailable.
- Cleanup and tuning of enemy caps, spawn timing, and basic difficulty.

5.3 Documentation and Tooling Finalization

- **SDD v3.0:** Final design document, incorporating all previous changes plus final clarifications on architecture, UI flow, database design, and extensibility.
- **SRS v3.0:** Final requirements document, including non-functional requirements and an explicit list of future/out-of-scope features (e.g., online leaderboards, new modes).
- **README / User Manual:**
 - Final description of features and how to run the project.
 - Links to Jira board and notes about TestRail runs and summaries.
 - Section describing documentation evolution across versions.
- **TestRail:**
 - Final Test Run created to cover the stable feature set near the due date.
 - Snapshot 4 summary exported and stored in the repository.

5.4 Future Work

Although the current version completes the course objectives, the following areas are identified as promising future work:

- Online leaderboards and cloud-synced player profiles.
- Additional game modes (timed, endless, cooperative).

- More varied enemy types and behaviors.
- Expanded accessibility options (configurable HUD, colorblind modes, haptic feedback settings).
- Potential support for additional AR platforms and devices.