Sprint 2 Artifacts

Team #10
Asa Maker, Brandon Dodge, Zach Sevart, Josh Dwoskin, Jonn Gott

Sprint 2 Goal

The goal of Sprint 2 is to establish the foundational HUD interface framework for the Meta Quest Pro using Unity. This sprint focuses on implementing the baseline HUD overlay, modular widget system, performance optimization, and layout persistence. The sprint prioritized building a stable, modular interface environment using mock data, enabling future integration of health and phone-sourced data without introducing architectural or performance risk.

Sprint 2 Requirements

- R5: HUD Overlay Baseline (MVP)
- R6: Widget System Core
- R7: HUD Performance Optimization
- R8: Session Persistence

Sprint 2 Delivered Artifacts

Artifact R5: HUD Overlay Baseline Implementation

A minimal in-headset HUD overlay implemented using Unity and deployed to the Meta Quest Pro. The HUD exists as a world-space interface anchored relative to the user's head position and provides a stable container for modular widget components. Visibility controls implemented to allow the HUD to be enabled or disabled at runtime. This artifact establishes the core visual framework required for all future widget rendering.

- Unity scene hierarchy showing HUD anchor and Canvas structure
- Screenshot of HUD overlay running on Meta Quest Pro
- HUD positioning and anchor configuration
- HUD visibility toggle implementation

Artifact R6: Widget System Core Framework

A modular widget system implemented to support dynamic creation, placement, repositioning, and removal of widgets within the HUD environment. Widgets designed as independent Unity prefabs with their own transforms and behavior scripts. A centralized Widget Manager implemented to handle widget lifecycle and ensure modularity. Mock test widgets used to validate system functionality.

- Widget Manager system implementation
- Example widget prefab structure
- Screenshot of multiple widgets attached to HUD
- Widget creation and removal test results

Artifact R7: HUD Performance Optimization

Performance optimization performed to ensure stable rendering performance at or above 72 frames per second on Meta Quest Pro hardware. Optimization focused on reducing unnecessary update loops, improving UI hierarchy efficiency, and minimizing rendering overhead. Performance validated using Unity Profiler and Quest performance monitoring tools.

- Unity Profiler performance capture
- Quest Pro performance monitoring results
- Optimized update loop and rendering configuration
- Measured framerate stability results

Artifact R8: Session Persistence System

A persistence system implemented to store and restore HUD layout and widget configuration across application sessions. Widget identifiers, positions, and active states saved locally and restored when the application restarts. This ensures a consistent user interface experience and enables reliable long-term widget configuration.

- Persistence system implementation script
- Saved layout configuration example
- Demonstration of layout restoration after application restart
- Verification of reliable state recovery

Sprint 2 Risks Identified & Mitigations

- Risk: HUD interface instability or discomfort in headset
    - Mitigation: Stable world-space anchoring and performance validation
- Risk: Widget system architectural limitations
    - Mitigation: Modular prefab-based widget design and centralized lifecycle management
- Risk: Performance degradation from UI rendering
    - Mitigation: Early performance optimization and profiler validation
- Risk: Persistence system failure or state corruption
    - Mitigation: Structured local storage and controlled state restoration process

Sprint 2 Outcome Summary

Sprint 2 successfully established the core interface architecture for the Meta Quest Pro HUD system. The team strives to implement a stable HUD overlay, modular widget framework, performance-optimized rendering pipeline, and layout persistence system. These artifacts provide a fully functional interface foundation capable of supporting future integration with phone-sourced or “Quest Move” health and sensor data. The successful completion of Sprint 2 enables Sprint 3 to focus on Android integration and communication bridge development without requiring changes to the HUD interface architecture.