

Executive Summary

Piano Hero is an augmented reality piano-learning system that overlays visual guidance onto a real keyboard using computer vision and real-time interaction inference. The system tracks a player’s hands via Google MediaPipe, detects ArUco markers for stable keyboard alignment using OpenCV, and renders AR overlays to assist with timing and accuracy during practice.

Built in Unity and tested on Android 13, the app uses a smartphone as the processing hub and routes its display to XREAL 1S AR glasses via wired USB-C mirroring. A rhythm-game interface presents falling notes synchronized to song timelines, while real-time hand tracking provides scoring and interactive feedback.

The project demonstrates modular software design, real-time input processing, game logic, and user feedback systems across seven Agile sprints.

Background

- Learning piano traditionally requires expensive instruction or self-guided practice with limited feedback. Existing tools rely on audio recognition and on-screen displays, disconnecting learners from the physical keyboard. AR offers a compelling alternative by overlaying guidance directly onto the instrument.
- Piano Hero is inspired by Stanford’s homography-based AR piano research using commodity hardware: a smartphone, ArUco markers, and AR glasses.

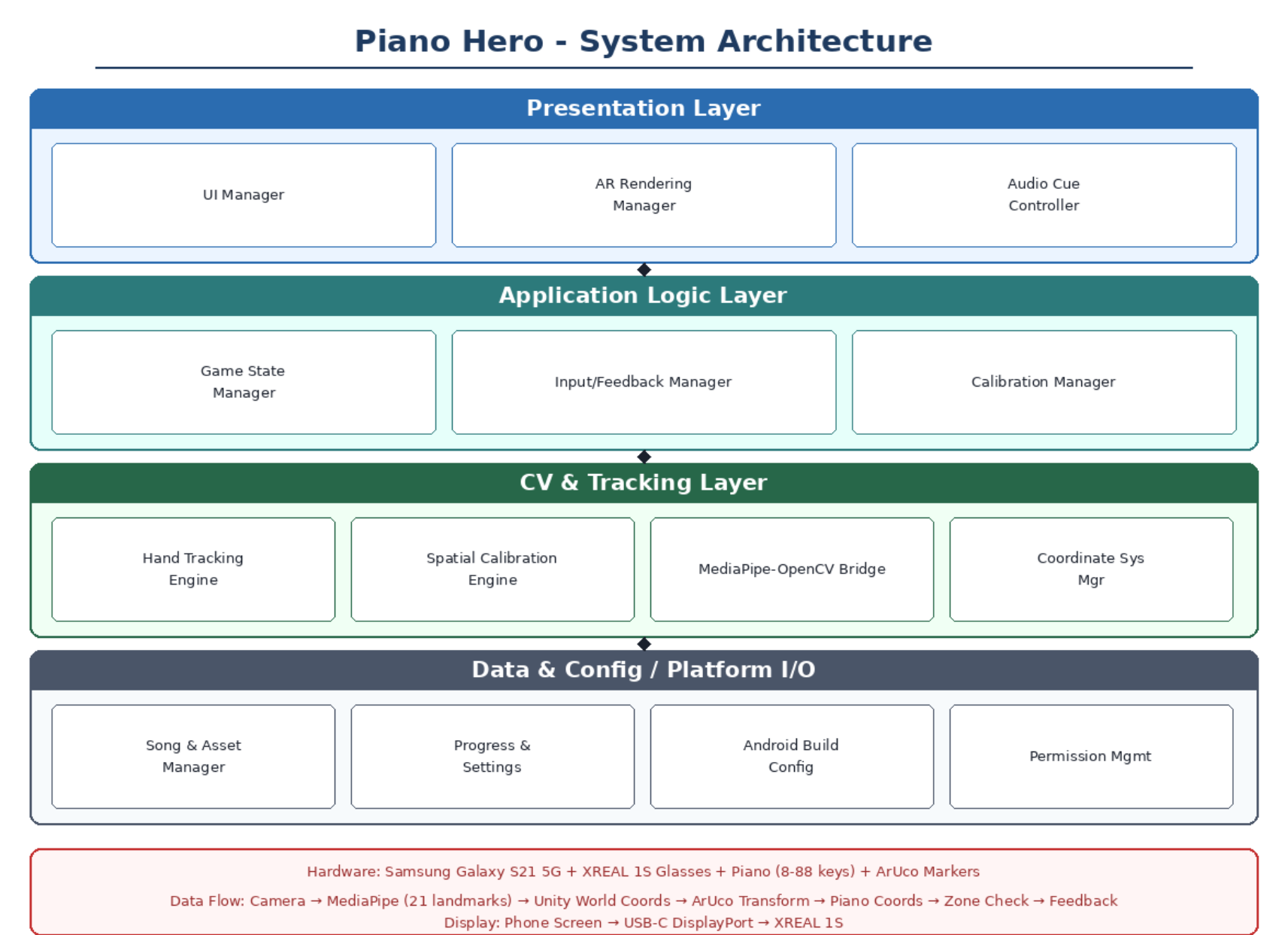


Figure 1. Piano Hero Conceptual Architecture

System Architecture

The architecture follows a four-layer model:

Presentation: UI, AR Rendering, Audio Cues

App Logic: Game State, Input/Feedback

CV & Tracking: Hand Tracking, Spatial Calibration

Data & Config: Songs, Progress, Platform I/O

Table 1. Key Technology Stack

Component	Technology	Purpose
Hand Tracking	MediaPipe Hand Landmarker	21 landmarks per-hand detection at 25+ FPS
Spatial Calibration	OpenCV ArUco	Marker detection, depth estimation, piano plane construction
Spatial Reference	ArUco Markers (DICT_4x4_50, IDs 0-3, 6cm)	Physical piano plane anchoring and 3D coordinate reference
Game Engine	Unity (Android Build)	Scene rendering, game logic, dual-camera system
Processing Platform	Samsung Galaxy S21 5G	On-device CV + Unity runtime, rear camera source
AR Display	XREAL 1S (USB-C)	Wired AR glasses display via USB-C DisplayPort output

Key Subsystems

Hand Tracking Engine: Subscribes to MediaPipe landmark output each frame. Uses the knuckle (MCP) joint for horizontal zone positioning and the fingertip for press detection. Casts rays through knuckle screen positions and intersects with the calibrated piano plane to produce world-space fingertip positions.

Spatial Calibration Engine: Detects ArUco markers via OpenCV, estimates depth using perspective projection, computes plane center/normal from 4 marker positions, and publishes a piano-to-world coordinate transform.

Game State Manager: Schedules note arrivals from a SongData asset, spawns falling note cubes above target key zones, hit-tests fingertip positions against zones when notes are visually inside the zone box, triggers audio playback and zone hit pulses on successful hits, and increments the score via ScoreManager.

Implementation Results

The Unity + MediaPipe + OpenCV pipeline was confirmed stable on Android (Samsung Galaxy S21 5G, 128 GB). Key outcomes across eight sprints:

- Hand tracking at 25+ FPS with 21 landmarks per hand via MediaPipe
- Spatial calibration using 4 ArUco markers (DICT_4x4_50, 6cm) with coplanar ray-plane projection
- Fingertip spheres rendered in AR, anchored to calibrated piano plane
- Wired USC-C DisplayPort output to XREAL 1S AR glasses
- Falling notes gameplay with audio-synchronized timing and hit/miss scoring

Conclusions

Piano Hero successfully demonstrates that commodity mobile hardware can power a real-time AR piano learning system. The modular, layered architecture isolates perception, logic, and rendering, enabling independent tuning and future extension. Key lessons learned include the critical role of on-device GPU acceleration for sustaining 25+ FPS hand tracking, and that wired display output eliminated the latency and reliability issues of wireless mirroring. Future work includes advanced timing analytics, cloud-based progress tracking, and a community-driven song library with user-uploadable content. We thank Dr. Wang for his guidance and support.

References

1. Google, “MediaPipe Hands: On-device Real-time Hand Tracking,” 2023. mediapipe.dev
2. OpenCV, “ArUco Marker Detection,” OpenCV Documentation, docs.opencv.org
3. Unity Technologies, “AR Foundation,” Unity Documentation, docs.unity3d.com
4. Stanford CS, “AR Piano: Homography-based Spatial Registration for Augmented Reality Piano,” Stanford University.