

COREY HENDERSON

SOFTWARE ENGINEER

6 years of experience in game development, specialising in systems architecture, multiplayer networking, and VR using Unity (C#). Known for delivering clean, scalable, and high-performance gameplay systems. A passionate gamer and creative problem-solver who thrives in dynamic environments and loves tackling complex technical challenges.

+61 455 459 270

corey.henderson.dev@gmail.com

Melbourne, Australia

coreyhenderson2.wixsite.com

linkedin.com/in/corey--henderson

EXPERIENCE

SEP 2023 - APR 2025

PlaySide Studios - Software Engineer

- Accomplished the full development lifecycle of the published VR game Dumb Ways: Free For All, including two post-launch DLCs, as measured by successful release and continued updates on the Meta Quest Store, by leading back-end systems development in Unity (C#) from inception.
- Collaborated within a team of 40+ other developers delivering core functionality for multiplayer minigames and online play.
- Implemented multiplayer networking components, AI steering/pathing systems, and interactive VR mechanics such as slingshots, firearms, melee weapons, and throwable items.
- Partnered with designers to build custom components and assisted art team in reducing draw calls to optimise performance.
- Worked closely with QA to action high priority issues and bugs.
- Integrated unit tests for features and reviewed team merge requests to ensure code quality.
- Trained developers on Git best practices and used TeamCity for build integration.
- Engineer responsible for reviewing minigame design briefs before they were approved for development.
- Selected for the skeleton crew during final development weeks, providing timely delivery of a polished final build, by resolving critical bugs and executing last-minute performance optimisations.
- Worked in a proprietary engine for published VR game Bobber Bay Fishing, guiding multiplayer integration using built-in net code.

JUL 2019 - SEP 2023

Lightmare Studios - Lead Programmer

- Accomplished full-stack development for Infinity Wars Classic, a digital TCG by Lightmare, by building front-end and back-end systems in Unity and leading a small engineering team through roadmap planning, task delegation, and pipeline management.
- Handled PlayFab cloud services, for player account management and automated scheduled tasks for daily rewards and store rotation.
- Maintained AWS database health on RDS and analysed service costs.
- Delivered monetisation systems for in-game purchases using virtual currency and Steam wallet transactions, by implementing store purchasing flows through PlayFab and SteamWorks API.
- Designed and implemented the offline AI system for card combination calculations and opponent turn action predictions.

OCT 2019 - APR 2020

Deakin University - Software Developer

- Sole developer on delivering a Windows application to automate PDF report generation, ensuring elimination of manual reporting tasks, by designing and building a standalone Windows Forms program in C# (.NET Framework).
- Streamlined data processing through transformation of spreadsheet inputs into formatted PDFs, by integrating external libraries for Excel parsing, JSON serialization, and document stream writing.
- Handled client communication to manage and discuss features being implemented and provided feedback for the feasibility of new features, estimated time frames and budget costs.

EDUCATION

2016 - 2019

DEAKIN UNIVERSITY

Bachelor of Game Design and Development

- Achieved High Distinction average in game development units.

SKILLS

TECHNICAL

- C#, TypeScript, JavaScript
- Minor experience in C++, Node.js, Java, HTML, CSS, Python, PHP, XML, SQL, OpenGL
- Unity3D Engine (primary), Unreal Engine 4 (minor experience), Proprietary Engine's
- Photon Multiplayer Networking, Fusion, PUN, Matchmaking, Chat
- PlayFab Cloud Services, client-side API, LiveOps, Admin billing & management
- AWS Lambda, RDS Database Management, SES Emailing, Cloud9 & EC2 instances.
- FMOD Audio Library in Unity C#
- Oculus VR Development
- Steamworks API, Inventory Service, Build Publishing, Patch Notes

SOFTWARE

- Visual Studio, VS Code
- Git, Fork, SourceTree
- Gitlab, Bitbucket
- Jira, Trello, Hansoft/Helixplan
- TeamCity build pipelines

GENERAL

- Writing technical documentation
- Leading teams and mentoring junior engineers.
- Outlining features and functionality for development
- High-level code architecture diagrams

EXPERIENCE (CONTINUED)

Unity

- Worked extensively with UI layout components building custom UI with full back-end script integration. Have done a complete overhaul of main menu UI, refactoring the architecture of classes, and splitting up canvas' for improved performance.
- XR/VR Toolkit and Oculus SDK integration in Unity.
- Built feature unit tests for build stability and merge request build pipelines in the Test Runner.
- Created various custom editor windows, for project asset management, unit test validations & API request testing using Editor GUI. Created script editors and custom drawers of mono behaviours for debugging and allowing designers ease of access for testing components.
- Async/await operations of various features and API responses, including UniTask asynchronous library.
- Integrated back-end audio services and components using FMOD API audio library.

Photon (multiplayer networking)

- Used Fusion V1 and V2 in Unity for session connection of players joining lobbies, setting up room properties, syncing player states, and creating solutions for real-time syncing of gameplay events using RPC's.
- Implemented peer-to-peer real-time network communications for multiplayer gameplay actions in PUN.
- Added matchmaking & game queuing support in Photon Matchmaking, private game inviting and friend online status', using a hybrid system between Photon Chat & PlayFab.

PlayFab (back-end cloud services)

- Handled player accounts, updating custom user data, implementing player statistics, supporting inventory item granting & consuming, and friend list support.
- Managed cloud-script functions (JavaScript) for client-server API communications, to retrieve and modify user account data, execute scheduled tasks, and setup push notifications for mobile.
- Implemented scheduled tasks for leaderboard rewards, weekly store rotations, inventory data validation and mass player item granting.
- Implemented store/shop purchasing systems for virtual currencies and real money transactions through user Steam wallets (Steamworks).

Amazon Web Services

- Helped manage project AWS cloud, integrating client-server communications with the database (Amazon RDS) through Lambda, creating and setting up all functions in Node.js.
- Ensured backend reliability and cost-efficiency of RDS database health and optimized resource spending, by monitoring system performance and analyzing service costs, reducing company monthly spendings by 50%.
- Setup email templates and implemented email validation through Amazon SES. Handled users, roles and permissions in IAM. Setup and managed identity pools for client-server API communications.
- Used the Cloud9 virtual environment to setup and debug Lambda function code, using AWS CLI to install npm modules and update function configurations. Tracked EC2 instances to stop unused services and reduce company resources.

Misc

- Oculus VR development in different engines, using Meta Quest devices, including developer tools for app deployment, profiling, performance analysis, and debugging.
- Developed networking systems and FMOD audio services in an unreleased proprietary engine.
- Created navigation systems for AI steering agents which included target seeking, random wandering, waypoint pathing, collision avoidance, and movement bounds. This was used for all dynamic animal movement for underwater, flying, and terrain based creatures.
- Proficient usage of Git source control practices, using professional commit types, rebasing, merging, conflict resolving, trunk-based development, and MR/PR practices.
- Implemented Steam back-end, supporting in-game overlay, and player account linking. Setup Steamworks inventory service for creating item definitions and loot box item generators.

References available upon request