PETER CAREY

SOFTWARE DEVELOPER

LANGUAGES

C# ♦♦♦♦♦
C++ ♦♦♦♦
Python ♦♦♦
.js / .tsx ♦♦♦
Rust ♦♦

EDUCATION

2014 - 2016

Bachelor - Game Design

Focus: C# and Unity

2021 - 2023

Master - Creative Industries

Research: VR experience design and semiotics in interactive systems

---

CONTACT

Email: petermarkcarey@gmail.com

Phone: (+61) 497 797 473

LinkedIn: linkedin.com/in/pmcarey/

GitHub: github.com/DraconInteractive

Portfolio: https://pcarey.dev

SKILLS

APPLICATION DEVELOPMENT

.NET application architecture
ASP.NET web applications
WPF Desktop applications
React + Electron front end applications

API & INTEGRATION

REST / JSON web services
Service integration and API design
Authentication and external system integration
Data serialization and messaging pipelines

MACHINE LEARNING INTEGRATION

On-device ML inference
Hybrid ML + Utility AI systems
Model training and deployment pipelines
TensorFlow / ML-Agents integration

DEVOPS & INFRASTRUCTURE

CI/CD (GitHub Actions, Buildkite)
Cloud services (AWS, Azure)
Build automation & artifact pipelines
Release and environment management

DEVELOPMENT PRACTICES

Agile development workflows
Code review and quality assurance
Technical documentation and standards
Secure development and privacy practices

HARDWARE & DEVICE INTEGRATION

Bluetooth device communication
Sensor telemetry processing
Embedded device integration (ESP32)

PETER CAREY

SOFTWARE DEVELOPER

WORK EXPERIENCE

Senior Developer (Unreal C++, WPF C#)

BetterThanReality (Official Site) 2025 - 2026

Developed industrial training applications for heavy equipment systems, combining real-time simulation software with supporting desktop tooling. In addition to the simulation environment, I developed a C# WPF desktop application used to configure, launch and manage training scenarios.

The primary project involved modelling the operational behaviour of a Komatsu haul truck, implementing a structured state system representing electrical isolation, hydraulic circuits and mechanical components. System interactions were driven by state changes and event logic derived from equipment schematics and maintenance procedures.

I implemented interaction frameworks and component logic allowing users to inspect and manipulate equipment systems while progressing through procedural workflows such as inspection, isolation and maintenance tasks.

The system integrated with a Go-based backend service responsible for authentication and training data storage. Both the launcher application and simulation runtime communicated with the backend through REST APIs, with data persisted in a PostGres database.

These systems combined application logic, state-driven modelling and backend integration to deliver repeatable training and evaluation environments for technical personnel.

Senior Developer (Full Stack - Rust, C#, C++, Javascript)

Futureverse Inc (X: Project Announcement) 2023 - 2025

Developed a cross-platform platform for generating and rendering 3D assets from procedural graph definitions. The system consisted of a desktop graph editor, a shared runtime library, and engine integration layers that interpreted graph instructions and generated runtime assets.

I contributed to the development of UBF Studio, a desktop application built with React, Electron and Node.js. The application allowed users to author node graphs, manage asset resources such as meshes and textures, and export packaged bundles containing all dependencies required for runtime execution.

The platform used a core runtime library written in Rust, responsible for deserializing graph definitions, validating node relationships and executing instruction pipelines defined by the graphs. This runtime acted as an engine-agnostic execution layer that ensured consistent behaviour across multiple host environments.

I implemented runtime integration layers for both Unity and Unreal, written in C# and C++ respectively, allowing the platform to interpret graph instructions and dynamically construct assets and application state within each engine. These integration layers bridged the Rust runtime with engine APIs and handled resource binding, rendering configuration and runtime asset construction.

The system integrated with a GraphQL backend service used for retrieving graph definitions and asset metadata, with packaged assets distributed through AWS cloud infrastructure for runtime consumption by supported environments. User authentication and relationship data were stored in a secure PostGres database hosted on-site.

In addition to runtime development, I helped define and maintain the graph schema specification, including node definitions, validation rules and versioning standards used across the editor, runtime and engine integrations.

I also developed CI/CD pipelines used to validate graph definitions, build runtime libraries and distribute compiled artifacts across supported platforms, ensuring compatibility between editor, runtime and integration layers.

PETER CAREY

SOFTWARE DEVELOPER

WORK EXPERIENCE

Senior Developer (Unity C#, Python TensorFlow)

Altered State Machine (YT: Project Announcement) 2023 - 2023

Developed mobile applications for iOS and Android integrating machine learning–driven decision systems into a real-time runtime environment.

I integrated TensorFlow-built models into the Unity application runtime, enabling on-device inference for multiple agents during execution. Model outputs were evaluated by a rule-based decision system that selected actions based on the current application state.

A key focus was runtime performance optimisation, enabling 16 model inferences per second across eight agents while maintaining ~75 FPS on older Android devices, allowing all decision logic to run locally without server-side processing.

I also implemented systems for training data collection and model integration using Unity ML-Agents, extending the framework to support custom model configurations and agent behaviours. Training input data and output artifacts were sorted, processed and stored using a python oriented CI/CD processes.

In addition, I contributed to CI/CD pipelines used to build training environments, run automated training sessions, and generate model artifacts and analytics for integration into mobile application builds. Training sessions were run on containerized EC2 sessions, each session running multiple parallel training sesions. By injecting training parameters into the CI process, we were able to generate model varietals en-masse, with data analytics being used to automate model sucess / selection.

Head of Development (Unity C#)

Coortical 2022 - 2023

Joined the company at its formation and was responsible for establishing the development capability and technical direction of the business. This included recruiting the initial engineering team, defining the technology stack and development workflows, and implementing the infrastructure required to support multiple client projects.

Development was centred around Unity applications written in C#, targeting standalone VR devices. I designed and implemented core application frameworks used across projects, including modular interaction systems, reusable UI components and shared application templates that accelerated development across the team.

I also established the company's engineering infrastructure, implementing GitHub-based version control, shared package repositories and project templates. Development followed Agile practices with two-week sprint cycles, while CI/CD pipelines and AWS S3 storage were introduced to support build automation and artifact distribution.

In addition to architectural responsibilities, I remained the primary hands-on developer, mentoring junior engineers while implementing core application systems and supporting client project delivery.

During this period the company delivered several client applications, including projects requiring integration with external systems such as Learning Management Systems (LMS) for authentication, training progress tracking and reporting.

PETER CAREY

SOFTWARE DEVELOPER

WORK EXPERIENCE

Senior Developer (Unity C#, ESP32 C++)

Zarmada (Project Site) 2021 - 2022

Developed C# applications using Unity for immersive training environments targeting HTC Vive and later Oculus Quest hardware.

A key component of the project was integration with a custom Bluetooth peripheral attached to a real taser device. I implemented the communication layer responsible for receiving device telemetry, including trigger state, safety state and IMU rotation data, and deserializing the device packet protocol for use within the application runtime.

The application translated hardware events into system behaviour, modelling device operation and enabling users to interact with the equipment through a controlled training environment.

I also contributed to the development of interactive scenario systems, including a voice-driven workflow where users selected dialogue responses and branching outcomes were triggered through voice recognition.

The project was delivered by a small engineering team under tight timelines, requiring rapid iteration and close coordination across hardware integration, application logic and user interaction systems.

Lead Developer (Unity C#, ESP32 C++)

St John WA (Project Site) 2018 - 2021

Developed a cross-platform training application platform using C# and Unity, delivering interactive learning modules across PC, mobile and web environments. Working with subject-matter experts, the existing Senior First Aid course was restructured into approximately thirty modules combining guided instruction with timed assessment scenarios.

The application supported PC VR, iOS, Android and WebGL, requiring the design of interaction systems that functioned across both immersive and traditional interfaces. Shared application frameworks were implemented to support user interaction, workflow progression and evaluation across all platforms.

To support the wide deployment targets and hardware constraints, I implemented optimisation and asset management systems including platform-specific configuration of assets and runtime resource management using Unity Addressables.

A key component of the system was integration with a custom CPR training peripheral built around an ESP32 microcontroller attached to a training manikin. I implemented firmware for the device and developed the application-side analysis systems used to evaluate compression depth and compression rate in real time.

I also developed CI/CD pipelines using GitHub Actions to automate application builds and deployment across environments, including generation of addressable asset bundles and distribution through staging and production infrastructure.

The platform integrated with the organisation’s backend systems to support authentication, course management and certification tracking through a Learning Management System (LMS).

PETER CAREY

SOFTWARE DEVELOPER

TO WHOM IT MAY CONCERN,

I am writing to express my interest in the AI & Mining Operations role. My background focuses on building real-time simulation systems, runtime platforms, and machine-learning driven decision systems.

Across my recent roles I have worked on systems that model physical processes and translate those models into interactive or operational software. At BetterThanReality I developed simulation systems in Unreal C++ that modelled the operational state of a Komatsu haul truck, including hydraulics, isolation procedures, and subsystem interactions. These simulations were used to train engineers to diagnose faults and perform maintenance procedures, requiring the system to accurately represent the behaviour of complex mechanical systems.

Earlier in my career I worked on AI-driven agent systems developed in collaboration with FIFA, integrating TensorFlow models into a real-time Unity runtime. These systems combined machine learning inference with utility-based decision frameworks to drive player behaviour in real time, requiring careful optimisation to run efficiently on mobile hardware. I have also built cross-platform runtime systems in Rust and C++ that interpret abstract procedural graphs and generate runtime assets across multiple engines, as well as hardware-integrated simulations that process sensor data and device telemetry in real time.

A recurring theme throughout my work has been translating abstract systems (ML models, physical equipment schematics, or procedural graphs) into robust, production-ready runtime software. I particularly enjoy working on foundational systems and early-stage platforms where the architecture and core modelling decisions shape how the technology evolves over time.

The opportunity to apply these skills in creating a real-time simulation layer is intriguing, and I would welcome the chance to contribute to a platform where simulation directly informs operational outcomes in the field.

Thank you for your consideration. I look forward to the opportunity to discuss the role further.

SINCERELY,

Peter Carey