

# Dave Rzepa

Software engineering · Technical leadership · Contract work

San Cristóbal de las Casas, Mexico | Global contract opportunities

david.rzepa@gmail.com | [www.dave-rzepa.dev/ask](http://www.dave-rzepa.dev/ask)

<https://www.linkedin.com/in/david-rzepa-60012a3a/>

I build reliable systems, help teams deliver, and make complex software easier to work on. My experience spans backend services, real-time networking, cloud infrastructure, developer tools, graphics, and engineering leadership. I'm bringing that breadth to contract projects across industries.

## Relevant experience

### Lead Programmer · Parallel

Jun 2024 – 2026

#### Made an application inspectable by AI agents

Built a Unity MCP server and driver that let an LLM play and inspect the application, alongside agent test scenarios, profiling, and a coverage ledger.

MCP · AI agents · LLM integration · C# · Python · test automation · Unity

#### Automated issue-to-review engineering workflows

Developed an autonomous bug-fixing workflow, deterministic issue triage, a PR readiness dashboard, and build-provenance checks connecting Jira, GitHub, and delivery pipelines.

Jira · GitHub · CI/CD · developer experience · automation · AI agents · Python

#### Established reusable agent development practices

Created repository instructions, canonical agent skills, playbooks, and coordination around shared Unity resources.

AI agents · documentation · developer experience · workflow design · resource coordination

#### Built cloud observability with infrastructure as code

Built Grafana infrastructure on Google Kubernetes Engine using Terraform, with KPI dashboards and an analytics-breadcrumb prototype.

Terraform · Grafana · GKE · Kubernetes · Google Cloud · observability · analytics

#### Owned build and release automation

Built and maintained GitHub-based build pipelines, platform build matrices, caching, versioning, and release delivery through Applivery and Steam.

GitHub Actions · CI/CD · build engineering · release engineering · caching · Unity

#### Refactored authentication and backend integration

Worked on login architecture, Google Play Games and Steam authentication, token refresh, and API-client integration.

authentication · token refresh · API integration · C# · backend services

#### Improved network and serialization correctness

Refactored low-level netcode and worked on resynchronization, serialization, replay correctness, a serialization analyzer, and server tests in CI.

networking · serialization · distributed systems · C# · static analysis · testing · CI/CD

### **Adapted a desktop application for mobile**

Optimized an existing PC game for mobile, including texture and GPU work, blur render textures, mobile rendering, and shaders.

performance optimization · profiling · mobile · GPU · rendering · Unity · HLSL

### **Rebuilt and extended a complex client UI**

Rebuilt the client UI, including collection and deckbuilder views, page architecture, card movement, draft-mode screens, and reward reveal flows.

UI architecture · C# · Unity · refactoring · state management

### **Built AI simulation and evaluation tooling**

Developed a headless match harness, heuristic AI, genetic deck evolution, cloud benchmarking, and server-side solo AI decks.

simulation · heuristic AI · genetic algorithms · benchmarking · C# · cloud computing

### **Connected structured design data to LLM workflows**

Worked on data for LLM-assisted card design, starter decks, and progression and unlock systems.

LLM integration · data pipelines · C# · API integration

## **Principal Programmer · Offworld**

Jan – Jun 2024

### **Modernized a production engine integration**

Updated Starship Troopers to Unreal Engine 5.4, including Nanite and World Partition.

Unreal Engine · C++ · Nanite · World Partition · engine migration

## **Lead Programmer · PopReach**

2021 – Aug 2023

### **Hired and mentored eight programmers**

Interviewed, hired, and mentored a team of eight programmers. Evaluated an inherited codebase and architected and estimated a rebuild for PAYDAY: Crime War.

technical leadership · hiring · mentoring · architecture · estimation · code review

### **Established shared application foundations**

Implemented dependency injection with VContainer, async/await state machines and API wrappers, a PlayFab configuration pipeline, a UniRx UI system, and an Addressables pipeline, then trained the team in their use.

C# · dependency injection · async/await · reactive programming · architecture · configuration management · Unity

### **Integrated managed backend services**

Integrated PlayFab authentication, remote configuration, player data and inventory, matchmaking, and multiplayer servers.

PlayFab · authentication · backend services · data storage · configuration management · API integration · C#

### **Built serverless functions and user-data migrations**

Used Azure Functions with a shared DLL between client and server, robust user-data migrations, and queue-backed leaderboard prize granting using Azure Queue Storage.

Azure · Azure Functions · C# · serverless · data migration · queues · distributed systems

### **Integrated real-time communication**

Integrated Photon Fusion for real-time networking and Photon Chat for messaging.

real-time systems · networking · Photon Fusion · messaging · C#

### **Supported live delivery and operational visibility**

Deployed eight feature releases and three hotfixes. Added New Relic logging, dashboards, and alerting, and helped the analytics team with KQL queries and technical graphs.

release engineering · New Relic · observability · KQL · analytics · incident investigation

### **Built continuous deployment across clients and backend environments**

Implemented Jenkinsfile-based continuous deployment with AWS build agents, Google Play and TestFlight publishing, and a .NET tool for deploying branches and pull requests to PlayFab environments. Included Slack integration, restartable stages and deployment guards.

CI/CD · Jenkins · AWS · C# · .NET · PlayFab · TestFlight · Google Play · release engineering

## **Senior Programmer · Bossa Studios**

2018 – 2021

### **Built a data-oriented entity system**

Developed a pure C# entity-component system for Worlds Adrift, usable inside and outside Unity, with bitwise entity filtering and T4-generated components and marshaling to synchronize SpatialOS and ECS data. The 2019 CV details batch component changes, JSON entity blueprints with multiple inheritance, throttle masks for amortizing system work, Zenject dependency injection, and unit tests around core routines.

C# · data-oriented design · ECS · code generation · distributed systems · SpatialOS · performance optimization

### **Solved latency and distributed simulation handovers**

Implemented spline-based extrapolated ship movement for latency compensation and handover between physics servers, NTP time synchronization, and diagnostic graphing in Worlds Adrift.

distributed systems · networking · time synchronization · NTP · latency · C# · simulation

### **Extended asynchronous cloud persistence**

Modified Scala server code to save and load ship-blueprint XML through AWS, making extensive use of futures and promises.

Scala · AWS · asynchronous programming · XML · backend services · data storage

### **Implemented custom GPU rendering and parallel physics**

Built custom Unity HDRP effects for I Am Fish: underwater scattering and absorption with compute shaders and signed distance fields, water normals, caustics, reflections, procedural glass caustics, and a jobified buoyancy simulation.

HLSL · compute shaders · GPU · rendering · physics · parallel programming · Unity · mathematics

### **Built custom visibility and performance systems**

Developed custom occlusion culling for Surgeon Simulator 2 with jobified runtime baking and compute-shader portal visibility, and optimized and debugged custom netcode. Archived CVs also list UI and HTTP networking work on Surgeon Simulator 2.

compute shaders · HLSL · parallel programming · performance optimization · networking · Unity

### **Tuned autonomous behavior and control systems**

Originated Hogwash during a game jam and developed and tuned behavior-tree heuristics, a PID controller for path following, and ORCA dynamic avoidance.

heuristic AI · behavior trees · PID controllers · control systems · simulation · mathematics · C#

### **Led a server-side SDK and architecture migration**

The 2019 CV describes spearheading the Worlds Adrift migration of server-side logic from the SpatialOS Scala SDK to the C# SDK, including a bespoke data-oriented ECS and generated synchronization systems.

C# · SpatialOS · distributed systems · code generation · software architecture · migration

### **Reduced vector serialization size in a distributed simulation**

Implemented a new Vector3 serialization schema for Worlds Adrift, reducing the representation from 192 bits to 96 bits per entity; the archived CV reports that often only 64 bits changed. This is a representation-level improvement, not a claim of overall network bandwidth reduction.

serialization · networking · C# · bitwise operations · performance optimization

### **Validated an ECS across client and server**

Rewrote Worlds Adrift's Blight feature using the ECS on both client and server, exposing and filling gaps in the framework API and demonstrating its use for a substantial production feature.

C# · ECS · API design · client-server systems · refactoring

### **Delivered the final Worlds Adrift update with its community**

As the last developer on the Worlds Adrift team, designed and implemented its final update in collaboration with the community, including the end-of-world events.

delivery · community collaboration · product design · C# · live operations

### **Worked on Python web services and data science**

An archived Bossa project lists Django, Celery, natural language processing, deep learning and statistical analysis. The 2019 CV describes learning Django and researching data-science and machine-learning techniques. The records do not establish model architectures, dataset sizes, production scale, or quantified outcomes.

Python · Django · Celery · NLP · deep learning · statistical analysis · machine learning

## **Lead Programmer · Ubisoft London**

2014 – 2018

### **Led delivery on a long-running mobile product**

Managed an internal team of three programmers on Hungry Shark Evolution, maintained an established codebase, and delivered gameplay features, optimizations, and bug fixes.

technical leadership · team management · mobile · maintenance · performance optimization · delivery

### **Built a rapid-prototyping framework and six prototypes in twelve weeks**

At Future Games of London (Ubisoft), built a framework for rapid prototyping and six distinct prototypes in twelve weeks. Developed a fuller gameplay and frontend prototype and an online multiplayer prototype using deterministic lockstep with Photon TrueSync.

prototyping · framework design · Unity · C# · networking · deterministic lockstep · Photon TrueSync

### **Integrated cloud features and live operations on a mobile product**

Implemented cloud-save frontend, remote JSON configuration and Amazon Insights A/B testing, analytics, error logging with LogEntries, customer-support integrations, and GameSparks live-event client, backend logic and dashboards. Later replaced GameSparks live events with an in-house backend and built workflow tools for reviewing remote configuration changes.

backend services · remote configuration · A/B testing · analytics · observability · API integration · GameSparks · mobile · tooling

## Unity programmer, game designer and Head of Games · UNIT9

2010–2014 (documented projects; Head of Games in 2014)

### Head of Games: strategy, proposals and hiring (2014)

Performed market research and developed business strategies. Served as technical and game-design consultant and prototyper on business proposals, and interviewed applicants for multiple roles.

technical leadership · market research · business strategy · prototyping · hiring · technical consulting

### Creative direction and lead engineering on Decromancer (2013)

Creative Director and Lead Unity programmer: game design, UX, narrative and copywriting, art direction and marketing, alongside C# engineering using NGUI and Flurry Analytics for iOS and Android. Worked closely with an illustrator, provided creative briefs and feedback, and developed narrative and gameplay direction.

Unity · C# · NGUI · Flurry Analytics · iOS · Android · UX · creative direction · product design

### Procedural systems and lead development on Electric Tentacle (2012)

Lead Unity programmer and lead game designer. Built modular infinite-level generation using Perlin noise, allowing zone-specific changes without code edits, plus object pooling for performance. Used C#, SpriteManager2 and Flurry Analytics for iOS and Android, and designed gameplay and UX.

Unity · C# · procedural generation · Perlin noise · object pooling · iOS · Android · UX · performance optimization

### Procedural geometry and scoring for Domino's Pizza Hero (2011)

Lead Unity programmer. Developed real-time mesh and texture deformation and mesh relaxation algorithms, analyzed dough shape to score evenness, and generated simplified mesh colliders for topping physics. Integrated native plugins and Google Analytics, and built UI view and transition systems.

Unity · C# · procedural geometry · mesh deformation · physics · algorithms · native plugins · Google Analytics · UI architecture

### Programming, gameplay and level design for Nano Panda (2011)

Documented as solo programmer in the 2012 CV and lead Unity programmer in the 2014 CV. Implemented the project in C# with EZGUI and SpriteManager2, and contributed gameplay and level design.

Unity · C# · EZGUI · SpriteManager2 · gameplay design · level design

### Unity development and game design for Xtreme Xrunch Kart (2010)

Unity programmer and game designer, contributing C# implementation, Unity GUI, gameplay design and level design.

Unity · C# · UI · gameplay design · level design

### Programming, game design and sound design for W.A.R.P. (2010)

The 2012 CV records programming, game design and sound design responsibilities on W.A.R.P. at UNIT9.

Unity · programming · gameplay design · sound design

### Built reusable architecture and interface frameworks

Developed internal Unity frameworks, UML diagrams and technical specifications, including interfaces between Unity and layers built by other teams. Built custom view and transition sequencing frameworks used in Domino's Pizza Hero and Electric Tentacle, plus procedural Bézier and B-spline track generation.

software architecture · UML · technical specifications · API design · UI architecture · C# · splines · framework design

### Combined analytics, UX and rapid prototyping

Selected analytics events and analyzed results across several games using Flurry and Google Analytics. Designed UX wireframes, iterated on feedback loops and balancing, and developed numerous prototypes at UNIT9. The archived CV discusses retention, conversion, acquisition cost and revenue metrics as product-design considerations.

analytics · Flurry Analytics · Google Analytics · UX · wireframing · prototyping · product design

## Open-source and personal software engineering · Independent projects

Selected projects

### Created ZzzOps, an open-source agentic engineering workflow

ZzzOps turns project policy, specifications, tasks and dependencies into a durable goal graph backed by GitHub Issues, and provides autonomous coding agents with verification, blocker handling and resumable execution. Its Python-based implementation supports Codex and Claude Code plugins and is published under Apache-2.0. Dave identifies it as his open-source work.

AI agents · agentic engineering · Python · GitHub API · developer tooling · workflow design · verification · open source

### Created HyperTween, a Unity ECS tweening framework

HyperTween is an MIT-licensed C# tweening framework built on Unity ECS, designed for high-performance transform updates, a fluent API, extensible outputs, complex sequencing, accurate timing, conflict detection and compatibility with managed Unity components. Dave describes its development as almost entirely traditional hands-on coding.

Unity ECS · C# · data-oriented design · Burst · Unity Job System · performance optimization · API design · animation · open source

### Explored simulation, numerical optimization and programmable UI

A personal project, Void Exiles, documented in the 2019 CV used Unity NetCode and HDRP, dynamically sized matrix and vector classes for bounded-variable least-squares thruster allocation, a hybrid GameObject view layer, XML-based customizable UI binding, and runtime C# compilation for player control macros. Its listing under a combined Bossa/ personal heading does not establish a commercial release or employer ownership.

C# · Unity ECS · NetCode · HDRP · least squares · numerical optimization · simulation · XML · runtime compilation

### Directed AI-assisted development of a multilingual recruiter assistant

Conceived and directed this career site through collaboration with Codex, from the semantic career-record concept to the minimal conversation interface. The Node.js/Express service uses Gemini through Vertex AI to answer recruiter questions and compare uploaded job descriptions against source-backed contributions, with separate system instructions, untrusted-input boundaries, structured outputs and validated citation IDs. Related project videos load automatically to thumbnails; PDF exports are assembled from selected records rather than generated claims. Deployment uses Google Cloud Run and service-account OAuth instead of a browser API key. Local IP-to-country lookup selects a default language; UI translations are generated on demand and cached by language and UI version in private Cloud Storage. Questions, answers, citations and documented gaps are retained privately for 90 days for service improvement; original files and full job descriptions are retained only by explicit opt-in. Append-only analytics storage is retrievable through authenticated owner tooling rather than a public dashboard. Document text is extracted by a separate IAM-protected Cloud Run service with its own unprivileged identity and bounded parsing. A shared atomic daily request budget persists across web-service restarts. Dave guided design iterations, factual corrections and feature priorities while Codex assisted implementation and verification.

AI engineering · LLM integration · Gemini · Vertex AI · Node.js · Express · Google Cloud Run · OAuth · structured outputs · PDF generation · internationalization · Google Cloud Storage · semantic data · privacy · Codex

## Project context

Project background from the listed sources.

### Parallel TCG

Parallel TCG is a science-fiction digital trading-card game from Parallel Studios. Players choose among five factions, build decks, and compete in strategic card battles, using units, abilities and banked cards to build combinations.

<https://www.youtube.com/watch?v=ddHMjFQO5aI>

Parallel Studios — Parallel TCG

### **Starship Troopers: Extermination**

Starship Troopers: Extermination is Offworld's cooperative first-person shooter set in the Starship Troopers universe. Teams of up to sixteen players fight Arachnids, complete objectives, gather resources and build defensive bases. This product context does not establish that Dave personally implemented each of those features.

<https://www.youtube.com/watch?v=E8FRZR728u8>

Offworld — Starship Troopers: Extermination

### **PAYDAY: Crime War**

PAYDAY: Crime War was a mobile first-person shooter developed by PopReach under licence from Starbreeze. It adapted the PAYDAY heisting setting for mobile, with cooperative heists and four-versus-four competitive matches, progression and character customization. It is a separate product from PAYDAY 2 and PAYDAY 3.

<https://www.youtube.com/watch?v=0dpk63di67Q>

Starbreeze — PAYDAY: Crime War

### **I Am Fish**

I Am Fish is a Bossa Studios physics-based adventure about four fish separated from their home in a pet-shop tank. Players guide the fish toward the ocean using swimming, flying, rolling and biting across environmental challenges. The product's emphasis on physical movement and water gives context to Dave's documented rendering and buoyancy work.

<https://www.youtube.com/watch?v=EqqTpOOEAK0>

Curve Games — I Am Fish

### **Surgeon Simulator 2**

Surgeon Simulator 2 is Bossa Studios' physics-based comedy surgery game, set in the Bossa Labs Medical Facility. Players perform deliberately chaotic operations while navigating environmental puzzles and a story campaign. It originally supported cooperative multiplayer and player-created levels; the current publisher's Steam listing says multiplayer was subsequently removed.

<https://www.youtube.com/watch?v=SwDL4aIN0rE>

Publisher-maintained Steam listing — Surgeon Simulator 2

### **Hogwash**

Hogwash is Bossa Studios' family-friendly asymmetric multiplayer game, developed for Apple Arcade. Three pigs try to spread mud across a farm while a farmhand attempts to keep them under control. Bossa describes the project as originating in a studio game jam. Dave's individual AI, navigation and design contributions are recorded separately.

<https://www.youtube.com/watch?v=nVQC3NNnTvo>

Bossa Studios interview — Making of Hogwash

### **Worlds Adrift**

Worlds Adrift was Bossa Studios' multiplayer sandbox MMO, featuring player-created floating islands, buildable airships, exploration and multiplayer physics. Its shared-world simulation used Improbable's SpatialOS. The game closed in 2019. This context explains the distributed simulation and ship-state synchronization problems in Dave's contribution records.

<https://www.youtube.com/watch?v=L9MFYOUTi5k>

Bossa Studios — Worlds Adrift community announcements

### **Hungry Shark Evolution**

Hungry Shark Evolution is a mobile aquatic action game from Future Games of London / Ubisoft. Players control a shark, eat sea creatures, explore the ocean and progress toward larger predators. Dave's documented work covers technical leadership, ongoing features, performance, cloud integrations and live operations; those contribution records establish his personal scope.

<https://www.youtube.com/watch?v=n84KBOCmI9w>

Ubisoft — Hungry Shark Evolution

### **Decromancer**

Decromancer is UNIT9's single-player tactical battle-card RPG. Players build an army of illustrated cards, arrange front-line and rear-guard formations on a five-by-four board, cast spells and explore a campaign world. The studio's case study credits Dave as Creative Director and Lead Programmer and discusses animation sequencing, observer-based UI updates and custom editing tools.

<https://www.youtube.com/watch?v=PQOjTHGgStw>

UNIT9 — Decromancer case study

### **Electric Tentacle**

Electric Tentacle is UNIT9's mobile high-score game and sequel to Nano Panda. Players create electrical bolts to pull miniature animal characters through obstacles, collect stars and rescue companions in an endless-level setting.

UNIT9 credits Dave with game design and lead development; his CV separately describes procedural level generation and pooling.

<https://www.youtube.com/watch?v=SAIVb795Gis>

UNIT9 — Electric Tentacle

### **Domino's Pizza Hero**

Domino's Pizza Hero is a UNIT9 and CP+B iPad experience combining pizza-making gameplay with ordering. Players stretch dough, add sauce, cheese and toppings, practice preparation challenges and compete on scores. The app connected virtual pizza creation to a real ordering flow, providing a commercial interaction-design context for Dave's geometry and scoring algorithms.

<https://www.youtube.com/watch?v=QzWEeOoFMfU>

UNIT9 — Domino's Pizza Hero

### **Nano Panda**

Nano Panda is UNIT9's physics puzzle game. Players launch small pandas toward targets, using the attraction between pandas to alter their trajectories, navigate hazards and collect stars. This gameplay context comes from a contemporary review; Dave's programming, gameplay and level-design responsibilities come from his archived CVs.

<https://www.youtube.com/watch?v=oPI-uUMehDM>

Common Sense Media — Nano Panda review

### **Xtreme Xrunch Kart**

Xtreme Xrunch Kart is UNIT9's Unity-based iPhone promotional game for baby carrots. Players steer a rocket-powered shopping cart using the accelerometer, while crunching real carrots supplies an unusual input for tricks. The agency's published credits list Dave as a developer; his archived CV also records gameplay and level-design work.

<https://vimeo.com/90246013>

UNIT9 — Xtreme Xrunch Kart

### **W.A.R.P.**

W.A.R.P. (Wayward Asteroid Redirection Program) is UNIT9's physics-based mobile game about defending Earth from asteroids by manipulating their paths using gravity. The official project page describes achievements and social score comparison through OpenFeint. Dave's archived CV records programming, game design and sound design. The historical CV lists the project in 2010; no release date is inferred from the current archive page.

<https://www.youtube.com/watch?v=WF91VMUinD0>

UNIT9 — W.A.R.P.

## **ZzzOps**

ZzzOps is Dave's open-source agentic engineering workflow. It gives autonomous coding agents durable goals backed by GitHub Issues, dependency tracking, verification and resumable execution, with Codex and Claude Code integrations. The implementation is Python-based and Apache-2.0 licensed.

<https://github.com/david-rzepa/zzzops>

ZzzOps — public repository

## **HyperTween**

HyperTween is Dave's MIT-licensed C# tweening framework built on Unity ECS. It offers a fluent API, extensible outputs, complex animation sequencing, accurate timing, conflict detection and support for managed Unity components. Dave says it was developed almost entirely through traditional hands-on coding; it is an example of his data-oriented Unity engineering.

<https://dave-rzepa.dev/hyper-tween/latest/>

HyperTween — documentation

## **Bossa data-science project (name not documented)**

An unnamed Bossa Studios research/development project documented in Dave's archived CVs. The available record lists a Python web service using Django, Celery, natural language processing, deep learning and statistical analysis. Its product name, client, datasets, model architectures and production outcomes are not documented.

## **Future Games of London incubation prototypes**

An internal incubation programme at Future Games of London / Ubisoft. Dave's archived CV records a rapid-prototyping framework, six distinct prototypes in twelve weeks, a more developed gameplay/frontend prototype and a deterministic-lockstep multiplayer prototype using Photon TrueSync. Individual prototype titles and commercial releases are not documented.

## **Void Exiles**

A personal multiplayer space-game and simulation project described in Dave's 2019 CV. Its recorded experiments include Unity NetCode and HDRP, least-squares thruster allocation, hybrid GameObject views, XML-configurable UI and runtime C# control macros. The record does not establish a public release or employer ownership.

## **Ask me about Dave Rzepa — AI career assistant**

This website replaces a fixed résumé with a recruiter-facing conversation over a comprehensive structured career record. Visitors can ask about Dave, explore specific projects or upload a job description for relevant experience, then download a source-backed PDF. Dave conceived and directed its AI-assisted development with Codex. The implementation uses Node.js, Express, Gemini through Vertex AI and Google Cloud Run, with automatic language selection, cached UI translations and project-linked video embeds.

<https://www.dave-rzepa.dev/ask>

Live career assistant

## **Education**

MEng in Computer Science · First Class Honours

Imperial College London

This document selects experience from Dave's owner-provided career record. Selection does not imply experience with every requirement of a particular job. Contact Dave to confirm project scope and availability.