

Tiogshi Laj

2346 Brackenwood Place, Nanaimo, BC, Canada | +1 250 734 3467 | tiogshi@gmail.com

True full-stack software developer. From application architecture design, hardware and virtual resource allocation, database engineering and schema design under relational or object storage paradigms, business rules design, API and human interface design, back-end services, software automation systems, front-end development, and embedded endpoint implementation; and all in addition to 3D modelling, 2D & 3D animation, and light VFX experience on top. His core competency is rapid self-education in new problem domains. His work over the last two decades still supports the safety training of petroleum extractors and refinery operators, the safety & compliance training of forestry and roadworks firms, and doctrinal training for first responders.

EXPERIENCE

Unreal Engine, Python, and ASP.NET Developer | Animated StoryBoards | 2023 to 2026

Worked in C++ on an Unreal Engine-based multi-player gymnasium-scale VR training system used for training first responders and law enforcement, working on issue resolution and on developing new features including procedural content generation for customizing end-users' VR environments.

Worked in C# on the ASP.NET-based multi-host applications which configure, launch, and stores recordings and analyses for the above VR training system.

Worked in C# and Python to add Blender support to the company's existing 3DS Max distributed rendering platform. Provided support to artists on moving their workflows from 3DS Max to Blender, including on topics such as rigging, lighting, VFX compositing, render optimization, and video production pipeline best practices.

Chief Technology Officer | Praxis Technical Group | 2015 to 2023

Instituted stable, change-managed, revision-controlled, data- and configuration-driven workflows for the assembly of artist-driven projects to enable high-quality turn-around of large, sometimes hours-long multimedia presentations.

Responsible for script review, editing, checking against primary sources, and client relationship management. Scripts covered topics including safety and regulatory compliance in the petroleum extraction, petroleum refining, fertilizer production, metal extraction and refining, and forestry industries, among others.

Responsible for software development, prototyping, and discovery projects, including developing online Learning Management Systems, developing Virtual Reality training content, and supporting other employees in developing HTML5 training courses.

Responsible for the development and maintenance of a wide variety of in-house software platforms, including distributed rendering and automation platforms, image- video- and data-transformation pipelines, bespoke CI/CD (continuous integration, continuous delivery) platforms responsible for product building and distribution.

Throughout, worked with numerous technologies, including but by no means limited to Python, Java, C++, C#, JavaScript, TypeScript, Lua, MySQL, Postgres, Azure web services and virtualization, Amazon Web Services and virtualization.

Trained artists on the use of Blender, as a replacement for Maya and 3DS Max, for 3D modelling, animation, and video production. Personally produced hours of animated training material, in pure 3D, and in integrating aerial photography with accurate models of client sites. Produced architectural visualizations in symbolic and photoreal styles for designing and previewing building renovations, both in-house and for clients.

Software Developer and IT Specialist | Praxis Technical Group | 2007 to 2015

Worked with Java on medium-fidelity training simulators for the chemical and refining industries, employing an entity-component framework adaptable to a wide variety of simulated processes, and to implement various user interfaces, from Java AWT-based emulations of Honeywell DCS/HIS, to remote rendering on the server in AWT forwarded for presentation in a web browser using Flash, to integrating the Java-based simulator through a C# based connector API to interact with it via the Yokogawa DCS Centum VP HIS application.

Ported those same Java-based simulations to Flash, and then later to JavaScript, so they could be integrated directly into browser-based training products instead of locally installed applications.

Provided IT support to various clients in the local area, including a law firm and construction firm. Deployed and administered Windows domains, deployed and maintained network storage and PBX systems, sourced and deployed new workstations for productivity and CAD workloads, provided support and training for Microsoft Exchange, SharePoint, and Office suite.

EDUCATION

Bachelor of Science, Computer Science | Vancouver Island University | Nanaimo, BC, Canada | 2001 to 2006

TECHNOLOGY EXPERIENCE

C++	C#	Java	Lua	Python	
JavaScript	TypeScript	Smalltalk	Ruby	O'Caml	
PhotoShop	Krita	GIMP	Blender	Unreal	Godot
MySQL	MS-SQL	Postgres	SQLite	LiteDB	Redis
Arduino IDE	Pico SDK	KiCAD	FreeCAD	OnShape	
Linux shell	Visual Studio	VS Code	Eclipse	JetBrains	
Google suite	OpenOffice suite	Microsoft Office	Active Directory	Exchange	
Azure	AWS	Colocation	Self-hosting		

SKILLS

Application Design	Business rules	API integration	Realtime operations
Full-stack development	Backend services	Toolchain development	Front-end development
3D modelling (props, architectural visualization, photogrammetry, cleanup, optimization, rigging)			
2D & 3D Animation	Video compositing	Video editing	VFX
Realtime graphics	Photoreal rendering	NPR rendering	
Embedded systems	Small network platforms	Web-scale platforms	

STANDARD OF ETHICS

At this time, practically all publicly available products marketed as “AI” are built on unacknowledged and uncompensated theft of labour, and a noteworthy sector of the market has for years existed solely to wash their users’ hands of responsibility for the processes they have deferred to those products. *I will not touch them.*

I am open to and approving of the use and creation of all machine learning processes and tools based exclusively on ethically produced and compensated training data, and to uses where auditable human decisions stand exclusively and unarguably responsible for their output.

SELECTED PROJECT SYNOPSES

Personal Works | Ongoing | <https://cara.app/tiogshi/portfolio>

Various 3D modelling, rigging, and animation. Personal practice and meditative-art projects.

3D Animations for LNGC | Praxis Technical Group | 2020 to 2022 | <https://youtu.be/qoBacUpdn3c>

This video is a selection of shots I worked on for the training material of an LNG refinery in Kitimat, BC, Canada.

The models were provided by the engineering firm responsible for building the refinery, but were provided as a raw polygon soup, with few to no materials, metadata, or differentiation between separate logical objects; for instance, all plant-wide piping of all sizes, purposes, and physical materials – from 10mm office fire suppression systems to a 1,250mm river water intake – were delivered as a single mesh with over 500 million vertices.

My tasks were to identify objects from the diagrams, review the scripts for relevant machinery and locations, extract and separate the required components from the raw models, add materials, light and compose the various visualization layers, and then animate labels, camera angles, and objects synchronized to a provided narration audio track, with editing of narration timing permitted but discouraged.

Learning Management System | Praxis Technical Group | ~2022

In addition to producing training courseware, one of Praxis' other revenue streams was hosting Learning Management Systems for their small and mid-size clients – those with a need for serving anywhere up to 2,000 student-courses per year. Larger clients generally already have an LMS product sold to them by their existing Business Information System vendor or Enterprise Asset Lifecycle Management vendor.

In pursuit of reducing Praxis' overhead in deploying and maintaining dozens of separately configured, separately customized LMS using turnkey products such as Moodle, I sought to implement an in-house LMS platform aimed to address several shortcomings in Moodle, VTA, SuccessFactors, SCORM Cloud, and Articulate's offerings. We also sought a design that would give users the option to self-host in case they had a regulatory requirement to do so.

With the intent to support small self-hosted private LMS (up to tens of simultaneous users) as well as web-scale cloud offerings (ideally up to tens of thousands of simultaneous users across our entire client base), I designed and implemented a scalable architecture based on a microservice model with optional federation of components. All the microservices required for a single "federation" could be co-resident on a single host; or each microservice could exist on at least one separate host; or any combination in-between, to suit expected or actual system loads. Static content for a given federation instance could be served locally or from a designated CDN.

The implementation was in TypeScript, compiled to JavaScript and executed on NodeJS, with nginx used in-house but the use of other HTTP servers was possible. The front-end was implemented as an SPA in TypeScript and using React for client-side rendering. The database could be SQLite for single-host configurations or Postgres for federated systems, with migration possible between the two. Distribution both in-house and for self-hosting clients was intended to be via publishing Docker images to which configuration and licensing would be attached at instancing time.

Several web-based training communication standards were supported by the LMS, including SCORM 1.2, SCORM 2004, AICC, xAPI, and a proprietary in-house protocol designed to address the fact that most of those others listed were designed in an era where JavaScript-initiated synchronous HTTP requests were considered standard. By the mid '10s, however, most browsers silently fail when synchronous requests are made or silently convert them to asynchronous ones, making reliable recording of progress through a training module difficult-to-impossible in unreliable network environments, especially those of mobile devices.

3D Animations for DPW | Praxis Technical Group | 2023 | <https://youtu.be/KwZcFkLM9wE>

This is a prototype-grade mockup, produced in just 12 hours, as a proof of concept during the proposal phase of a project for DPW, a firm owning multiple container ports worldwide.

My task was to take some aerial photography of the site, and from it mock up a 3D model of the port. I used pre-made models for trucks, vessels, and cranes, but had to create bespoke materials for all of them, in addition to modelling the containers and port grounds.

After this proposal, I proceeded to make a similar type of aerial-reconstruction 3D model for four of their ports, aiming for accurate enough models to be recognizable from ground level.

Mobile home-made gaming console | Personal Project | Ongoing

I started a project with my daughter – at the time, 7 years old – to teach her soldering, give her an outlet and motivation for her art, and get her engaged with programming and making in general.

The current iteration of her home-made mobile gaming console has a backlit 160x128 pixel 16-bit colour screen, driven over SPI by a Raspberry Pi Pico 2350 dev board mounted to a third generation of custom PCB, and a headphone jack for audio output. It runs on 3x AA batteries for over 30 hours.

There are two finished games and several tech demos, all using music and sound effects by myself, and a mixture of art assets drawn by myself, by my daughter, and some Creative Commons art. All the games and tech demos are written with a Hardware Abstraction Layer I wrote from scratch, that lets me build the games and demos for Windows, Mac, or Linux using SDL2, or build it for the Pi Pico using the Pico SDK.

The HAL provides a feature-rich engine for games to use. It has a two-channel PWM-driven square wave audio generator for music and sound; approximate in fidelity to “PC Speaker” audio of the 80s and 90s, but with two voices instead of one. It can render bitmaps, opaque or with 1-bit alpha channel support and with vertical/horizontal flipping, optimized for rendering directly from ROM to save RAM utilization. It can fill arbitrary convex 2D polygons with solid colours or with scrollable, scalable 2-colour bitmap patterns. It can rasterize 3D triangles with a per-triangle solid fill, with or without a Z-buffer and using separately-optimized pipelines for each. Bitmaps and 3D models are loaded from ROM to RAM using a managed asset pipeline and the C++ RAI code paradigm, handling decompression and garbage collection where applicable automatically as games move from one state to another.