

# § Coranna Howard

A programmer and crafter from the Seattle, Washington area, Coranna is skilled in technical writing, vocational teaching, data architecture & reverse-engineering, modern computer graphics systems, software repackaging, and the design & implementation of low-level APIs.

## § professional work

¶ [Pierce College](#) — Lakewood, WA, U.S.A.

» **2017–present** — Clubhouse Specialist

Supervised students and maintained tech in an Intel Computer Clubhouse at Lochburn Middle School.

¶ self-employed — Tacoma, WA, U.S.A.

» **2014–present** — personal, gigs

Developing [Quanta](#), [togo](#), and Attendance Reporter, tackling gigs in a medley of fields, helping fellow programmers, running amok with far too many crafts, opinionating on tech culture, and learning new sub-fields.

¶ [Austin Powder](#) — Cleveland, OH, U.S.A.

» **2011–2014** — software repackaging

Architect of packaging methodology & repository of 675+ packages (approximately 211 products across 75 vendors) for a network of 850 Windows computers. Documented methodology & structure and taught the art of repackaging. Wrote utilities for IT in VC++, JScript, and CMD.

» **2008–2009** — IT

Three small contracts for hardware and software deployments. Built cross-platform software to track assignment & progress of deployments and to allow the project manager to securely share user info with the technician.

» **~2005–2008** — IT

Numerous small jobs to keep the gears turning: utility software (in C#, C++, Visual Basic 6.0 + .NET, BlitzMax, JScript/WSH, and Windows CMD), software & hardware deployment, documentation, and user support.

# § www

**email** [me@komiga.com](mailto:me@komiga.com)

**github** [github.com/komiga](https://github.com/komiga)

**website** [komiga.com / resume](https://komiga.com/resume)

**updated** 2018-01-29

## § skills

C, C++, C#	CMake, Premake
Lua, Python	Linux, POSIX
Java, HTML	Windows, WinAPI
Clang, GCC	Visual Studio
Bash, CMD	Excel, GSheets
OpenGL, LOVE	Blender, GIMP
GLFW, SDL	MSI, InstallShield
git, GNU Make	Android SDK

## § volunteer work

¶ Linux videogame QA

» **'12–'14** — Humble Inc., 17 games  
» **'12** — Santa Ragione, [Fotonica](#)  
» **'12** — Subsoap, [Faerie Solitaire](#)

¶ scientific research

» **'15–present** — HCI thesis on trans people and speech training (*identity withheld to prevent deanonymization*)

## § background

» **'13–'14** — [Algorithms, Part II](#)  
» **'13 × 2** — [Algorithms, Part I](#)  
» **'11** — *Repackaging and Application Migration using AdminStudio 9.5*  
» **post sec., CS** — autodidact  
» **K-12** — autodidact / homeschool  
» **English** — C1-C2, U.S.–U.K. hybrid  
» **French** — A1, Standard/Metro.  
» **Japanese** — sub-A1, Eastern

## § portfolio

- » '16 — [sounds for contemplating the universe](#), a soundtrack for stargazing (*Fermi Paradox Jam*)
- » '14 — [Kaleidograph](#) (*stills*), an interactive generative art program (*JavaScript, p5.js*)
- » '14 — [Onomo](#), a slow, dark platformer concept (*Ludum Dare 30, Lua, LOVE, 34.7h*)
- » '13 — [Prisma](#), a color-based twitch puzzler (*Ludum Dare 26, Lua, LOVE, 35.1h*)

## § code

- » [togo](#), app & game super-library (*C++, Lua*)
  - Data-oriented design, open types, open interfaces.
  - Digestible alternative to the C++ Standard Library.
  - Game engine (WIP) with pipeline tooling.
  - Imaging and windowing.
- » [Quanta](#), adaptive tracking toolkit (*C++, Lua*)
  - Expressive time and nutrition tracker, WIP Android companion.
  - Universal description language (English read- & write-able).
  - Extendable, rapid-iteration tooling and data analytics with Lua.
  - Linux integration (CLI tools, data vessels).
- » [Attendance Reporter](#) (*JS, Go*)
  - Developed for Intel Computer Clubhouses and other publicly-funded organizations.
- » [Pickle](#), static site generator (*C++, Lua*)
  - Non-dogmatic; user controls structure.
  - Lua-based template language and userspace.
  - Bare-bones web server for rapid iteration.
- » [precore](#), Premake 4.4 extension (*Lua*)
  - Modularity & reusability layer atop Premake.
- » [igen](#), C++ interface generator (*Python*)
  - Generates function declarations (preserving docs) from their implementations using libclang.
- » [include\\_sort](#), C & C++ #include sorter (*Lua*)
  - Sorts #include statements in user-defined order.

## § contributions

- » [mooege](#),<sup>†‡</sup> *Diablo III* server (*C#, '11*)
  - Networking (Battle.net, game layer), game world, Linux support.
  - Asset RE, packet RE, Protocol Buffer implementations, documentation.
  - Designed prospective production-grade server architecture.
  - Later: PR arbiter/project manager, working with many important contributors.
  - Co-architect alongside Hüseyin Uslu.
- » [ParkPoints](#), gamified park participation app (for the *Parks and People United Through Technology* hackathon by Metro Parks Tacoma, '17)
  - Joint with Andrew Dickinson, Grace Bergman, Jasmine Scott, Krystaal McClain, and Robin Choi.
- » [spirv](#),<sup>†</sup> binary SPIR-V codec (*Go, '15*)
  - Implementation of the provisional specification.
  - Contributed fixes upstream (to Khronos).
  - Co-architect alongside Jim Teeuwen.
- » [Maximus](#),<sup>†</sup> module manager (*BlitzMax, '10*)
  - Core architecture and command-line client.
  - Joint project with Christiaan Kras, who maintained GUI client & web service.
- » [Pygments](#) (*Python*)
- » [libc++](#) (*C++*)
- » [GLM](#) (*C++*)
- » [gltext](#) (*Go*)
- » [SPIR-V Specification \(provisional\)](#)
- » [Golang Specification](#)
- » [Elixir Getting Started tutorial](#)

<sup>†</sup> — Defunct.    <sup>‡</sup> — [Commit history](#).

## § familiarity

Supplementary in-depth look at Coranna's knowledge.

### ¶ concepts & domains

- **API & system design** — extensive: very many API projects, strong attention to detail, strong architectural cohesion mindset
- **algorithms & data structures / abstract data types** — extensive: many uses of non-trivial structures & algorithms; task distribution, hashing, PRNGs, graphs, tries, hash tables, priority queues, stacks, radix sort, merge sort, quick sort, LZW, Huffman coding, RLE; basic complexity analysis; intermediate-level education, strong intuition
- **data storage** — proficient: very strong understanding of binary & text formats, proficient in non-trivial serialization
- **language, parsing, pattern matching** — extensive: a universal description language, a Quake-like, and a JSON-like; multiple uses of Lua as a DSL
- **reverse-engineering** — extensive: data & algorithms for a handful of games, several cryptographic breaks
- **game systems & design** — wide: several successful game jams, several small experiments, several game emulation projects; strong intuition
- **networking** — wide: several game server emulation projects, experiments
- **graphics** — narrow: 2D games & experiments; strong intuition, embarking on 3D
- **mathematics** — narrow: algebra, functional linear algebra, basic calculus, basic set theory, basic logic, extensive notation; strong intuition
- **cryptography & security** — narrow: daily PGP use, basic practical implementation – I won't store passwords unsalted, or use SHA1, or send your boss's (nor their boss's) plaintext credentials to IT
- **time tracking** — proficient: daily, very detailed (beyond work), personal software

### ¶ programming languages

*Standard library familiarity is equivalent to language familiarity herein.*

- **C++ (03–11)** — proficient: very many projects over very many domains
- **Lua (5.0–5.2, LuaJIT)** — proficient: many projects over many domains
- **C (89–11)** — extensive
- **C# (2.0–5.0, .NET, Mono)** — wide: large emulation project, several apps, professional
- **Python (2–3)** — wide: large Blender automation project, several utilities
- **Go (0.x–1.x)** — narrow: community projects, some language spec fixes
- **HTML (4–5)** — wide
- **CSS (2.1–3)** — narrow
- **JavaScript / ECMAScript (4–6)** — narrow: several websites, professional
- **Java (SE 6)** — wide: coursework, projects
- **TypeScript** — narrow: experiments
- **BlitzMax** — proficient: many projects, professional
- **Bash** — extensive: daily
- **Windows CMD** — extensive: professional
- **Assembly (Intel, x86)** — narrow
- **Erlang** — minimal
- **Elixir** — minimal
- **Scheme** — minimal
- **Lisp** — minimal

## ¶ toolkits

- **POSIX** — extensive
- **Windows API** (Win32/WinAPI) — narrow: professional
- **Android SDK** — narrow
- **OpenGL** (2.1–3.3) — wide: rendering pipelines, game emulation, extensive architectural knowledge
- **Vulkan** — narrow: specification fixes, SPIR-V tooling, some architectural knowledge
- **DirectX** — minimal
- **SDL** (1.x–2.x) — extensive
- **GLFW** (2.x–3.x) — extensive
- **LÖVE** — extensive
- **Stingray / bitsquid** — narrow: extensive architectural knowledge, no direct use
- **Quake-like engines** — minimal: modding & mapping for *Half-Life 2: Deathmatch* (Hammer) and early *Medal of Honor* games
- **GameBryo** — minimal: modding *The Elder Scrolls IV: Oblivion*
- **GameMaker** — narrow: dabbling, professional
- **Unity, Unreal, XNA / FNA / MonoGame** — minimal: basic architectural knowledge through QA & debugging, no direct use
- **LLVM** — narrow: tooling
- **p5.js** — narrow: generative art
- **node.js** — narrow
- **React** — minimal

## ¶ software

- **Linux** (*Ubuntu, Debian*) — proficient: daily
- **git** — extensive: daily
- **Clang** — extensive: daily
- **GCC** — wide
- **Premake** (4.x) — proficient: daily
- **GNU Make** — wide
- **Sublime Text** (2) — proficient: daily
- **vim** — narrow: daily
- **IDA** (5.x) — narrow
- **Visual Studio** — narrow: professional (I avoid IDEs)
- **Blender** — narrow: general use, extensive automation (scripting)
- **Microsoft Excel** (2010) — extensive: professional
- **Google Sheets** — proficient: professional
- **InstallShield** (17–21) — proficient: professional
- **Windows Installer (MSI)** (3.0–5.0) — extensive: professional
- **AdminStudio** — extensive: professional
- **Windows** (XP, 7–8.1) — extensive

## ¶ spoken language

- **English** (*U.S.–U.K. hybrid; U.S. dominant*) — proficient (C1–C2): first language
- **French** (*Standard/Metropolitan*) — narrow (A1): actively learning
- **Japanese** (*Eastern/Tokyo*) — minimal (*sub-A1*): pronunciation, kana, few kanji, few phrases

