Mark Kaldas

+1 (403) 805-2962 ● mark_kaldas@yahoo.com ● markpwns1.github.io ● linkedin.com/in/mark-kaldas-ca

EDUCATION

- B.Sc. Computer Science at the University of Calgary

LANGUAGES

C# Lua Python Java
Rust C++ PHP Javascript
Typescript HTML CSS
SASS GLSL SQL Delphi
Haskell OCaml Prolog
x86, MIPS, 6502 Assembly
VB.NET French

TECHNOLOGIES & TOOLS

.NET Framework .NET Core
Unity3D Node REST React
Wordpress Laravel MongoDB
jQuery SQL WebGL POSIX
WinAPI Windows Forms MVC
Love2D SFML JUnit (and unit testing in general) Git (including Github & Gitlab) Jira PL/SQL

SKILLS

- Desktop, web & game development
- UI/UX design & implementation
- RESTful API design & implementation
- Graphics & shader programming
- Procedural generation
- Technical art

PROFESSIONAL EXPERIENCE

- Information Technology and Support — The Children's Link Society — 2020 - 2023
    - This role was a combination of sysadmin and full-stack developer (I was the whole IT department). Also the primary contact for any software firms or IT consultancy firms.
    - Maintained and modified their website (Wordpress & PHP) and public-facing web apps written in React-Redux & SASS for frontend, PHP & Laravel backend, and a MySQL database. Resulted in major usability improvements and much fewer customer calls to tech support as a result.
    - Wrote an in-house tool for managing clients using React, NodeJS, and MongoDB. Resulted in immense productivity improvements over the unscripted excel spreadsheets they were using.
    - Wrote a tool in jQuery and Node for parsing Google Forms results and aggregating, generating, and exporting statistics, automating a large portion of my job and freeing me for other tasks.
    - Quickly investigated, diagnosed, and fixed a number of sudden website outages, minimising downtime.

ADDITIONAL EXPERIENCE

- One of four leaders of the University of Calgary Game Design Club — 2020 - 2023
    - Giving advice and technical help to club members, running game jams and other events, and giving programming and technical lectures.
- Many team-based game jams (like hackathons but for making video games)
    - Close coordination with 3 to 5 teammates and using coordination tools like version control, Trello, etc. to get the product delivered in a tight deadline (usually 48 hours).

NOTABLE PROJECTS

- Solo developer of the short game *NAFFTA: SET THE STEPPES ALIGHT*
  - Made in Unity, C# & GLSL using a variety of advanced graphical techniques for a unique look.
  - Well received and offered a publishing deal and funds to turn it into a full-length game
  - An infinite, procedurally generated world with a render distance of over 3 km (absurdly large)
  - Heavy use of procedural animations, incorporating inverse kinematics and tweening
  - All textures were procedurally generated by a custom tool written in WebGL, jQuery, and GLSL
  - https://botmark.itch.io/naffta (this is the original version, not the commercial release)
  - Many more games on my itch.io, as well as unpublished on my hard drive

- Author of the Jammy programming language
  - A highly sophisticated optimising compiler written in Javascript targeting Lua
  - Language is a hybrid between functional and OOP with a batteries-included standard library
  - Specially designed for quick iteration speed (for making games quickly for game jams)
  - Excellent compile-time error reporting
  - https://github.com/markpwns1/jammy

- A custom image board client for mobile, aggregating the results of 3 sites and implementing their premium-only features myself, to make available to everyone
  - *Vast* improvement over the first-party sites in terms of feature set, performance, and UX
  - Written in plain jQuery, HTML, and CSS, because lightweight sites offer better battery life
  - Reverse-engineered some of their systems, created novel implementations for others
  - https://booring.herokuapp.com/?domain=danbooru&q=rating:g,landscape

- A single-page flashcard webapp for memorising vocabulary in the *Genki* textbooks
  - Written in React, and the flashcard data was obtained by web-scraping study sites
  - Offers a streamlined user experience, and is very good at what it does
  - https://markpwns1.github.io/genki-crammer

- A hyper-optimised raycaster engine (think *Wolfenstein 3D*) as an exercise in optimisation
  - Written in C++, SFML, and GLSL, offloading much of the work to the GPU.
  - Level format designed for highly efficient ray-wall intersection checks
  - Decoupled rendering and update timesteps
  - Blazingly fast. Reaches a quadruple digit framerate on even modest devices
  - https://github.com/markpwns1/raycaster

- Countless others that won't fit on this resume! github.com/markpwns1 & markpwns1.github.io

Note that none of these were school or work projects! I made them for fun.
Please feel free to ask about anything you've read!