David Alexander Stuart

San Francisco, CA, https://das.li, alex@das.li, (612) 619-1039

EDUCATION

Master of Science in Computer Science — Salt Lake City, UT
University of Utah — 2013

- Committee: Adam Bargteil (chair), Joshua Levine, Cem Yuksel
- Thesis: **Coarse Tetrahedral Meshing for Interactive Simulation**

Bachelor of Science in Computer Science — Minneapolis, MN
Bachelor of Science in Mathematics — 2010
University of Minnesota

PROFESSIONAL EXPERIENCE

Developer Tools Engineer — Oct. 2021—Dec. 2022
Groq

I developed GroqView, a visualization tool for debugging and profiling Groq's ML accelerator chips.

- Implemented optimizations like streaming and binary data formats in our Haskell server and Elm client
- Created new interactive visualizations, such as power-use plots and multi-chip dataflow diagrams
- Met with users to understand needs, mentored interns

Senior Software Engineer — Mar. 2021—July 2021
Lumi

I and a distributed, remote-first team developed a web-based marketplace using functional programming.

- Enabled Haskell Language Server IDE features on 100,000-line Haskell codebase
- Wrote and optimized typed SQL queries in Haskell with Esqueleto
- Refactored sophisticated form validation library in PureScript based on optics and Generic instances

Personal sabbatical — May 2019—Feb. 2021

During my sabbatical I explored functional programming in the context of game development.

- Released *Peoplemon*, the first commercial role-playing game made entirely in Haskell
- Abstracted engine from *Peoplemon* code, added 3D rendering
- Published research paper at Haskell Symposium 2020 about DSLs for functional reactive programming

Tutor — Sept. 2019—May 2020
Brighter Minds SF

I taught algebra, geometry, and calculus to middle and high schoolers in one-on-one sessions.

- Guided students in doing homework, developed enrichment activities
- Helped students improve their grades and outlook on math

Software Engineer Nov. 2017—May 2019
Machine Zone

I and a core R&D team led development of *Crystalborne*, MZ's first fully 3D mobile game.

- Implemented animated camera and controllers, billboard rendering, tool interfaces using C++
- Developed tutorials, UI overlays, animation sequences, modular geometry using Lua
- Worked closely with art director and artists to plan workflows

Senior Software Engineer Dec. 2016—Nov. 2017
Fathom

I led a team that built and operated a web-based quoting tool for 3D printing using JavaScript and MongoDB.

- Implemented interactive 3D rendering, interface to geometric analysis service
- Established continuous testing infrastructure and practice
- Mentored junior engineers in tests, functional programming, professional development

Guitar Player June 2016—present
Larry and the Millennial Falcons

I am a founding member of a band that puts on live-band karaoke shows.

- Learned, arranged, and performed over 200 songs; adapted to unfamiliar singers
- Bimonthly club show with two 2-hour sets

Senior Software Engineer Sept. 2013—April 2016
Autodesk

I and an Academy Award-winning team developed Bifrost, a platform in Maya for physical simulation.

- Created Bifrost's first modeling tools for implicit surfaces using C++
- Fully remote member of a globally distributed team
- Resolved customer escalation in my first week

Before my time on Bifrost I developed Tinkercad, a web-based 3D modeling tool built on microservices.

- Implemented geometric modeling operations in JavaScript
- Added full-stack privacy features for COPPA compliance using JavaScript and Go
- Prototyped novel texture mapping scheme for our BSP-tree geometry representations

Graduate Associate June 2012—June 2013
Disney Interactive

I developed the AI system used in *Disney Infinity*.

- Optimized game engine's implementation of behavior trees in C++
- Created new gameplay features as extensions of AI system

Teaching Assistant Sept. 2010—May 2012
School of Computing, University of Utah

I helped run courses in computer architecture and scientific computing.

- Held weekly study sessions, assisted students one-on-one
- Corrected homework and exams

Research Assistant July 2011—Sept. 2011
Disney Research Zurich

For three months I investigated simulation and control problems.

- Devised geometric models of magnetic forces on elastic bodies
- Built and measured physical apparatus for evaluating models

Research Assistant Feb. 2009—Aug. 2010
GroupLens Research, University of Minnesota

I studied the social effects of changes to Wikipedia's user interface.

- Wrote Python scripts to scrape Wikipedia and compile databases
- Modeled sentiment in user interactions, organized human coding of hundreds of interactions

Teaching Assistant Sept. 2008—May 2009
UMTYMP, University of Minnesota

I taught calculus to a class of 40 talented middle-school students.

- Directed 30 study sessions, held call-in office hours
- Wrote 900 detailed homework corrections

TOOLS

Programming languages:

- *Good with* Haskell, Elm, C, C++, JavaScript, Lua, Python
- *Some experience with* C#, Go, Java, PureScript

Development environments:

- Git, Haskell Language Server, Perforce, Vim, Visual Studio (Code), Xcode

Libraries/systems:

- Esqueleto, Linux, macOS, Maya, OpenGL, PostgreSQL, Servant, SDL, TBB, Unity

PUBLICATIONS

David A. Stuart
Scripted Signal Functions
ACM SIGPLAN International Symposium on Haskell, Virtual Event, August 2020

David A. Stuart, Joshua A. Levine, Ben Jones, Adam W. Bargteil
Automatic Construction of Coarse, High-Quality Tetrahedralizations that Enclose and Approximate Surfaces for Animation
ACM SIGGRAPH Conference on Motion in Games, Dublin, Ireland, November 2013

Aaron Halfaker, Bryan Song, **D. Alex Stuart**, Aniket Kittur, John Riedl
NICE: Social Translucence Through UI Intervention
International Symposium on Wikis and Open Collaboration, Mountain View, California, October 2011

SOLO PROJECTS

Lightarrow/FairyBow

- Interactive 3D graphics in Haskell abstracted over backend (currently GPipe)
- Batched mesh and sprite rendering, animated scene graph
- Optimizations to control allocations and garbage collection pauses

Peoplemon

- Implemented salient features of Pokémon Generation I using functional reactive programming
- Commercially released to over 4200 players on macOS/Linux/Windows
- Made from scratch with Haskell and SDL, all original assets

Scheme game programming system

- C extensions to Scheme for quickly writing correct, fast game programs
- Integrates manual and garbage-collected memory allocations
- Efficiently marshals data between language implementations

Vuess Vow Vong Vo Vold Vown Vhe Vey

- Game and engine made from scratch in C++, for Linux and Windows
- Text rendering, frame-accurate sound looping, animated sprites, twenty-two levels