Andy Hanson

andy-hanson.com andy-hanson@protonmail.com (703) 402-2284

work

Microsoft , Redmond, WA *Software Engineer II* May 2016-present

Resolved issues in the TypeScript compiler and language services such as go-to-definition, find-all-references, completions, code fixes, and suggestions. Fixed bugs and updated services to work in more situations and with new language features.
Wrote types-publisher, which publishes the type definition repository DefinitelyTyped to NPM, and dtslint, which tests type definitions and standardizes the structure of definition packages.
Worked on the .NET core garbage collector. Wrote a program to run tests, collect traces, analyze the trace in-depth, collect metrics, and report metric values as tables and graphs.
Used TypeScript, C++, C#, and Python.

FaunaDB , San Francisco, CA *Associate Software Engineer* Jun 2015-Jan 2016

Helped write libraries for using a new database for Python, Ruby, JavaScript, and C#. Each included a REST client, error handling, (de)serialization, query language, and ORM.

IntentSoft , Bellevue, WA *Software Analyst and Developer* Jul 2014-Mar 2015

Worked on a content layout system in C#which took as input a tree of content nodes (analogous to DOM nodes) and output a tree of graphical nodes (which have exact locations).
Rewrote part of system that queried content locations to be closer to purely functional.

school

Rensselaer Polytechnic Institute , Troy, NY GPA 3.94/4.0 2012-2014

Studied Computer Science. Included: parallel programming, computational physics, language-endowed intelligent agents, advanced computer graphics, intermediate logic.

Digipen Institute of Technology , Redmond, WA GPA 3.49/4.0 2011-2012

Responsible for programming, sound, and some graphics on a team of 3. Used ProjectFUN.

code

noze programming language `github.com/andy-hanson/noze` 2018-

Statically-typed programming language compiling to C. Features: modules, records, (discriminated) unions, (asynchronous) interfaces, lambdas, templates, overloading, type argument inference. Has "specs", a feature for calling functions that aren't available yet. Runtime can be written in noze itself. Compiler can evaluate constants and specialize functions on constants and known lambda bodies.

tslint linter 2016-2018

Contributed many new lint rules to this project and made older rules simpler and more performant.

mason programming language 2014-2016

Made a dynamic language compiling to JavaScript. Featured dynamic type checking, collection builders, async for, pattern matching, polymorphic functions, traits, a standard library with dynamically generated types (such as records and predicate types). Wrote several web games in mason.

dum build system 2015-2016

Made a build system that transforms file-by-file using jstransformers, no build script or plugins necessary.

painterly rendering WebGL rendering experiment spring 2014

Team of 2. Real-time WebGL renders a 3D particle system of brush stroke sprites which orient and curve along edges of objects and around centers of light using a custom shader.

logic online online logic practice spring 2014

Created a deduction language. Inference and equality rules are asserted or proved within the language.

polo jeans WebGL 2D game engine fall 2013

Lead a team of 4. Ran meetings and designed architecture, game logic, and assets. Game was pong.

LEIA natural language dependency grammar parser fall 2013

Wrote parser on team of 6. Users write English sentences to ask for information about local food.

ghost control control systems research video game summer 2013

Working for control systems researchers, created a web game that saves/loads ghost data from server.

wahoo Scala 2D game engine fall 2012

Created a game engine on top of LWJGL. Features: dynamic lighting, loose quadtree collision detection, and type-based (via reflection), state-based GameObject scripting.

games 2009-2013

Web games and desktop games. Many 2D PyGame games, and one game in OpenGL and one in OGRE, both with PyGame as a base. Unfinished game written with Wahoo.