Will Gueble

(206) 327-2981 | will.gueble@gmail.com
www.willgueble.com

Education

Wesleyan University | Middletown, CT | Sept. 2011 – May 2015

- B.A. in Computer Science and Philosophy, May 2015. GPA: 3.5
- Selected Undergraduate Coursework: Computer Graphics, Algorithms, Automata Theory, Computer Structure, Data Structures, Vectors & Matrices, Writing About Video Games

Professional Experience

Software Engineer | SRC Inc. | Dec. 2016 – Jan. 2017

- Wrote bit-buffer translators to handle several new modes of communication between an embedded hardware platform and its connected software GUI applications.
- Implemented numerous display features across multiple software GUI projects.
- Developed templated de/serialization algorithms for parsing both YAML and XML files.

QA Tester | Undead Labs | Summers 2014, 2015

- Helped to implement several new features for a load-testing tool:
  - Using timers, changed the tool to run continuously on an asynchronous thread.
  - Enhance the tool to generate virtual interface methods using delegates and programmatic reflection, saving development time whenever new methods are added.
- Received training as part of a QA automation class for developing shell scripting fundamentals.
- Wrote comprehensive test cases and documented bugs with exact reproduction steps.

Back End Developer, Intern | Viget Labs | Summer 2013

- Built a Hacker News clone (news.ycombinator.com) from the ground up using Ruby on Rails.
- Worked with UX and Front End Development Interns on a collaborative Ruby on Rails project.

Instructor, Tutor, & Course Assistant | Wesleyan University | 2012 – 2015

- Designed and co-taught a full credit Comp. Sci. course on web development (2015).
- Worked as a course assistant and private tutor teaching Java to beginning CS students

Technical Experience

Projects

- Developed an original board game using SDL2, with multiplayer functionality using net sockets.
- Created a working prototype for a planned 2D platformer, complete with SceneGraph space partitioning, collision detection and resolution by projection, GLSL vertex and fragment shaders, and a hybrid OpenGL 3.3 and SDL2 approach to handling input and window creation.
- Built a 3D ray-tracer from the ground up as part of the final evaluation for my undergraduate Computer Graphics course.

Languages and Technologies

- Languages: C++ ('11/'14), C#, Java, UNIX, make, GLSL, C ('99) HTML, CSS, Ruby/Rails, YAML, XML
- Technologies: OpenGL (2.1/ 3.3+), Git, Perforce, JIRA, Unity, Unreal, SDL2, Cygwin, Jenkins