

Brett Schiff

<https://www.linkedin.com/in/brett-schiff>

brettschiff@gmail.com

636-233-0949

Software Developer

Languages:	Proficiency: C++ C	Familiarity: Lua C#
Skills and Software:	Visual Studio Git GNU Compiler SVN Doxygen Oracle VM VirtualBox Windows	JavaScript Clang HTML Agile Development Linux VirtualBox
Academic Projects:	<div><div>Shortstack(team of 6)</div><div>Physics Programmer</div><div>August 2017 - Present</div></div> <p>Shortstack is a 2D, local co-op, side-scrolling platformer where players are gnomes who stack on each other to combine abilities built in a custom engine made in C++.</p> <ul style="list-style-type: none">Designed and implemented the physics system in C++<ul style="list-style-type: none">Separating Axis Theorem(SAT) for collision detectionUtilities: raycasting, debug hitbox display, point collision for particlesWorked with ImGui to incorporate physics into the game's editorContributed to gameplay programming in Lua<ul style="list-style-type: none">Enemy AI and player movement <div><div>Relics of Light(team of 4)</div><div>Physics Programmer</div><div>August 2016 - May 2017</div></div> <p>Relics of Light is a top-down 2D puzzle-adventure game featuring open-world exploration, puzzle challenges, and unlockable character abilities built in a custom engine made in C.</p> <ul style="list-style-type: none">Created the physics system in C using SAT for collision detectionDeveloped input system using a wrapper around GLFW<ul style="list-style-type: none">Captured and processed input from Windows for scroll-wheel inputDesigned the overworld and all underground puzzles in Tiled	
Education:	<div><div>DigiPen Institute of Technology</div><div>Graduation May 2020</div></div> <p>BS in Computer Science and Real Time Interactive Simulation</p>	
Personal Projects:	<div><div>Neural Network</div><div>December 2017 - Present</div></div> <p>A simple neural network that I wanted available to use in future projects.</p> <ul style="list-style-type: none">Current Features:<ul style="list-style-type: none">Any reasonable number of inputs, hidden layers, and outputsFully-functional backpropogationTested by averaging numbers and playing a simple ASCII gameUpcoming Features:<ul style="list-style-type: none">SerializationMixing of different neural networks in addition to backpropogation for increased learning rates and simulation of "evolution"	