

Adam Rezich

Programmer / Designer

4257 Twin Peak Lane
Rapid City, SD 57702
+1 (605) 939-0108
adam@rezich.com
<http://rezich.com>

Mission

To work with technology to make great things.

- Adaptable; able to quickly learn and adapt to new languages, tools, and workflows
- Able and willing to relocate
- Experience working remotely
- 15 years of largely self-taught programming and game & web design experience

Education

DigiPen Institute of Technology—Bachelor of Science in Computer Science and Game Design (2013-2015; incomplete)

Programming languages

C/C++, C#, JavaScript/NodeJS, HTML/CSS, Ruby, Python, PHP, SQL, Visual Basic

Software

Visual Studio, Unity, Godot, XNA/MonoGame, Xamarin, Vagrant, JIRA

Source control

Git, SVN, Mercurial

Operating systems

Windows, Linux

Work experience

ScrapTF (Sept. 2015–July 2016, July 2017–Nov. 2018)

(PHP, CSS, JavaScript) I have done extensive contract web design and programming work for ScrapTF, a Team Fortress 2 hat-trading community website. I added features to the website—both user-facing and administrator-only—as requested by my employers. The ScrapTF technology stack is built around a hand-written PHP CMS that interfaces with both the Team Fortress 2 and Steam APIs to enable users trade items with ScrapTF “bots” in order to “bank” these virtual items, so users can trade them, give them away in raffles, and so forth. I designed and implemented a “Christmas event” “game” on the site that increased user engagement so much that the system was reused for various future holiday events. I also entirely rewrote the control panel that site administrators use to set the prices of various items, optimizing it for performance while adding various features such as user-defined sales heuristics that highlight items whose prices should be changed. ScrapTF was not set up to have developers other than the two founders working on it, so before I began working on the website proper I taught myself how to use Vagrant and set the ScrapTF repository up such that myself and other future hires can use simple commands to run a development version of their website in a local VM. ([Website](#), [design sample 1](#), [design sample 2](#))

Student projects

Super Magical Jumping Dude

(Python) A 2D puzzle platformer created at DigiPen Institute of technology for my freshman solo game project. I designed and implemented the gameplay, levels, music, sound effects, and graphics. DigiPen now uses this game as part of their curriculum to teach kids programming and game design in their “ProjectFUN” summer camps. ([YouTube video](#), [Windows executable](#))

OmniArch

(C++, JavaScript) A 2D action platformer created at DigiPen Institute of Technology for my sophomore team game project. Built from scratch in C++, with JavaScript scripting, I built a large part of the underlying engine of the game, especially the Entity-Component System and its memory allocators. I worked with my team to ensure that the various parts of the engine that the others were working on (graphics, physics, scripting integration) all connected properly. ([Website](#))

Game School Simulator 2015

(C) A simple 2D management sim created at DigiPen Institute of Technology for my freshman team game project. Created from scratch in C (using only a provided simple library for graphics calls), I designed and implemented the underlying architecture of the game engine, helped to create the gameplay flow, and designed most of the user interface. ([Website](#))

Unending Bonus

(Zilch) A pseudo-3D arcade-style shooter created at DigiPen Institute of Technology for a 2D Game Design course. Created using DigiPen’s proprietary Zero Engine (similar to Unity). I imposed upon myself the restriction of only using the engine’s debug line drawing functionality to create the game’s visuals; the game does not have any raster graphics at all. I did the 3D projection math manually, and created an entire underlying framework for semi-efficiently drawing vector graphics in this manner. ([YouTube video](#), [Windows executable](#))

Personal projects

Additional personal projects available on my website, <http://rezich.com>.

SoulsRL

(JavaScript, NodeJS) A browser-based roguelike game inspired by Dark Souls. I mostly focused on building a browser based “roguelike” game engine from scratch and playing with client-server interactions using Node instead of focusing on finishing the gameplay. ([Play in browser](#), [GitHub](#))

dotabutt

(NodeJS) Dota 2 is a competitive multiplayer video game, and Dotabuff (<https://www.dotabuff.com>) is a commercial player stat-tracking website using the Dota 2 API. dotabutt is my own open-source clone of basic Dotabuff functionality, including my own NodeJS Dota 2 and Steam API wrappers. I stopped working on it when it became clear that in order to compete with Dotabuff, I would need to scale my database storage solution beyond the terabyte hard drive in the public-facing development server in my bedroom, by about a terabyte a month. dotabutt hasn’t been updated since Dota 2 was in beta in 2013, but the project remains open-source for educational purposes regardless. ([GitHub](#))

References available upon request.