Adam Rezich

Programmer / Designer

8942 Redmond-Woodinville Road NE
Redmond, WA 98052
+1 (605) 939-0108
adam@rezich.com
http://rezich.com

MISSION

To express myself creatively, whether through programming or design, in the creation of digital interactive experiences.

EDUCATION

DigiPen Institute of Technology

Bachelor of Science in Computer Science and Game Design (2013 - 2015)

TECHNOLOGY SUMMARY

Programming languages

Experienced in: **C/C++**, **C#**, **JavaScript/ECMAScript**, **HTML/CSS**, **Ruby**, **Python**, **PHP**, **SQL**

Software

Visual Studio, **Unity**, **Xamarin**, **Game Maker**, **Flash**

Source control

Git, **SVN**, **Mercurial**

Operating systems

Windows, **Linux**

WORK EXPERIENCE

ScrapTF (Sept. 2015—July 2016)

(**PHP / CSS / JavaScript**) I spent nine months doing contracted design and development work for ScrapTF, a *Team Fortress 2* hat-trading community website. I did both backend PHP and frontend HTML/CSS/JavaScript. Sometimes I even got to flex my game design skills a bit—easily the most satisfying part of the job was building the collaborative holiday event system, designing reward systems, and making a "quest log" and giant "XP bar" that animate *beautifully*. Aside from that, I mocked up and then implemented new website features as requested by my boss, refactored the backend codebase, and occasionally helped out with community content moderation.

[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 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 game team project. Built from scratch in C++, with JavaScript scripting. I built a large part of the underlying engine of the game, especially the entity and component system. I worked with my team to ensure that the various parts of the engine that the others were writing (graphics system, physics system, scripting integration) all connected together properly.

[[Website](#)]

Unending Bonus

(**Zilch / Zero Engine**) A pseudo-3D arcade-style shooter created at DigiPen Institute of Technology for a 2D Game Design class. Created using DigiPen's proprietary Zero Engine (very similar to Unity). I imposed upon myself the restriction of only using Zero's debug line drawing functionality; as such, the game contains no sprites or other art of any kind. I did the 3D projection math myself,along with the entire underlying system that allowed me to group lines together as "VectorSprites".

[[YouTube video](#) | [Windows executable](#)]

Game School Simulator 2015

(**C**) A simple 2D management sim created at DigiPen Institute of Technology for my freshman game team project. Created nearly from scratch in C (using DigiPen's proprietary "Alpha Engine" library for graphics calls), I designed the underlying architecture of the game engine, helped create the gameplay flow, and designed most of the user interface.

[[Website](#)]

PERSONAL PROJECTS

crrsh

(**C# / Unity**) A school project that I eventually scrapped, I'm working on rebuilding it in Unity and eventually completing it. The game puts the player in the role of a computer technician using a command-line remote recovery shell to try and figure out why communications with the manned lunar research base have ceased. So far I've created a sort of emulator for a fake retro PC, and an emulator for a fake vector monitor.

[[Play in browser](#) | [YouTube video (old version)](#)]

SoulsRL

(**JavaScript / Node.js**) A browser-based roguelike game inspired by *Dark Souls*. I mostly focused on building a browser-based roguelike engine from scratch and playing with client-server interactions using Node.js instead of focusing on the gameplay.

[[Play in browser](#) | [GitHub](#)]

arena

(**C# / XNA**) A short-lived attempt at making a *Dota 2* or *League of Legends*-style "moba" using only abstract shapes, much like the *Geometry Wars* series, instead of polygonal models. I stopped working on it once I enrolled at DigiPen Institute of Technology, but I learned a lot about networking and the difficulty in synchronizing gameplay across multiple clients.

[[GitHub](#)]

References available upon request.