Zachary Loveless

Contact: Zack (He/Him) – zack.loveless@live.com – Yuma, Arizona

TECH SKILLS

- Languages: C#/.NET, Python 3, Bash, Batch, SQL
- Frameworks: ASP.NET, .NET Core/Framework, React, React Toolkit (RTK)
- Operating systems: Windows, Linux (Debian, Fedora)
- Networking: Basic network configuration and troubleshooting
- System administration: Experience with system upgrades, backups, virtualization, and automation.
- Tools: Git, Docker, Docker Compose, Ansible, Hypervisors (Proxmox)

WORK EXPERIENCE

Software Developer Drift Glass Ventures (Remote) February 2024 – September 2024

- Developed a double-entry accounting financial modeler in Python.
- Created a React application in TypeScript for viewing simulation results from a custom financial modeling program.
- Integrated all components into a single-page application including running python in the browser with web assembly.

Computer Lab Aide (part-time) Arizona Western College (Yuma, AZ) August 2008 – June 2023

- Delivered tier one / frontline support to customers and aided them in achieving success in an academic environment.
- Managed over 140 machines and assisted hundreds of students per day administer their accounts.
- Performed troubleshooting and debugged lab hardware and faults.

Web Developer (Contract, full-time) S2 Systems (Remote) March 2019 – October 2019

- Implemented designer wireframes for websites in JavaScript, C#, and HTML on Amazon EC2.
- Overhauled CSS development process to use reusable SCSS snippets.
- Maintained a multitenant website project in ASP.NET Core with senior developer.

EDUCATION

- BASc Public Administration, Northern Arizona University, Yuma, Arizona. December 2021

SKILLS

- Solid talent for automation and organization.
- Excellent ability to write.
- Strong analytical and reasoning capabilities.
- Effective customer service skills.
- Good time management ability.

PORTFOLIO

Totem Arts 2020 – present Mixed privacy

- **Technologies**: Ansible, Docker, Proxmox, CI/CD, Git, Open ID Connect
- **Website**: https://totemarts.games
- **Sources**:
  - https://github.com/zloveless/ansible-common
  - https://github.com/zloveless/ansible-containers

Totem Arts is a not-for-profit game development group using the Unreal Development Kit to make the award-winning spiritual *remaster* for C&C Renegade, calling it Renegade-X. Shortly after I joined the team in 2020, we began working on a sequel to Renegade-X called *Firestorm* in the same game engine. I maintain team services required to keep Renegade-X operational throughout development of Firestorm and help the team better communicate and organize their own visions for the game. I use Ansible to deploy Docker Compose files (stacks) containing definitions for our infrastructure using a custom Ansible role that I wrote. We use Proxmox to virtualize our infrastructure so it can be restored in the event of disaster.

RxCmd 2014 Open Source

- **Technologies**: C#/.NET (Framework), Managed Extensibility Framework (MEF)
- **Source**: https://github.com/zloveless/RxCmd

RxCmd is a command-line utility that interacts with a Renegade X game server's RCON-like (remote console) interface. The team behind Renegade X developed a rudimentary remote console interface for moderating game servers, so to learn it, I wrote a console app that let you connect to an instance of a game server, subscribe to logs, and add bots. The structure was modular, allowing commands to be loaded at startup and executed later in the console. The basic commands included wrappers around fetching basic information and subscribing to game logs for testing an IRC regulator.

Financial Modeler 2024 Closed Source

- **Technologies**: Python, TypeScript, React, HTML, CSS/SASS, Git
- **Link**: https://fplan.driftglassventures.com

At Drift Glass Ventures, I built a financial modeler in Python that simulates the cashflow of an idea the user has. The model gave the user a better picture of where obvious shortfalls get highlighted as errors in the model output. The modeling software includes both a python library and command-line component, as well as a React website. The python library allows the user to install the python library locally in their own environment, while the React website provides a web editor using Monaco (embedded VScode) to create a financial model. The most difficult problem that I faced was in implementing the editor's autocomplete which at the time of implementation is not as complete as a regular VScode with a proper language server (LSP) for Python.