

Curtis Rose

CurtRose@gmail.com -- (602) 327-2009 -- <https://curtrose.com>

EXPERIENCE

Roche Tissue Diagnostics, Tucson, AZ – Software Engineer

May 2021 – Present

- Designed two machine learning algorithms for processing tissue samples to help identify specific biomarkers.
- Developing mixed reality applications to help pathologists and histotechnicians be more efficient in the lab.
- Developed additional mixed reality applications for demonstrating early prototypes of hardware designs.
- Designed robotic workflow automation using the ABB Yumi robot.
 - Automated the movement of tissue slides from one container medium to another.
 - This involved image processing to determine the existence of a slide(s) in the first container, and the existence of an empty slide position(s) in the other.
 - The camera being used was the onboard Cognex camera system.
- Created a concept demonstration animation using Unity and SolidWorks models.
 - This demonstration is meant to show the utility of a potential hardware upgrade. The successful demonstration resulted in the upgrade being accepted and added to current and future Roche products.

Raytheon Technologies, Tucson, AZ – Senior Software Engineer (DoD Secret)

Feb 2018 – March 2021

- Developed and maintained embedded software for the Exoatmospheric Kill Vehicle.
- Designed and implemented a new software unit that ran on a preemptive, multitasking real-time operating system.
 - This software unit was a critical improvement to the EKV system that had strict timing and processing throughput requirements.
- Wrote telemetry software for capturing system state used in testing and debugging of the system software.
- Constructed tools that were used in testing, verifying, and automating tasks that had previously been completed by engineers.
 - These tools saved hundreds of engineering man-hours per build and resulted in quicker turnaround on various artifact generation and testing results.
 - They also reduced the dependency on ‘tribal knowledge’ which helped protect from the loss of that information over time.
- Integrated Google Test to help facilitate the development of accurate and consistent software.

United States Geological Survey, Flagstaff, AZ – Software Developer (DoI Secret)

Jan 2016 – Feb 2017

- Developed multi-spectral image processing software for data gathered from various NASA missions.

GRADUATE COURSEWORK

- Designed an autonomous agent that determined collisionless actions while navigating a maze using a **Feed-Forward Neural Network**.
- Developed a **Convolutional Neural Network** whose goal was to discriminate between a fixed number of classes that could be found in an image dataset.
- Studied Markov Decision Processes, Bayesian Networks, speech recognition, and natural language processing.

EDUCATION

Arizona State University

MS Computer Science (2019 - Present)

Northern Arizona University, Flagstaff

B.S. Computer Science (2013 - 2017)

SKILLS

- Programming Languages
 - Expert - **C, C++, C#**
 - Advanced - **Python, Javascript**
 - Experienced - **Java**
- Applications
 - Visual Studio, Unity, MatLab
- Six Sigma Specialist
- Agile Developer

AWARDS / RECOGNITION

- **Leadership-Sponsored Achievement Award** (Raytheon Technologies - Apr 2020)
 - Recognized for innovation in an ongoing effort of automation through pipeline and tool development to aid in the testing and development of embedded software
- **Achievement Award** (Raytheon Technologies - Recognized 2 times - Aug 2018, Mar 2019)
 - Recognized for leading the development and delivery of a software product to the customer on time under a strict schedule resulting in the successful fielding of critical hardware and ensuring receipt of monetary reward to the company for meeting the delivery milestone
- **Spot Award** (Raytheon Technologies - Recognized 4 times - May 2018, Sep 2018, Nov 2018, Sep 2020)
 - Spot Awards are recognitions of exceptional performance by department managers, team leads, and senior engineers.

PERSONAL PROJECTS

- **3D Chess Game Variant** [<http://curtrose.com/unity.html>]
 - I made this game as a challenge to myself during a two week period of time.
 - Some requirements for this game were that it had to be able to run from my website, it had to include 3D models and animations, and that it included a full menu system to navigate through.
 - I completed the main game early and so I decided to add new game modes to stress test my earlier design decisions to see how well they scaled to new ideas.
 - This was a fantastic opportunity to find weaknesses in my earlier designs and iterate over them, in place, so that they wouldn't negatively impact the existing features, but would be able to support the new game modes.
 - Integrated as a Web3 app that interfaces with software wallets and identifies users through the wallet.
- **Virtual Reality First Person Shooter**
 - This game was developed in Unity for the HTC Vive. It was effectively a clone of the famous "Nazi Zombies" game mode from many Call of Duty entries. I developed the entire interface from the ground up. This included interacting with objects by detecting collisions with the VR controllers, picking up items, orienting items based on how they should be held, defining behavior based on follow on input.
 - A video can be seen on my website at <http://curtrose.com/portfolio.html>. There is an option for "Zombie VR Game" with two videos.
- **Game of Tiles**
 - Game of Tiles is a tile based JRPG written in Javascript. It was made in a GameJam setting over the course of three days. I designed this game from scratch in those three days, at that point, having never really made a game. I then assigned my teammates to portions of the game and we all worked together to get it done on time. This game is actually playable if you would like to take a look. The tilesets in the first world you find yourself in were designed by me using Aseprite, the remaining worlds were designed by my teammates.
<http://curtrose.com/GameOfTiles/GameOfTiles.html> (Make sure you have your volume on! The sound effects were all recorded by us, they are pretty funny)

- **Ray Tracer Algorithm**

- I developed a full ray tracing algorithm from scratch using JavaScript. This algorithm is, fundamentally, recreating a picture camera, but in reverse.
- The idea is that you trace vectors from the 'source' (read: camera) out into the world through each pixel in an image plane. When the vector hits something in the world (mathematical spheres in my example image) It would determine the color of the pixel it had traveled through to get there.
- I added a recursive element to this vector that would bounce around the world, taking into account the material properties of the objects it hit (reflectivity for example) and sum the color values into one. This resulted in much more interesting images with reflections and specular highlights.
- I also included light sources so that there would be shadows and more interesting reflections.
- There is an example output image in my portfolio I linked above.