

Elijah Nicasio

Riverside, CA | (951)-200-0937 | ElijahN8@gmail.com | elithenicasio.github.io

Work and Research Experience

Research and Development Engineer

September 2022 - April 2024

Utopia Compression Corporation

Los Angeles, CA

- Created nautical simulation in UnrealEngine5 to generate infrared camera data to train a machine learning model
- Implemented YOLO object detection algorithm into different legacy C++ and Python codebases in order to improve object detection and classification of algorithms that track various types of vehicles
- Maintained, debugged, and documented a sense and avoid system for autonomous UAV that uses machine learning to detect and avoid other aircraft using C++, Python, ROS, and AirSim, built on a NVidia Xavier
- Developed a web app in javascript that served as a bridge between two external applications that couldn't communicate with each other, piping data, making API calls, and dealing with authentication and authorization
- Developed a web UI in Python and Javascript for a deep learning detection and classification pipeline so the pipeline can be controlled remotely and with more precision. UI allows users to see detection locations on a map and allows users to control an external camera that can pan and zoom at will
- Aided office with general technical issues, such as building and fixing computers, and resolving network issues

Undergraduate Research Assistant

February 2019 - November 2020

Prof. Jiasi Chen, *University of California, Riverside*

Riverside, CA

- Contributed to a joint project with AT&T that attempts to show the benefits of 5G for AR environments on mobile devices. Project had users look into an AR space using cameras on mobile devices, placing down 3d models synced with other users. Other users would then download placed models over 5G, whose complexity would differ depending on distance from the user. Project was based off of ARCore and developed in Unity. Was responsible for core features such as UI, gestures, and 3D model synchronization

Software Visualization Intern

June 2019 - August 2019

Jet Propulsion Laboratory, NASA

Pasadena, CA

- Researched the feasibility of multiple game engines to determine what is most suitable for the mission-visualization team
- Created a fullstack, interactive visualization to test viability of web-based game engine
- Built a frontend displaying a to-scale visualization of the Solar System in Babylon.js. Implemented features such as real time shadows and tracking of live spacecraft. Additionally added support for virtual reality to visualization.
- Created a backend Flask server to pipe real time and historical telemetry data to Solar System visualization

Education

University of California, Riverside

Riverside, CA

Master of Science, Computer Science

June 2021

Bachelor of Science, Computer Science

March 2020

Projects (some projects can be found at elithenicasio.github.io)

GeoConnect (Match-3 Line Puzzle Game) (GDScript, Godot)

- Designed and developed a match-3 game in Godot
- Game has the user attempt to draw a line to connect identical tokens in order to score points. The user is rewarded for longer connections, and has to aim to beat increasing scores consecutively

VR Cube Slicing Game (Fruit Ninja VR Clone) (C#, Unity)

- Designed and developed a Virtual Reality arcade-style video game for the Oculus Rift in Unity
- Goal of the game is for the player to slice as many randomly spawning cubes as possible in sixty seconds, effectively translating the popular mobile game Fruit Ninja into the VR space. Implemented various gameplay elements such as real-time mesh cutting and VR specific audio and visual cues

Augmented Reality Game Advisor (MS Capstone Project) (C#, Unity, UWP, Python, Tensorflow)

- Developed a program that gave board game advice to user in augmented reality, depending on what game user is looking at
- Program consisted of two parts: a Unity/Universal Windows Platform client app on an AR device written in C#, and a Python server backend that uses a Tensorflow machine learning model to classify the board game a user is looking at
- Client program on AR device sends images to server program, which attempts to classify what board game the user is looking at using machine learning. The next best move for the user is then calculated based on the current board state and sent to the AR device, which displays the move to the user

Skills

Languages: Python, C++/C, GDScript, Javascript, C#

Frameworks/Libraries: Pytorch, Numpy, Tensorflow, Node.js, ROS, Django, Flask, OpenCV

Tools: Godot, Unreal, Unity, Babylon.js, UWP