Michael Yuen

Myyuen@uci.edu
Portfolio: https://luxebo.github.io

Education

August 2019-May 2021 **University of Southern California** Los Angeles, CA

- Masters of Science in Computer Science; GPA: 4.00, MS Honors
- Relevant Coursework: databases, algorithms, game machine learning, mobile games, game network architectures, web tech
- Grader for software engineering, graduate databases, programming systems, mobile games, Lead Grader and TA for AI

August 2016-June 2019 **University of California, Irvine** Irvine, CA

- Bachelors of Science in Computer Science specializing in Intelligence Systems; GPA: 3.79, Cum Laude
- Relevant Coursework: databases, operating systems, artificial intelligence, data structures, analysis of algorithms, machine learning, game development, information retrieval, graph theory, computer vision

Work Experience

January 2016-April 2016 **Architect Internship at LA Design Group** Pomona, CA

- Sketched a house design on paper using drafting tools
- Computer Aided Design on the house design with Autodesk Inventor and AutoCAD
- Presentation of the house design in AutoCAD in a professional setting with the company

August 2019-May 2021 **Teaching Experience** Los Angeles, CA

- Course Producer for Software Engineering, Grader for Graduate Database Systems, Grader for Intro to Programming Systems, Grader for Mobile Games, Lead Course Producer for Artificial Intelligence (Spring 2020), and Teaching Assistant for Artificial Intelligence (Spring 2021)
- Graded coursework, helped organize teaching staff, hosted office hours both online and offline, and answered debugging and theoretical questions on Piazza (online forum of class)
- Built a Machine Learning homework that is autograded for Artificial Intelligence that required students to build a K-Nearest Neighbors classifier from scratch, use sklearn to build ensembles, and use Pytorch to build convolutional neural networks

March 2021-April 2021 **UN Disaster Solving Project** Los Angeles, CA

- Worked on a short hackathon style contract with the UN under Professor Vangelis to build a disaster solving project using Unity WebGL and Photon (general multiplayer and Voice)
- The Client wanted an application to allow users from multiple countries to connect to one another on the web and use a board game that simulates a disaster in order to collaborate for a solution

July 2021-August 2022 **Microscape Full Stack/Unity Developer** Remote, CA

- Worked in a startup in biotech focused on a Unity product that allows users to add and render medical data as a 3D image
- Implemented multiplayer functionality with rooms and multi-user interaction for the application with Photon PUN2
- Adjusted the volume renderer algorithm and worked on different shader, cropping, filtering adjustments to the image
- Adjusted sliders to work on the VR medium, including annotations using a brush tool
- Used AWS Lambda to save and load filtering/cropping/shader settings that are set for a specific image
- Added a multi asset feature, allowing for users to load in multiple 3D image assets and change their transforms, filters, etc

April 2022-December 2022 **Prism Unity Developer** Remote, CA

- Worked in a startup (part time) in biotech focused on a product using IoT to interact with a Unity game application
- Prototyped games including a new minigame involving shooting a slingshot with breathing rate of a Polar waistband
- Worked on a breathing game that used a chromium web browser to watch Netflix, Spotify, etc while gathering medical data
- Helped add small features such as UI popups and UI functionality to a flappy bird game

September 2022-December 2022 **Norvoc Bioscience Software Developer** Irvine, CA

- Worked in a biotech company. Project is focused on data visualization and making the data interactive from ML models
- Built this project using React, Node.js, Mongodb, Synology servers, and Three.js for visualization
- Specific React features used are splitting the rendering of html into different components and using state to save data from the ML models pulled from the routes of the proxy server of Node.js queried from the Mongodb database hosted on Synology
- Used Three.js to create generalized cylinders, along with using open GLSL to write shader code for a gradient

December 2022-September 2023 **Game Of Silks Unity Developer** Remote, CA

- Worked in a metaverse/NFT/crypto startup that wants to use NFTs of real horses to let users gamble in real horse races
- Built a dialogue system, created scene transitions, saved scenes to asset bundles on AWS S3, and optimized for webGL
- Worked with an art team designing materials, lighting, shaders, models, animations, rigging for Unity
- Ingested 10000 real horses from the jockey club API through using AWS lambda functions and inserting into a SQL table
- Created an ASP.NET API to allow users to buy land NFTs and access the SQL database through a Three.js frontend
- Used Moralis to allow users to mint NFTs using a smart contract, connecting to their Metamask wallet

October 2023-Present **Robot Sea Monster Games Tech Lead/Unity Developer**
Remote, CA

- Working in a game studio that builds client projects, which goals were software architecture, project managing, scoping and delegating tasks for junior devs, estimating time per task, code reviews, technical documentation, and talking to clients
- Led a team of developers to build a social media application on Unity for a client in Android and iOS, core components include using client's Firebase backend and API, swiping between a grid of instagram-like posts with object pooling and a map system, globe asset with markers and locations, and Photon PUN2 to play a simple multiplayer game
- Successfully completed the project and extended the contract twice for maintenance periods
- Helped on an LLM escape room game (Harker's Escape), building basic gameplay mechanics to support the LLM such as physics with gameobjects, crafting mechanics, 3C (Character, camera, controller), json parsing, and UI input and menus
- Successfully handed off the project to other devs to complete after starting a solid codebase

January 2024-Present **Stellar Dust (Robot Sea Monster Games Tech Lead)**
Remote, CA

- Led development, project managing, and architecture for the vast majority of the year on an internal VR RTS game

- Game's goal is to have two players send ships against each other in an indirect graph map system through nodes, with the winner taking over the enemy's home node (base), which the game has multiplayer and AI
- Game is to release within the next month(s) to open beta and full release, with potential to do live service and F2P
- Key coding functions include general gameplay programming, tools programming, network architecture and backend (dedicated server migration), game design, QA and bug fixing, and building and dev ops pipeline
- Led a team of up to 6 developers, scoping and delegating tasks, estimating time per task, code reviewing, adjusting the Monday board with priorities, and making sure all devs finish their work with quality code and as seamlessly as possible
- Wrote technical docs scoping tasks and architecture on Confluence, wrote weekly patch notes and versioning
- Git flow was built in a way to have each dev put their code in a certain branch, which then create pull requests for review
- Setup Oculus authentication, leaderboard systems for both Oculus and Playfab, with fake AI players integrated; this leaderboard scoring system is integrated for competitive play, queuing and matchmaking through Playfab's queues
- Revamped the menu UI multiple times to use new UI assets, along with having correct Photon Fusion 2 flows through Photon callbacks, and calling correct http calls to talk to a PHP laravel hosted backend to call Playfab's dedicated server API
- Scoped multiple massive refactors, including removing Tilia to use OVR Input exclusively (and possibly OpenXR in the future), migrating from Photon fusion 1 to fusion 2, changing from Photon host mode to Photon dedicated server mode
- Made game flow and progression through saving player data on Playfab, allowing for different arena types (game modes), such as competitive mode, casual mode, play vs ai, and tutorials, with different objectives on different maps
- Prototyped an initial AI to be able to play basic gameplay, with designer settings to allow for editing the AI heuristic
- Worked on many gameplay mechanics; unit tracking and behavior, node adjustments and types, in-game objective changes, random and predefined map generation, player controller input variations, hand tracking functionality, map manipulation (rotation, panning, scaling), game settings, preference menus and UI tooltips on controller, node GUI types, and art passes
- Built both linux docker dedicated server to Playfab and client android apk for Quest Store in weekly Sprints for testing in QA; two environments with dev and prod release channels to share to QA and other testers through the Quest Store
- Fixed hundreds of minor and major bugs, assigned and worked on many smaller tasks (music, UI changes, and more)

Research

January 2020-May 2020 **MTL Constraints with Abstract MDP on Value Iteration** Los Angeles, CA

- Read multiple papers and researched topics on safety constraints in reinforcement learning, including reward hacking, reward shaping, and reward function design
- Concept based on simplifying the abstract Markov Decision Process into a smaller variant in order to allow for logic formulas to dictate how the policy and utilities are calculated in Value Iteration
- Used networkx, pymdptoolbox, and py-metric-temporal-logic to represent the MDP as a graph problem and used metric temporal logic on the MDP
- Wrote short reports in Latex to explain the work done

June 2021-July 2021 **XROS** Los Angeles, CA

- Helped Powen Yao on his PhD research project related to peripersonal space on VR
- Used SteamVR to build a simple cube slicer demo in VR, where the user can grab colored swords around their waist and slice cubes into smaller meshes
- Added audio sfx and music in order to make the demo more immersive
- Helped record some basic metrics and gather data for an ML in VR paper

Projects (Engineering)

September 2018-December 2018 **Pete The Penguin: A 3D Game using Unity**

- Used the Unity Engine to create a 3D survival game with a group of 7 people
- Did level design, environment, and documentation of the game
- Utilized Agile development through Sprints and iterative development in Github
- Presented and demoed the game in a game showcasing at UCI

July 2018-Present **My Portfolio Website**

- Learned HTML, CSS, JS, Bootstrap, DOM manipulation, and jQuery on my own to create a website hosted under UCI's Computer Science department's servers
- Created a Color Game, a Todo-list, a Patatap Clone, and a Portfolio of previous artwork that I have done
- Link is here: https://www.ics.uci.edu/~myyuen/

September 2018-December 2018 **Kaggle Competition**

- Used sklearn and keras libraries to build K-Nearest Neighbors, Random Forests, and Neural Networks to predict rainfall
- Worked in a group of 3 people through Kaggle's kernels to build a series of ipython notebooks
- Reached the top 10% of the private leaderboard

February 2019-March 2019 **Search Engine**

- Used Python, Mongodb, and Flask to make a search engine on scraped websites from UCI's ICS domain
- Created an inverted index using Mongodb and a simple GUI with Flask
- Worked in a group of 3 people to implement cosine similarity, tf-idf, and other ranking methods

April 2019-June 2019 **Object Reconstruction**

- Took many images of a dragon model using a set of scanners and a projector
- Used Python's OpenCV and Numpy libraries to generate a mesh from the images taken from scanners
- Used camera calibration to determine camera parameters, generated a baseline mesh from triangulation
- Finalized a mesh from MeshLab using alignment and poisson surface reconstruction

August 2019-December 2019 **Applied ML in Hovenstar**

- Used Hovenstar, a turn based game in Unity, as a testbed in order to train deep q-learning models
- Used ml-agents to train different reinforcement learning models with a python backend, showcasing the impact on Unity
- Took many tile assets from Unity's asset store and used openCV transformations to create a large dataset of tiles for a GAN to generate tiles
- Built a GAN to generate many iterations of generated tiles, creating batches of usable sprites for Hovenstar or any other tile based game
- This game ended up being: https://store.steampowered.com/app/2881620/Paraside_Duality_Unbound/

January 2020-May 2020 **Dungeon Smiths**

- Used the Unity Engine to create a unique mobile (Android) maze crawler game with a group of 6 people
- Game has an exterior 3D maze that forces the player into 2D minigames upon reaching an enemy
- Developed multiple minigames including two final bosses, simple cutscenes, and dialogue functionality
- Utilized Agile development through Sprints and iterative development in Github

January 2021-May 2021 **Inside Job**

- Used the Unity Engine and the Photon PUN2 network architecture to create a simple social deduction game
- Built a lobby system using Photon’s Room properties, allowing players to create and join games hosted in different locations
- Built a better dead reckoning system for synchronization of game objects for moving, shooting, and interacting with objects
- Built a proximity text and voice chat from Photon’s Chat and Voice API

January 2021-May 2021 **A Series of Web Tech Projects**

- Used HTML, CSS, and JS for frontend, Python Flask for backend, the TMDB API, and Azure to build a movie searcher
- Used Angular and Bootstrap 4 for frontend, NodeJS for backend, the TMDB API, and GCP to build a movie details site
- Used Android Studio to develop a mobile application for movie details using the same backend and hosting service
- Used HTML, CSS, JS, NodeJS, Heroku, and MongoDB to create a camp search application that allows users to sign up, login, and search, create, and review campgrounds

January 2021-May 2021 **Movie Searcher**

- Used HTML, CSS, and JS for frontend of a webpage, allowing users to search for movies hosted on the TMDB API and receive details of each movie or TV show that results from the search
- Used Python and Flask to parse JSON calls from The Movie Database API as a backend
- Hosted the website via Azure in order to host dynamic content, connecting and hosting server side code to each client

January 2021-May 2021 **Movie Details: Website**

- Used HTML, CSS, Angular, and Bootstrap 4 for frontend of a webpage, allowing users to use a responsive website to access recent and featured movies and TV shows through carousels, find details and social media of movies, and access cast and review details, including a search feature
- Used RESTful APIs through NodeJS to make callbacks to The Movie Database API as a backend
- Hosted the website via Google Cloud Platform in order to host dynamic content, connecting and hosting server side code to each client

January 2021-May 2021 **Movie Details: Android App**

- Used Android Studio with Java for frontend of a mobile application, allowing users to use a responsive Android application to access recent and featured movies and TV shows through carousels, find details and social media of movies, access cast and review details, use a search feature to find the info needed, and add movies and TV shows to a watchlist
- Used RESTful APIs through NodeJS to make callbacks to The Movie Database API as a backend
- Hosted the backend via Google Cloud Platform in order to host dynamic content and have the app access the backend

January 2021-May 2021 **Campgrounds**

- Used HTML, CSS, JS, and Bootstrap 4 for frontend of a webpage, allowing users to use a responsive website to sign up, login, logout, add campgrounds to the site, and review and comment on campgrounds other users create
- Used Mongodb to store campground, review, and comment data for each user, along with authentication
- Used RESTful APIs through NodeJS to make callbacks to the Mongodb database as a backend
- Hosted the website via Heroku in order to host dynamic content, connecting and hosting server side code to each client

September 2022-Present **Social Deduction Games Webpage**

- Built a website at https://luxsdg.github.io hosted by github pages (project) in order to design multiple social deduction games
- Playtested these games using discord with some friends locally and online
- Tested these pen and paper games against myself in curated tests (through statistical analysis and other measurements)
- Built a turn based board game that will be transitioned to a video game on a separate Notion webpage/Miro diagrams

Skills

- **Languages:** Proficient: C#, Python; Familiar: Java, SQL, HTML, CSS, Javascript, Typescript; Basic: C++, C
- **Systems and Software, Proficient:** Windows, Photon, Oculus VR, Unity, Playfab, AWS, Three.js, Git
- **Systems and Software, Familiar:** Linux, Oracle VM, Visual Studio, Jupyter, Numpy stack, Android Studio, Firebase, NodeJS, Flask, Azure, GCP, Postman, MongoDB, Docker, React, React Native, Moralis, Magic Link, Metamask, ASP.NET, Confluence/Jira, Monday, Notion, Miro, Figma, 1Password, Slack, Discord, Toggl
- **Photon:** PUN2, Fusion 1, Fusion 2, Photon Voice, Photon Chat
- **Playfab:** Playfab auth and accounts, leaderboards, dedicated servers, matchmaking queue, client/server APIs
- **Oculus VR:** Meta Quest 2, 3, Pro; Quest Store release channels, Oculus Auth, OVR Input (controller, hand tracking)
- **AWS:** Lambda, Cloudwatch, S3, Dynamodb, SQS, SNS, Eventbridge, Cloudfront, Parameter Store