Yao Lin

https://github.com/matudinglin

Email : matudinglin@gmail.com
Mobile : +1-213-551-4786

Professional Skills

- **Languages:** C++, C#, Python, Java, SQL
- **Libraries & Frameworks:** Unity Engine, XR Interaction Toolkit, Oculus Integration, Unreal Engine, OpenGL, Qt, Git, Keras
- **Interests:** VR/AR, Software Development, Computer Graphics, Human–Computer Interaction, Game Development

Education

- **University of Southern California** California, USA
  Master of Science in Computer Science; GPA: 4.0 *Aug. 2022 – June. 2024(Expected)*
- **Dalian University of Technology** Dalian, China
  Bachelor of Engineering in Computer Science and Technology; GPA: 3.66 *Sept. 2018 – June. 2022*

Internship

- **CViSS Lab at University of Waterloo** Ontario, Canada
  Research Intern advised by Prof. Chul Min Yeum | Team Leader *Jun. 2021 – Sept. 2021*
  - Designed a **virtual reality** point cloud viewer application to visualize building structures, allowing users to explore and collaborate in 3D virtual environment.
  - Designed **10+ features** for building structure assessment, such as measuring tool, annotation tool and panorama image viewer; Processed and rendered different formats of raw point cloud data in game engine.
  - Developed the application using C# and Oculus Integration on Unity Engine, and deployed it on **Oculus Quest 2**.
- **The Future Lab of Tsinghua University** Beijing, China
  Research Assistant Intern advised by Prof. Qi Lu | Main Developer *Sept. 2021 – Dec. 2021*
  - Built an end-to-end real-time non-destructive fruit quality detection system by analyzing spectrum data collected from spectrometer, which has been adopted by the produce industry with **92%+ accuracy** for passion fruit classification.
  - Proposed a complete analysis method for fruit quality detection, including data collection, data analysis, data preprocessing, data calibration and model establishment using **PLS regression model**.
  - Wrote a software for chemometric analysis and serial port communication with GUI in **C++** and C#, which allows the system to show the analyzed result in **real-time** and accelerate data collection process by **250%**.

Research & Development

- **VR Exploration Tool for Visually Impaired People** University of Wisconsin-Madison, USA
  Research Assistant advised by Prof. Yuhang Zhao | Main Developer *Jun. 2022 – Sept. 2022*
  - Developed a **VR application** which enabled visually impaired people to explore and navigate in virtual environment.
  - Developed **8+ features** to enhance the user experience, including three view modes, finger gesture navigation, avatar movement, object outline highlight and object interaction.
  - Developed the application using C# on **Unity Engine** and deployed it on **IOS** and **Android** devices; Received positive feedback from user experiments involving exploration, navigation and finding tasks in virtual environment.
- **Automatic Generation of Indoor-scene Image Segmentation Datasets** Dalian University of Technology, China
  Research Assistant advised by Prof. Xin Yang | Indie Project *Dec. 2018 – Nov. 2019*
  - Generated indoor-scene image datasets for image segmentation model training using **C++** and **OpenGL** with a speed of generating **60+ images per second** from different position and angle.
  - Improved the photorealism of indoor-scene images by developing the generator on **Unreal Engine 4** using Blueprint and C++, and generated over **20000+** images for model training.
  - Validated the generated datasets (simulation images, depth images and ground truth images) using **Python** and Keras; Improved the accuracy of indoor-scene image segmentation model by **13%**.

Projects

- **C-like Language Compiler:** Designed and implemented a C-like language compiler using **C++** with **Qt** user interface, including lexical analysis, syntax analysis, error handling, semantic analysis, interpretation and execution features.
- **Design and Implementation of CPU:** Designed and implemented a static five-stage pipeline CPU that supports about 56 basic MIPS instructions; Responsible for developing Instruction Decode module, ALU module and a testbench program.