Yingtong Yu

Portfolio: JARVISHHH.github.io Email: yingtongyujobs@gmail.com
Github: https://github.com/JARVISHHH

Education

Brown University **Sep. 2022 – May 2024 (Expected)**
Master of Science in Computer Science - GPA: 4.0/4.0 *Providence, RI*

Courses taken: Advanced Computer Graphics, 3D/2D Game Engines

Nankai University **Sep. 2018 – May 2022**
Bachelor of Engineering in Computer Science - GPA: 3.79/4.0 *Tianjin, China*

Technical Skills

Languages: C/C++, Java, Golang, Scala, Python, Shell, SQL
Developer Tools: VS Code, Visual Studio, Goland, IntelliJ, Anaconda, Virtual Box, Vim, Grafana, Qt Creator
Technologies/Frameworks: Linux, Git, Elasticsearch, Thrift, RPC, JavaFX, Flask, SQLite, OpenGL, Engine, GLM

Work Experience

eBay **Jun. 2023 – Aug. 2023 (Expected)**
Software Development Engineer Intern *San Jose, CA*

ByteDance(TikTok) **Apr. 2022 – Jul. 2022**
Back-end Software Development Engineer Intern | Golang *Beijing, China*

- Reduced the latency of the packing part of the Suggestion Middle Page **from 160ms to 10ms** by reducing the number of RPC calls and parallelizing different processes, and increased the speed by about **1500%**.
- Refactored an entire API service, making it more readable and extensible.
- Added metrics and AB test, built **Grafana** dashboards to visualize performance.
- Implemented Pinyin fuzzy search and supported the proximity-based filtering with **Elasticsearch**.
- Integrated the new version of the recommendation engine and provided more informative search bar options, such as property type of real estate, tips to switch cities, etc.

Projects

Ray Tracing and Path Tracing | *C++* **Feb. 2023 - Present**

- Implemented traditional ray tracing and path tracing in C++, from reading scene data to outputting an image.
- Implemented basic features, like reflection, refraction, shadows and etc.
- Accelerated intersection calculation with bounding volume hierarchy(BVH) and k-dimensional tree(k-d tree).
- Implemented phong illumination model, implicit and explicit intersection, soft shadows under area lights, depth of field, texture mapping, super sampling and etc., for traditional ray tracing.
- Implemented four basic BRDFs (diffuse, glossy reflection, mirror reflection and refraction), Cook-Torrance microfacet model, importance sampling, stratified sampling and etc., for path tracing.

Escape - A Game Produced by Self-made 3D Game Engine | *C++* **Feb. 2023 - May. 2023**

- Designed and implemented a 3D game engine, and produced a dungeon escape rogue-like game using the engine.
- Implemented cylinder collision for dynamic game objects and Ellipsoid-Triangle collision for static game objects. Optimized collision checking with bounding volume hierarchy(BVH) and Hierarchical Grid.
- Implemented the A* algorithm based on the navmesh for pathfinding and behavior trees for decision-making.
- Implemented several graphics features based on OpenGL, like particle system, bump mapping, shadows, etc.
- Integrated a basic UI toolkit, including buttons, text and images.

Stylized Caustics: Progressive Rendering of Animated Caustics | *C++* **Apr. 2023 - May. 2023**

- Implemented the techniques introduced in the paper *Stylized Caustics: Progressive Rendering of Animated Caustics*.
- Designed and implemented the workflow framework, designed parameters for new input data and integrated new parameters into the existing scene file format for better readability and scalability.
- Projected generated photons from 3D space to 2D plane, and projected moved photons back to 3D space.
- Implemented greedy algorithm to assign photons to achieve minimal moving energy cost.

Parallelization of Triangular Raster Anti-aliasing Algorithms | *C++* **May 2021 - Jun. 2021**

- Studied the parallelization of a dominant anti-aliasing algorithm and achieved a **7-time** speed-up on the algorithm.
- Added the code of the low-pass filter based on the stencil code of the GAMES101(Assignment 2).
- Parallelized Fourier Transform and the multiplication operation in the frequency domain, using loop expansion, multithreading, multi-node technologies.