Gregory Du

2444 Virginia Street Apt. A, Berkeley, CA, 94709 | (978)-846-2178 | 17GregoryD@gmail.com

Education

UNIVERSITY OF CALIFORNIA BERKELEY (AUGUST 2017 – PRESENT)

- Expected Degree: B.A. Computer Science (Spring 2021)
- Relevant coursework: Computer Graphics (Spring '20), Algorithms, Discrete Mathematics and Probability Theory, Computer Security (Spring '20), Computer Architecture, Data Structures, Structure and Interpretation of Computer Programs, Linear Algebra

Skills & Abilities

- C++, Python, C, Java, PowerShell / Qt Framework, HOOPS Visualize, JUnit Testing / Keras, Pandas, SQlite3, LaTeX, Source Control (Git & Perforce), Windows and Linux development.
- Industrial Software Engineering (functional & design spec., coding, debugging & regression testing).

Experience

SOFTWARE ENGINEERING INTERN | CADENCE DESIGN SYSTEMS, INC. | MAY 2019 – AUGUST 2019

- Developed a variable highlighting mechanism for 3D Canvas in Allegro, Cadence's proprietary PCB design software. New functionality allows the end user to choose from various visual effects to apply to elements in the 3D Canvas. Created an accompanying GUI (Qt in C++ & HOOPS Visualize 3D rendering).
- Developed a machine learning procedure in Python to predict various performance metrics in Allegro's 3D Canvas. Implemented a dense neural net using Keras, a data collection mechanism, and a storage database to accurately predict the load time when using the 3D Canvas.

UNDERGRADUATE STUDENT INSTRUCTOR (DATA 8) | UC BERKELEY | AUGUST 2019 - PRESENT

- uGSI for UC Berkeley's Foundations of Data Science course. Teach labs of around 20 students and manage lab assistants to familiarize students with the fundamentals of data science, including basic data manipulation using the data science module in Python, probability, statistical analysis from large data sets, statistical theory, and basic nearest neighbor classification. Aid students in office hours as they work through various course assignments including homework and projects and attend weekly staff meetings.

Projects

- (In progress) Worked on implementing the material point method to simulate snow. Encoded core data structures and algorithms (particle rasterization, force and velocity calculation, and deformation gradient update) according to the paper authored by Stomakhin et al. Created an XML generation pipeline to render simulated frames using the Mitsuba renderer to create short animations.
- Implemented a path tracer in C++ to compute ray intersection, BVH construction, direct illumination, global illumination with Monte Carlo estimation, and adaptive sampling. Additionally, encoded depth of field and material BSDFs including diffuse, mirror, and glass.
- Implemented features in a rasterizer for scalable vector graphics including in-triangle tests for basic triangle coloration, antialiasing, SVG transformations, color interpolation using barycentric coordinate calculation, and pixel sampling methods for texture mapping including nearest pixel sampling and bilinear sampling. Currently working on adding various level sampling methods using mipmaps.