MACINTYRE SUNDE

Linkedin - Github - msunde@haverford.edu - 802-281-0985

Education

Haverford College: B.A. in Computer Science Graduated: Winter, 2022

GPA: 3.44 / 3.52 in major

Relevant coursework: Analysis of Algorithms, Computer Graphics, Conceptual Art, Linear Algebra, Computer Architecture

Work Experience

Teaching Assistant CS251: Haverford College Fall 2022

- Assisted students with labs and homework on the topics of computer architecture; programming languages and idioms; Haverford Educational RISC Architecture (HERA).
- Collaborated with professors Doughterty and Wonnacott to optimize the learning experience both in the classroom and during TA office hours.

Machine Learning Intern: Haverford College - Professor Rajesh Kumar Summers 2020 and 2021

- Goal: Develop a technique for continuous authentication for use in high security applications. Implemented and compared 1D and 2D CNNs with traditional neural networks with the task of differentiating between users' biometric mouse signatures. Results: The 2D CNN outperformed our other methods

Additional Experience

Archaeological Reconstruction Using Shape Grammars: Spring and Fall 2022

Project advised by Professor Aline Normoyle

- Researched shape grammars as a tool to procedurally generate structures and cities.
- Implemented a version of L-Systems to generate raster images of fractals and trees.
- Designed a grammar in Unity Engine to visualize the pre-Columbian city of Pachacamac (Peru) based on the layout of its foundations.
- This work included designing my own interpreted scripting language and using it to manipulate models in Unity

Noise Tools: Personal project inspired by Sebastian Lague Summer 2022

- Unity package for procedural texture and volume manipulation. I used compute shaders to experiment with different types of noise in 2 and 3D, including simplex and perlin noise generated with hash tables and pseudorandom functions.
- Designed a modular UI to manipulate and add different noise types in realtime.
- Implemented the marching cube algorithm to make meshes out of textures!

Compilers: Course taught by Professor David Wonnacott Spring 2022

- Built a working non-optimizing compiler in C++ for Andrew Appel's Tiger language. Used Appel's "Modern Compiler Implementation in C" as reference and Bison/Flex for the grammar and parser.
- Gained experience working with large pre-existing codebases and balancing the time available versus the level of importance of programming tasks.

Other Skills

Programming Languages: C#, Python, C++ (proficient); Java, HLSL/GLSL (intermediate)
Software: Unity, Visual Studio, Git, Pycharm (proficient); Eclipse, Blender, Photoshop (intermediate)
APIs: TensorFlow, Sklearn, ImGUI, OpenGL (intermediate)