linkedin.com/in/brennen-green
www.brennengreen.dev

Brennen Green

+1 (859) 312-0852
brennengreen@outlook.com

EDUCATION

Lexington, KY **University of Kentucky** **Graduation Date: Dec. 2022**

- **Major:** Computer Science, B.S. (GPA: 3.94, Specialization: **Computer Graphics / Math**)
- **Skills:** C++, C, Python, OpenGL, DirectX, Vulkan, GLSL, HLSL, GLES3, Linear Algebra, Numerical Methods, Computer Graphics, C#, Kotlin, Houdini, Blender, Unity, Machine Learning, Artificial Intelligence, SQL, SVN, Perforce

WORK EXPERIENCE

Graphics Programmer Intern **Blizzard Entertainment** **May 2022 – Aug. 2022**

- Contributed to the development of new particle system using **DirectX, C++, and HLSL** for AAA Game Engine
- Developed **compute shaders** in **HLSL** for multithreaded **simulation** of thousands of particles on the **GPU**
- Used frame capture tools such as **RenderDoc** for debugging and profiling of API events, buffers, and more

Android Graphics Engineering Intern **Twitch / Amazon** **May 2021 – Dec. 2021**

- Owned creator tools that improved time spent streaming by >15% developed with **Dagger, RxJava,** and **Kotlin**
- Pioneered system utilizing **OpenGL ES rendering** pipeline and **GLSL** shaders for rendering dynamic visuals
- Developed automated **unit tests** for MVP architecture to maintain **CI/CD** with JUnit5 and Mockito

Software Engineering Intern **Intel Corporation** **Oct. 2020 – May 2021**

- Analyzed system design using **computer architecture** knowledge to design validation tests in **C** and **Python**
- Validated pre-silicon processor firmware configurations simulated on **FPGA** in **Simics** to detect vulnerabilities
- Developed scripts in **Bash** to streamline validation workflow in **UNIX / Linux** environment

Full-Stack Software Developer **University of Kentucky** **May. 2019 – July 2020**

- Created a **data visualization tools** for data analysis with **Python**, **PHP**, **JavaScript**, and **SQL**
- Revitalized **DevOps** with automated **CI/CD** using **Git** version control / **GitLab** and deploying code in **Docker**

RESEARCH

Computer Vision Research Assistant **University of Kentucky** **Mar. 2020 – May 2022**

- Developed convolutional neural networks with **CUDA**, **Python**, and **Pytorch** for **computer vision** applications
- Developed **deep learning algorithms** for predictive analysis on machine health and aircraft engine failure
- Optimized open-source data visualization software for **real-time rendering** with **parallel computing** / **CUDA**
- Optimized CMOS imaging device using **CUDA** and **MATLAB** GPGPU Toolbox to improve runtime by >60%

PERSONAL PROJECTS

Teapot Game Engine:

- Created a **3D realtime rendering** engine in **OpenGL** and **C++,** with build automation in **CMake**
- Designed interactive GUI using **Dear ImGui** to interact with various engine parameters and the editor
- Profiled optimizations and performance using tools such as **RenderDoc,** and **NVIDIA Nsight**
- Reimplementing the rendering engine in **Vulkan** for better control over the computer **graphics pipeline**

Physically Based Ray Tracer:

- Implemented Monte Carlo ray tracing **3D rendering** algorithm in **C++** that handles multiple **BDRF/BSDFs**
- Distributed ray tracing workload across all cores using **C++ multithreading** and accumulating buffers
- Optimized the code base using acceleration structures such as bounding volume hierarchies

3D Voxel Engine:

- Developed voxel engine in **C++** using **OpenGL** for **physically based animation** and **VFX** experimentation
- Utilized **multithreading in C++** for meshing in parallel, making meshing a low-impact process in realtime

Houdini / Unity / Unreal Engine Projects:

- Created a signed distance field ray marcher in a **HLSL** shader in **Unity** and **C#**

OPEN-SOURCE CONTRIBUTIONS

Godot Game Engine: (Rendering, Core Technology, Engine)

- Fixed automatic generation of **GLSL** shaders in **C++** which caused visual error with normal mapping refractions
- Worked with the **physics** and **rendering** teams fixing bugs and regression issues with **Vulkan** and physics servers

Blender Foundation (3D Computer Graphics Software): (Tools, Editor)

- Used **C++** and **Boost** Library to contribute the development of a higher-precision tool for object selection