Keven Villeneuve

Graphics Software Engineer

✉ keven.villeneuve@gmail.com | ⌂ kevenv.com | ○ kevenv | ▦ kevenv | ✈ @keven_v

Skills

Languages: C++, C, Rust, Python, JavaScript, GLSL, HLSL, x86 assembly, MATLAB, Mathematica, LaTeX.

Libraries: OpenGL, Vulkan, DirectX 12, CUDA, TBB, WebGL, Three.js, Qt, STL, OpenCV, PyTorch, NumPy, SymPy, SciPy.

Tools: CLion, Visual Studio, CMake, VTune, Tracy, Renderdoc, PIX, Nsight, Git, Perforce, Maya, Arnold, Mitsuba, PBRT.

Education

M.Eng. Computer Engineering, McGill University, Montréal, Canada   Jan. 2017 - May 2019
- Thesis: Importance Sampling Polygonal Lights in Participating Media (Advisor: Derek Nowrouzezahrai).

B.Eng. Computer Engineering, Université de Sherbrooke, Sherbrooke, Canada   Sep. 2012 - Dec. 2016
- Specialization in Digital Signal Processing & Machine Learning.

Work Experience

Senior Software Engineer, NVIDIA - Omniverse   Mar. 2022 - Present
- Work on OmniGraph, the visual scripting system for Omniverse.
- Create and maintain the OmniGraph performance benchmarks to automatically detect regressions.
- Create and provide support for the Warp team's performance benchmarks using the same infrastructure as OmniGraph.
- Add Python code coverage support to OmniGraph CI/CD pipeline.
- Investigate and fix performance issues in OmniGraph using the Tracy profiler.

R&D Software Developer, Electronic Arts - SEED   Mar. 2020 - Mar. 2022
- Develop a GPU implementation (DirectX 12, HLSL, Compute) of "Direct Delta Mush", a cutting edge real-time skinning animation technique to significantly improve the quality of characters animation in video games and reduce artists time.
- Collaborate on major improvements to the algorithm to increase the runtime efficiency, leading to a publication at SIGGRAPH.
- Implement spherical lights using a ratio estimator to combine analytical lighting with ray traced shadows.
- Implement GGX environment prefiltering in SEED's real-time research renderer (C++) to improve the realism of shiny surfaces.
- Implement 4D mesh streaming in the renderer to support ongoing research on hyper-realistic characters in video games.

Software Developer, Maxon Computer   June 2019 - Mar. 2020
- Develop new features in Cinema4D, a 3D animation software (C++, OpenGL), to improve various modeling workflows.
- Resolve important software defects encountered by the users and internally on Windows and Mac.

Software Developer (Intern), Autodesk   Jan. 2019 - May 2019
- Develop a new C++ plugin as key support for the Bifrost procedural generation system in Arnold, a film quality renderer.
- Act as the principal communication point between two teams of vastly different backgrounds and development pipelines involved in this project.
- Debug tricky low-level crashes on linux related to smart pointers using custom memory allocators within shared libraries.

Software Developer (Intern), Autodesk   Fall 2014, 2015, 2016
- Develop a new tool in Maya (C++, Qt) using a skinning decomposition algorithm to allow importing complex animated characters into game engines without the usual tedious and time-consuming intermediate steps.
- Optimize the FBX importer of Maya using the Intel VTune profiler, giving 6x performance improvement.
- Optimize a module in Maya using hardware accelerated graphics (OpenGL), giving 4x performance improvement.
- Develop an automated test suite with Python to detect performance and usability regressions in Maya.

Engine Programmer (Intern), Ubisoft   Jan. 2014 - Apr. 2014
- Develop and debug features of a AAA game engine in C++ and C# to meet the changing demands of a game production.
- Collaborate with a team of hundreds of developers, artists and producers to ensure their problems are solved correctly.
- Document the features of the engine in a way that can be easily understood by artists.

Software Developer (Intern), Canadian Space Agency (CSA)   May 2013 - Aug. 2013
- Extend the 3D simulator (C++, Ogre3D) of the ISS robotic arm (CANADARM) to provide various networking functionnalities using boost::asio and the STL.
- Design a robust multithreaded client/server architecture to handle many TCP clients at the same time.

Publications

Direct Delta Mush Skinning Compression with Continuous Examples, SIGGRAPH 2021
<div align="right">2021</div>

Binh Huy Le, Keven Villeneuve, and Carlos Gonzalez-Ochoa

Practical Product Sampling for Single Scattering in Media, EGSR 2021
<div align="right">2021</div>

Keven Villeneuve, Adrien Gruson, Iliyan Georgiev, Derek Nowrouzezahrai

Importance Sampling Polygonal Lights in Participating Media, Master's thesis, McGill University
<div align="right">2019</div>

Keven Villeneuve

Projects

Operating System for PC — Jan 2022 - Present

- Create an OS from scratch for the x86-64 platform, boot using the UEFI firmware on real hardware and QEMU PC emulator.
- Composed of a higher-half kernel with 64-bit virtual memory, physical memory allocator, kernel threads.
- Implement drivers for the PIC interrupts controller, UART and PS/2 keyboard.
- Start to implement support for PCI express and ACPI.
- Optimize the console with SSE/SIMD for fast framebuffer scrolling.

Path Tracer — May 2017 - Present

- Develop a surface & volumetric unbiased Monte Carlo path tracer in C++ accelerated using a BVH.
- Add support for area and mesh lights with adequate multiple importance sampling (MIS) techniques.
- Implement diffuse, Phong, mirror, glass and dielectric BSDFs.
- Implement bidirectional light transport algorithms such as progressive photon mapping, volumetric VPLs and volumetric Lightcuts.
- Create a tests suite in Python to catch regressions by comparing images rendered to a reference renderer (Mitsuba).

GameBoy Emulator — Sep. 2023 - Oct. 2023

- Implement the CPU of the Nintendo GameBoy (8-bit 4 MHz Z80) in C++ with interrupts and hardware timers, passing the core tests suite.
- Implement the MMU and memory bus to emulate the memory mapping of the CPU with the other components (MMIO).
- Implement loading of simple ROM cartridges, MBC0 and MBC1 with banks switching.
- Add an interactive debugger with UI (Imgui) supporting disassembly, step instruction, breakpoints, callstack view, memory view, tilemaps view.
- Start implementing the GPU (PPU), able to render the boot screen.

Master's Thesis — May 2017 - Jan. 2021

- Develop a new importance sampling scheme using Mathematica to improve the rendering efficiency of scenes involving polygonal lights in participating media.
- Importance sample the geometric and transmittance terms of a finite set of oriented point lights at the surface of a polygonal light.
- Extend our solution to polygonal and mesh lights by using ideas from Stochastic Lightcuts and Adaptive Tree Splitting.

Awards

Graduate Excellence Fellowship (GEF), Electrical & Computer Engineering, McGill University.
<div align="right">2017, 2018</div>

Winner of "Most retro hack", McHacks: video game using Myo hand gestures controller.
<div align="right">2018</div>

Second place, McGill Physics Hackathon: 3D waves simulator in Three.js.
<div align="right">2017</div>

Second place, Startup Weekend Montréal: Android app as an "Airbnb" for parking spots.
<div align="right">2014</div>

Additional Experiences

Reviewer, Eurographics 2025.
<div align="right">2024</div>

Reviewer, SIGGRAPH Asia 2020.
<div align="right">2020</div>

Organizer, McGill's computer graphics papers reading group.
<div align="right">2018</div>

Teacher Assistant (TA), ECSE 446/546: Realistic & Advanced Image Synthesis, McGill University.
<div align="right">2018</div>

Student Volunteer, SIGGRAPH 2016.
<div align="right">2016</div>