Kevin Liu

E-mail: lykavin@hotmail.com | Tel: (+1) 412-535-2460 | Website: www.keliu.info

EDUCATION

Carnegie Mellon University — **Pittsburgh, US**
Master of Entertainment Technology, Entertainment Technology Center — *May 2021 (Expected)*

Shanghai Jiao Tong University — **Shanghai, China**
Bachelor of Engineering in Software Engineering — *Jun 2019*

- GPA: 3.67/4.30
- Relevant courses: Computer Graphics, Linear Algebra, Game Designing and Programming, Operating Systems, Computer System Engineering, Human-Computer Interaction

SKILLS

- Programming Languages: C#, C++, HLSL/GLSL, Python, C, Java
- IDE/Toolsets: Visual Studio, NVIDIA Nsight Graphics, Unity, CMake, git, Perforce, Qt, Photoshop, Android Studio

EXPERIENCE

NVIDIA — **Shanghai, China**
Software Engineer Intern, PerfWorks Team (C++), full-time — *Jun 2018 – Feb 2019*

- PerfWorks is a C++ API for collecting low-level performance metrics on NVIDIA GPUs
- Coded, co-designed and enhanced formulas for 5+ graphic metrics on Turing chips; investigated and resolved 3+ discovered metric bugs
- Designed and implemented 5+ D3D tests on GPU units such as ROP and L2 cache to ensure the accuracy of the metrics
- Wrote and deployed Python scripts to parse metric test configurations and generate coverage report automatically
- Facilitated the development of tests by improving the D3D test framework with utility macros, functions and classes

ACADEMIC PROJECT

Fluid Rendering in LabX | Chemistry Experiment Simulation (HLSL + C#, Unity) — **Shanghai, China**
Programmer — *Feb 2019 – Jun 2019*

- Rendered smooth particle-based fluid in real time with the screen-space filtering approach (20K particles, 1080p, 9.02ms average frame time on GeForce GTX 1060 6GB)
- Calculated and applied particle anisotropy efficiently using compute shaders to improve surface smoothness
- Implemented and compared two common filtering approaches, screen-space curvature flow and bilateral filtering
- Identified performance bottlenecks with tools such as Nsight, then reduced bilateral filter running time by up to 27.9% using techniques such as utilizing shared memory and employing loop unrolling

PERSONAL PROJECTS

Hanabi - Flower & Grass | Computer Graphics project (OpenGL + GLSL) — **Shanghai, China**
Personal Project — *Jan 2018*

- Hanabi is a mesh particle system demo that simulates petals and grass with modern OpenGL. Players can generate a vibrant environment with simple mouse interaction.
- Utilized geometry shader to generate individual grass blades; implemented basic light shading with Phong model; optimized performance with GPU Instancing (5000+ mesh particles on GeForce 960M)

Buoyancy Lab | Virtual Reality Teaching Environment (Unity + HTC Vive) — **Shanghai, China**
Programmer — *Mar 2018 – Jun 2018*

- Buoyancy Lab provides a VR environment that can simulate and visualize buoyancy forces
- Implemented physics-based hydrostatic force simulation with the formula $d\vec{F} = \rho g z dS\vec{n}$; improved parallelism with Unity Job Systems and quadrupled the frame rate

AR Graffiti | AR Creativity Software on iOS (Unity + ARKit) — **Shanghai, China**
Programmer & Designer — *Mar 2018 – Jun 2018*

- Eco-friendly graffiti app empowering artists to create street art anywhere, anytime
- Led a team of 3 to build a scalable art board system capable of redoing/undoing with Render Texture; achieved significant frame rate gain in heavy use cases compared to old sprite-based approach
- Designed and coded a spray brush system that can reduce spray flow as the device moves away from the wall

Project Bastion | AR Shooter Game (Unity + ARKit) — **Shanghai, China**
Programmer & Lead Designer — *Sep 2017 – Dec 2017*
Best AR Project, The 1st VR/AR Contest for College Students in China (Top 10%); Finalist (20%), Student Group, 2018 indiePlay China Indie Game Contest

- Project Bastion is a "shoot 'em up" game happening in 3D space: players build their own self-defense bastions, destroy others' bastions and compete for resources
- Designed and implemented 5 different kinds of enemy cubes; utilized custom shaders to achieve special effects such as a transparent force field; implemented the bastion builder for faster iterations and player customization