

HANNAH BOLLAR

hbollar@seas.upenn.edu - 408.242.0252 - hannahbollar.com - github.com/hanbollar

EDUCATION

University of Pennsylvania, Philadelphia, PA

MSE in Computer Graphics and Game Technology

Spring '20

BSE in Computer Science: Digital Media Design

Spring '19

Minor in Mathematics, SIGGRAPH Chairman and Mentor, Freshman Peer Advisor

SKILLS

C++/C | OpenGL/OpenVR/WebGL | CUDA | Java | JavaScript | Git | Eigen | Bullet | Python

Visual Studio | Qt Creator | UE4 | Eclipse | Processing | 3DS Max | Maya | After Effects

EXPERIENCE

NVIDIA : Computer Graphics Software Engineer Intern

Summer '18

- Worked with OpenVR/OpenGL adding controller support to NvWebView sdk samples.
- Used UE4 to refactor aspects of the NvWebView application run on Holodeck Engine.
- Worked on NvWebView hangouts feature for its Holodeck application plugin.
- Liaison between AR/VR UX research and Browsers teams to implement requested NvWebView features (locations: Austin, Durham, Helsinki, Santa Clara, Toronto)

Analytical Graphics Inc : Computer Graphics Software Engineer Intern Spring '18

- Modified JavaScript open-source Cesium library: camera, particle effects, mini game.

University of Pennsylvania : Computer Graphics Researcher

Summer '17

- Ragdoll Simulation under Dr. Chenfanfu Jiang for his "Interfering Forces and Learning Human Utilities" project. C++, Python, and Bullet.
- Point Selection Desktop Application under Dr. Stephen Lane, allows user to select mesh points, intersections, and previous outputs as input for a VR texturing algorithm. C++, OpenGL.

University of Pennsylvania : Teaching Assistant

- CIS 563, Physically Based Animation Fall '18
- CIS 561, Advanced Computer Graphics (Rendering) Spring '18
- CIS 560, Introduction to Interactive Computer Graphics Spring '17 Fall '17 Fall '18
- CIS 110, Introduction to Computer Programming Fall '16
- FNAR 264, Computer Science Workshop in Processing Fall '15 to present

PROJECTS

Game Engine Course Developer

Summer '18 to present

- Aiding an instructor in the development of course material through coding the project.

Flocking Boids: CUDA, C++

Fall '18

- CUDA implementation of the artificial life program created by Craig Reynolds to define boids that simulate birds in flight.

Snow and Jello using the Material Point Method: C++, Eigen

Fall '17

- Group project implementing MPM on APIC grid system.
- Responsibilities: Implementation of particle to grid and grid to particle transfers, calculation of stress deformation for force update, other math help.

Monte Carlo Path Tracer: C++, OpenGL

Spring '17

- Full Lighting Integrator, BVH acceleration structure, multiple importance sampling, global illumination, focal distance, different light-source types, and photon mapping.