

HANNAH BOLLAR

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

EDUCATION

University of Pennsylvania, Philadelphia, PA

BSE in Computer Science: Digital Media Design

Spring 2019

Minor in Mathematics, SIGGRAPH Vice Chairman and Mentor, AWE Orientation Volunteer

The Harker School, San Jose, CA

Spring 2015

LANGUAGES AND SKILLS

C++/C | Java | JavaScript | OpenGL/WebGL | Python | HTML5/CSS | Git | Bullet | Eigen

Qt Creator | Visual Studio | Eclipse | Processing | 3ds Max | Maya | After Effects | Photoshop | Illustrator

PROFESSIONAL EXPERIENCE

Analytical Graphics Inc: Cesium Graphics Software Developer

Spring 2018

Incoming developer intern for the Cesium team at AGI.

University of Pennsylvania: Computer Graphics Research Assistant Donovan Scholar Internship

Ragdoll Simulation under Dr. Chenfanfu Jiang: C++, Python, Bullet

Fall 2017

Ragdoll simulation for Dr. Jiang's "Interfering Forces and Learning Human Utilities" project.

Point Selection Application under Dr. Stephen Lane: C++, OpenGL

Summer 2017

Allows users to select mesh intersections and machine learning outputs. Used to train the texturing algorithm for real objects viewed from the Oculus Rift.

iD Tech: Course Instructor at Stanford University

Summer 2016

Taught lessons in Java with Eclipse, 3D-Modeling with Maya, and block-coding with Tynker.

University of Pennsylvania: Teaching Assistant

CIS 460/560, Introduction to Interactive Computer Graphics

Spring 2017 to Present

FNAR 264, Computer Science Workshop in Processing for Art and Design

Fall 2015 to Present

CIS 110, Introduction to Computer Programming

Fall 2016

PROJECTS

Material Point Method Physics-Based Simulation of Solids and Fluids: C++, Eigen

Fall 2017

APIC based grid system using Eigen and a template based backing for most methods to allow for easy simulation attribute transferring. Handled proper conversion of the particle to grid and grid to particle transfers and calculation of stress factor for force deformations.

Monte Carlo Path Tracer: C++, OpenGL

Spring 2017

An implementation of the Monte Carlo Path Tracer. Uses Multiple Importance sampling, a Bounding Volume Hierarchy acceleration tree, Photon Mapping, and Depth of Field adjustments.

Procedurally Generated Projects: JavaScript, WebGL

Spring 2017

Coded a moving bird's wing, spawning environment, shaders and post-processing effects, crowd simulations, and projects using Perlin Noise, L-Systems, and other procedural generation techniques.

3D Visualization Research Project (Penn Arts Council Grant): JavaScript, HTML

Spring 2017

Connected separately scanned 3D spaces, adding labels, and hiding specific sections of the museum for security reasons or aesthetic purposes. Worked on the project spring semester and supervised a student continuing the project over the summer.

Mini Minecraft: C++, OpenGL

Fall 2016

Group project to develop Minecraft from scratch. Responsibilities: procedural generation of caves below ground [L-Systems], dynamic terrain above ground [Perlin Noise Height Mapping], and implementation of the A* search algorithm for roaming sheep to follow the user.

Mini Maya: C++, OpenGL

Fall 2016

A partial implementation of Maya with a GUI including loading of objects, half edge manipulation with extruding, smoothing, inserting of edges and vertices, and shader manipulations.