Annie (Yue Hua) Lin

Berkeley, CA · ann.yhl@berkeley.edu · (917) 780-5881

EDUCATION

University of California, Berkeley | Berkeley, CA   Expected Graduation: May 2024
B.A. Applied Mathematics - Computer Science.   Cum. GPA: 3.3
Coursework: Computer Graphics & Imaging, Linear Algebra, 3D Modeling & Animation, Computational Color
Activities: Cal Hacks, Society of Women Engineers, Girls Who Code, 3D Modeling & Animation, Game Design & Dev

SKILLS & TECHNICAL TOOLS

Languages: C++, Java, Python, SwiftUI, JavaScript, HTML/CSS, SQL
Frameworks: FFmpeg, OpenColorIO, OpenImageIO, AVFoundation, Core Image, Core Graphics, Kakadu, React.js
Software: Autodesk Maya, Nuke, Adobe Creative Cloud (Illustrator, Photoshop, After Effects), Figma

EXPERIENCE

Software Engineer Intern | Walt Disney Animation Studios   May 2023 – Aug 2023
- Worked as 1 of 5 engineers on the Studio Tools team responsible for maintaining and developing Linux/macOS projects and iPad-based tools that support media playback and the filmmaking process.
- Conducted research and collaborated with color scientists and media engineers to identify, assess, and understand HDR requirements (e.g. bit depth, color space, file formats, etc.) with respect to existing iOS capabilities.
- Introduced and engineered an image compression method in C++ that converts and concatenates 32-bit/16-bit image files to 10-bit motion sequence files using Kakadu, OpenColorIO, and OpenImageIO.
- Developed a Python script to systematically generate H.265/HEVC mp4 movie proxies and H.265/HEVC HLS streaming proxies using FFmpeg.
- Implemented HDR viewing support into iPad playback tool using AVFoundation, Core Image, and Core Graphics.

Project Manager | UC Berkeley Undergraduate Graphics Group   Feb 2023 – May 2023
- Manage team of 4 animators to create a 1-minute 3D animation short across the entire graphics pipeline from pre-production to rendering and compositing.
- Use Autodesk Maya for 3D modeling, character rigging, animation, and effects; Arnold and Renderman for shading, lighting, and rendering; and Adobe After Effects for compositing.

Software Engineer Intern | Rocket Lawyer   Jan 2023 – March 2023
- Worked part-time as 1 of 6 software engineers on the Documents team to monitor, debug, and remedy issues related to Rocket Lawyer's legal document automation web-based application.
- Built new feature on the backend that supplements information about a document during the generation process using Java, Docker, Kubernetes, Google Cloud, Postman, and PostgreSQL.

Data and Graphics Reporter | The Daily Californian   Dec 2021 – May 2022
- Worked with two other student reporters on data-driven reporting efforts, extracting and analyzing key statistical insights from online public records from the U.S. Census and California Department of Education.
- Utilized React.js and the Recharts library to craft dynamic and informative line charts, effectively illustrating the declining trend in Black student enrollment in Berkeley and Alameda public education.
- Recognized as the 2nd place winner for the "Best Interactive Graphic" category in the 2023 Excellence in Student Media competition hosted by the California College Media Association.

PROJECTS

Non-Photorealistic Shaders | C++, GLSL
- Worked on a team of 4 to create an interactive .OBJ mesh renderer and implemented a series of shaders for non-photorealistic rendering, including Toon/Cel, Fresnel, and Voronoi shading.
- Independently developed a shader program for the cool-to-warm shading model as described in Amy Gooch's 1998 SIGGRAPH paper, as well as an edge detection algorithm.

Mesh Editor | C++
- Built a triangle mesh editor engine capable of loading and editing basic COLLADA mesh files by using de Casteljau's algorithm to evaluate and draw Bezier curves and surfaces.
- Added support for half-edge mesh manipulations (edge flipping and edge splitting) and loop subdivision.

Rasterizer | C++
- Built a simple rasterizer that takes in and renders SVG files in an interactive viewer with support for antialiasing via supersampling, hierarchical transforms, and texture mapping, including different pixel sampling and level sampling options for pixel and texel interpolation.