Bryson Lee

500 El Camino Real, Santa Clara CA 95053
brysonhlee@gmail.com - 808 391 5739
www.brysonlee.com

Education

University of Southern California, M.S. Computer Science (Multimedia/Creative Tech) *August 2019 - Current*

Santa Clara University, B.S. Computer Science and Engineering *September 2014 - June 2018*

- Emphasis in Software Engineering, Multimedia Computing, Computer Graphics Systems, Distributed Computing
- STEM National Science Foundation Scholar; SCU ACM SIGGRAPH President and Founder

Expertise

Studio Pipelines, Story/Editorial/Review, Computer Graphics Systems, Asset Management, Production Technology

- **Languages**: Python (most proficient), C++/C (minor proficiency), JavaScript/HTML/CSS, SQL
- **API / Frameworks**: PyQt/Qt, Django, AngularJS, Flask, RabbitMQ, Google App Engine + Cloud Platform, Kaltura
- **Environments / DevOps**: Git, Perforce, Docker, Jenkins, Digital Ocean, RHEL, Grafana, Linux/Windows/OSX

Experience

Blizzard Entertainment – Associate Software Engineer, Research and Development *October 2018 - Present*

- Architecting redesign of Blizzard Animation's core asset management system to improve speed, reliability and performance to support millions of high capacity assets; Rebuilding both client libraries and server API
- Leading initiative to revamp a home-grown render and image post-processing pipeline, including taking ownership over media submission and review systems; desk applications to browse media (video playback); web-based media tools
- Maintaining a render job submission API and packaging application/library used by the entire studio

Blizzard Entertainment – Research and Development Intern *June 2018 - October 2018*

- Reimplemented applications that monitor and manage render jobs to build production media and CG assets (caching, ect.)
- Improved capabilities of existing workflow tools dependent on asset element hierarchy, especially in animation, modeling, texturing, and layout departments reliant on robust element mappings

Lucasfilm, Industrial Light and Magic – Production Engineer, Intern *June 2017 - September 2017*

- Collaborated with Walt Disney Animation Studios on implementation of the *Dpix* software application for management of increasingly complex CGI media; involved multiple layers of post processing
- Developed core API service (Django, REST Frameworks) for automated global studio data transfer and to support remote data access, including a robust suite to stream video to upwards of several thousand clients
- Led rearchitecture of ILM's core media player, *RV*; alongside Pipeline TDs, created new plugin framework

Disney Interactive, The Walt Disney Company – Software Engineer, Intern *June 2016 - September 2016*

- Within the Media Technology Engineering Team, created production tools to manage video and image media, metadata, and transcoding (including metrics, statistical analysis, database management)
- Improved scalability of the tracking tools, including DGraph, a system to track web media including video and imagery
- Developed software to manage reindexing of digital media mappings for in-house web publishing pipelines

Projects

***Pipeworks*: Cloud-based CG Pipeline for Distributed and Remote Collaboration** *January 2018 - Present*

Awarded for Best Project of Senior Design Session, May 2018

Provides teams a cloud-based platform to manage CG art assets, shots, sequences, and scenes to streamline production. Includes a cloud API, SDK, desktop-based SaaS, and new design language.

- Python, PyQt5, Qt5, Flask, Google Cloud Endpoints, Google Cloud Storage, Google App Engine

Disney *Matterhorn* Digital Asset Manager *Summer 2016*

Web-based Image, video, and file manager for digital art assets. Built multiple administrative tools and interfaces for: video transcoding, metrics, diagnostics, meta data/legacy tracking, and indexing helper tools.

- Python, Flask, Google App Engine, Google Cloud Storage, Datastore API

***vcontrol*: A Lightweight Version Control System for Digital Art Assets and Binary Data** *January-March 2018*

Lightweight CLI VCS that is based on a simplified directory-oriented platform. Supports common VCS commands including committing, reverting, fetching, ect. for distributed version control and remote workflow support. Supports noncentral branching.

- Python

**learn more about my work and projects at www.brysonlee.com*