|  |  |  |  |
| --- | --- | --- | --- |
| **BRYSON LEE** | 500 El Camino Real · Santa Clara CA 95053  brysonhlee@gmail.com · 808 391 5739  www.brysonlee.com | | |
| **EDUCATION** | | | |
| **Santa Clara University,** **B.S. Computer Science and Engineering** CGPA 3.5 | | *September 2014 - June 2018* | |
| SCU ACM SIGGRAPH President and Founder  STEM National Science Foundation Scholar  Courses: Computer Graphics Systems, 3D Animation/Modeling, Software Engineering, Distributed Computing | | | |
| **EXPERTISE** | | | |
| Tools Development, Production Pipelines, Computer Graphics Systems, Artist Workflows, Studio Technology   |  |  | | --- | --- | | *Languages* | Python, C++, C, JavaScript, HTML, CSS, SQL | | *Workflows / Platforms* | Linux (RHEL7 + Centos), Perforce, Git + (Github/Gitlab), JIRA | | *APIs / Frameworks* | PyQt/Qt, Django, Docker, Ansible, Google App Engine, AngularJS, Flask, Node.js | | *CG / Studio Tools* | RV, Maya, Houdini, OpenGL | | *Databases* | PostgreSQL, MySQL, OracleSQL, Google Datastore | | | | |
| **EXPERIENCE** | | | |
| **Industrial Light and Magic (ILM / Lucasfilm) - Pipeline Engineer, Intern** | | *June 2017 - September 2017* | |
| * As part of the global Pipeline Engineering department, built tools and created software for ILM's in-house render farm system, asset management infrastructure, media creation pipeline, and data transfer services * Worked closely with Walt Disney Animation Studios in implementation of *Coda* and *Dpix* software for render queuing and media/review libraries, respectively * Created core API service for automated global studio data transfer and remote VFX Supervisor platforms * Led rearchitecture of ILM's core media player (*RV)*; alongside Pipe TDs, created new plugin framework | | | |
| **Disney Interactive - Software Engineer, Intern** | | *June 2016 - September 2016* | |
| * Within the Media Technology Engineering Team, created administrative tools for digital asset management, including video/image media + file metadata (Metrics, statistical analysis, database management) * Improved scalability of the asset manager through design of database helper functions * Designed and implemented digital pipeline software to manage reindex and asset mapping for CMS and art production pipelines | | | |
| **Santa Clara University, 3D Animation and Modeling - Teaching Assistant** | | *September 2016 - Present* | |
| * Taught Computer Graphics fundamentals and the 3D Modeling/Animation pipeline, including Maya and introductory technical direction | | | |
| **Hoana Medical - Software Engineer, Intern** | | *June 2015 - August 2015* | |
| * Engineered tool to analyze wireless data packages from sensory data and to detect anomalies * Achieved 99% message integrity through checksum tool implementation * Built medical monitoring mobile and web application that improved load times by 25% compared to the previous implementations with additional live-updating graphs | | | |
| **PROJECTS** | | | |
| **Distributed Creative Content Collaboration Platform (Senior Thesis)** | | *In Development* | |
| * A cloud-based suite of software that will enable small to medium sized teams to bring their creative ideas into reality by providing a comprehensive, yet modular set of tools to support a modern creative content pipeline * Python (Server Scripting, Core API), PyQt, Web-based technologies (Django, Flask) | | | |
| **Procedural Modeling Tools for Surrealist Art Style (Houdini, C++)** | | *In Development* | |
| **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 | | | |
| **Cubic Lattice Constructs for Better Volumetric Pre-Visualizations** | | | *In Development* |
| * This method creates a lossy optimization for volumetric simulations by building a cubic lattice in a 3D space, and providing a better pre-visualization. * C++ | | | |