

Lane Kolbly

lane@rscheme.org

(512)891-9885

GitHub: @lkolbly

Education

University of Texas at Austin

- Graduating in 2017 with B.S. in Computer Science.
- GPA 3.7, 3.91 in major
- A few recent classes: Embedded Systems Programming, Astronomical Instrumentation (currently taking), Object-Oriented Programming, Computer Architecture, and Operating Systems, just to name a few.

Liberal Arts and Science Academy High School

- GPA 3.9

Work Experience

Civitas Learning

Summer 2015

Worked on big-data summary tool in AWS Redshift SQL. Built usage monitoring tool for company's Redshift clusters.

Sundry Contract Work

Part Time Fall 2013 – Fall 2014, Summer 2014

Various web-based contract jobs ranging from credit card gateways (through authorize.net and USA ePay) to websites that scan Twitter & Facebook posts to recommend products. Dealt with Amazon AWS, Google App Engine (Python/XSL), PHP, and Visual Studio .NET.

Click Security

Summer of 2012 & 2013

Developed a Python cloud platform for distributing software packages. Developed a Python Twisted webserver to distribute WHOIS data from Cassandra DB. Wrote scripts to crawl the Internet to determine patterns behind “malicious” websites and used Disco map-reduce to process that data. Developed skills in statistical analysis, big data analysis, big data storage, corporate presentations, and documentation.

WPEngine

Summer of 2011

Set up Cacti and Nagios to provide advanced server information on the server cluster. Wrote programmer-friendly PHP API layer for the Google Charts API. Developed skills in writing and documenting APIs in a corporate environment. Maintained EC2 instance running for WebPageTest.org

A Few Projects

pillow.rscheme.org

Spring 2011 - Present

Built and continue to maintain personal website and sundry other projects, primarily using MySQL and PHP.

People (Re)mover

Early 2016

Wrote a program that removes people from vacation photos. Given multiple photos of the same

Thank you for reading

Lane Kolbly

thing from the same vantage point, it will remove foreground clutter (people, cars, animals) and create a single photo of the entire monument. At time of writing still in proof of concept phase. Uses OpenCV (from Python) to perform SIFT, nearest-neighbors (for alignment, using FLANN) and final image processing.

World Box

Spring 2015

Wrote an entity-based game engine which uses JavaScript to script the game's entities. The actual game engine is written in C++, and is a server that uses Google Protobuf to communicate to clients and V8 to manage the game's entities.

MCServer/Cuberite Nether Portals

Summer 2015

Implemented Nether portal generation in the open-source Cuberite Minecraft server software. Dealt with pull requests in a large open-source project and worked in preexisting code. Written in C++.

Identify Italy

Summer/Fall 2015

Built a project to answer why Italy is the world's largest elevator market by coupling machine learning to satellite imagery of Italy and other countries to measure how much of the country is composed of urban versus suburban lands. Built in Scala/mllib with the Spark library for fast big-data processing, with a PHP frontend for building training data.

Course OS (Class Project)

Spring 2015

Helped develop operating system for Raspberry Pi (ARM). Led virtual memory team and interfaced to the MMU. C, ARM assembly.

SolarCiv

Late 2014

Built web-enabled game inspired by the Civilization series, but set across the stars so that you must deal with speed-of-light delays to your colonies. This project required a complicated message-delivery system to ensure that the correct entities would get the correct messages at the correct time, while at the same time preventing deadlocks in concurrent code. Written in Go for the server, and AngularJS/ThreeJS/WebSockets for the frontend.

Quadcopter Panorama

Summer 2014

Programmed a Raspberry Pi to feed commands to a quadcopter running Arducopter to take panoramic photos from above the ground. Modified Arducopter to not ignore yaw parameter for "waypoint" command. Arducopter was in C, Pi software was Python.

Lyst

2012 & 2015

Built Myst-like game engine (as well as a Myst-like game based on my high school.) Built a world-description schema using XML with a rendering engine built in a web browser using JavaScript and CSS, along with HTML5 manifest files to preload the large images required for play. The game was web-based on a server built using Python Twisted, Websockets, and MongoDB for saved games.

2015 rewrite: Rewrote rendering engine to use WebGL and ThreeJS and support panoramas.

Blue November

Mid-2011

Built naval warfare game/simulation. Program featured voice-control, text to speech, and scriptable game elements with Python. Used C, C++, OpenGL, Gstreamer, Festival, Sphinx, and Python.

Thank you for reading

Lane Kolbly