Thomas Glezen

702.575.8759 | tcglezen@berkeley.edu

Education

UC Berkeley
May 2021
B.A. in Computer Science
GPA: 3.1

Coursework

Undergraduate
CS 61B: Data Structures
CS 61C: Machine Structures
CS 161: Computer Security
CS 170: Algorithms
CS 182: Deep Neural Networks
CS 184: Computer Graphics
CS 186: Databases
CS 188: Artificial Intelligence
CS 189: Machine Learning
EE 126: Probability and Random Processes
EE 127: Optimization
Stat 140: Probability
Stat 135: Statistics
Data 100: Data Science
Data 102: Data, Inference, and Decisions

Skills

Programming
Proficient in:
Python • NumPy • pandas • PyTorch • SQL

Also coded in:
R • Java • C • Bash • HTML • Swift • JS

Tools
Vim • Jupyter Notebook • Intellij

Other
Git • LaTeX
Docker • Debugging

Links

Github:// tcglezen
LinkedIn:// tcglezen
Website:// tcglezen.com/
Stackoverflow:// tcglezen

Experience

Lab Assistant | CS61B (Data Structures)
Aug 2018 – Dec 2018 | Berkeley, CA
A lab assistant for students taking data structures (CS61B) at UC Berkeley. Some of my responsibilities includes helping students how to:

- use git and resolve their git issues.
- Inheritance/Polymorphism
- how Java inheritance/polymorphism works and how to effectively apply inheritance to class projects.
- how to build searching and sorting algorithms and when to use each type of algorithm.
- understand how run time complexity works and how to calculate the run time of their own algorithms under different environments.

Projects

Path Tracer
Project revolving around light modeling of 3D images.

- Implemented camera ray generation so camera can generate 2D image given a 3D world and a direction/location
- Built a volume bounding hierarchy system to optimize rendering time for tracing path of light rays of 3D models.
- Coded bidirectional scattering distribution function which calculates how light reflects off of different types of surfaces.
- Programmed the model so that it can efficiently trace bounces after a hundred bounces.
- Implemented adaptive sampling in order to better perceive light coming from a single source point.

Language Detection
Developed a neural network that processes sentences and predicts its language.

- Modeled as a naive recurrent neural network which intakes a word at each input layer.
- Includes techniques such as Ensembles and dropout in order to avoid overfitting.
- Performs with an overall of 83% testing accuracy.