Jacob Oaks

jacoboaks.8@gmail.com | (717) 713-9259 | jacoboaks.github.io

EDUCATION

University of Pittsburgh

Dec 2021

BS: Computer Science
GPA: 3.81
Courses: Computer Graphics, OS, Algorithms, Formal Methods, Web Programming, Machine Learning, Data Science, Systems Software, Assembly Language and Computer Organization, Data Structures, Software QA
Dean's Honor List
Honors College

SKILLS

Proficient: Python, Java
Familiar: C/C++, JavaScript, HTML, CSS, SQL, GLSL, Swift, Visual Basic
Python Libraries: PyTorch, NumPy, Pandas, Matplotlib, Flask
Other Libraries: OpenGL, Junit, React
Tools: Git, Docker, Jupyter, Bash

EXPERIENCE

Associate Software Developer

Jan 2022 – Present

Carnegie Mellon University Software Engineering Institute

- **Knowing When You Don't Know (Uncertainty Quantification)**
  - Adapted an existing Python probabilistic object detection codebase towards our project's goals.
    - Built a general data pipeline, enabling our team to train models using field-specific in-house data (a project requirement), increasing customer engagement during demonstrations.
    - Implemented and tested against a collection of metrics that were of direct concern to the project. This allowed the team to discern and select models with higher performance, a KPI for the project.
    - Crafted connective tissue between the outputs of this codebase and the inputs to a stakeholder-facing demo application, allowing more frequent and higher quality demonstrations to garner funding and collaboration.

Emerging Technology Center Intern

Jan 2020 – Dec 2021

Carnegie Mellon University Software Engineering Institute

- **Knowing When You Don't Know (Uncertainty Quantification)**
  - Developed a Python system to generate and run ML training, calibration, and evaluation pipelines, serving as the engine for extensive experimentation that led to a publication at ICLR (top ML conference) – a KPI for the project.
    - Created a system to compile YAML-based DSL for ML into complex DAG workflows, automating much of the tediousness of experimentation and reducing completion time by a factor of three.
    - Automated artifact management with custom hash-based check-in/check-out system for workflow results, improving team collaboration by eliminating duplicate work and enabling intra-team artifact passing.
    - Built up Python tooling using Pandas and Matplotlib to automate large-scale analysis and impactful visualization of results. This allowed for much more agile experimentation and halved the time required to acquire key artifacts for our publication.
    - Optimized dataset-specific pre-processing procedures for over a dozen datasets of interest separately, allowing for the wide scope of experiments that led to a greater chance of acceptance of our publication.

- **A Series of Unlikely Events (Inverse Reinforcement Learning)**
    - Created and deployed an interactive web application (found here) using HTML, CSS, and the React library for JavaScript, demonstrating the applicability of the project's ML algorithm. This demo continues to be shown to both internal and external parties and has ultimately led to over $100,000 worth of funding thus far.
        - Visualized results of our algorithm in an intuitive spatial manner by using react-map-gl (a Mapbox wrapper for React) to display satellite imagery and DeckGL to visualize various data layers overtop of the satellite imagery, facilitating high levels of user understanding and engagement.
        - Collaborated with a design expert to create UI elements that felt responsive, natural, and fit our brand, further increasing accessibility and general interest in the application.
        - Implemented half a dozen interactive layers, keeping UX in mind by including concise descriptions of functionality, using a colorblind-accessible pallet, and using multi-threading where necessary to provide smooth asynchronous animations of multiple evolving UI elements.

PERSONAL PROJECTS

Spacedust

Feb 2021 – May 2021

Code | Demo

- Android spaceship battle game built on top of OpenGL ES, a subset of OpenGL meant for mobile and embedded systems.
- Spaceship is controlled using two custom implemented joysticks. All textures and animations, shaders/lighting calculations, simple physics, and UI elements are custom as well.

Ambulare

Feb 2021 – May 2021

Code | Demo

- Side-scrolling game engine built upon the light-weight java game library (LWJGL).
- Contains all custom components such as: textures, animations, a basic physics engine, shader programs, lighting systems, menus and user interface design, and a sound system using OpenAL.
- Uses a custom serialization format and a data loading process that provides precise and easy-to-understand failure messages to support players creating their own content ("stories").

GeoArt

Mar 2019

Code

- Created an iOS app in Swift for Steelhacks 2019 that systematically generates pieces of artwork derived from the unique latitude/longitude location of the device.
- Artwork generated for two locations will resemble each other semi-proportional to how close the locations are.
- The project won second place in the hackathon.