Raymond Chen

Website: https://gitraymond-chen.github.io/Raymond-portfolio/

Email: chenraymond187@gmail.com GitHub: https://github.com/GitRaymond-Chen

1720 SW 37th Street Apt. 164-B • Gainesville, FL 32607 • (929)-471-4030

EDUCATION

University of Florida December 2024

Computer Science B.S.

Relevant Coursework: *Data Structures and Algorithms, Info and Database Sys 1, Intro to Machine Learning, Operating System, Computer Organization, Computational Linear Algebra, Intro to Software Engineering, Programming Language Concepts, Discrete Structures, Human Computer Interaction, Algorithm Abstraction & Design*

GPA: 3.79/4.00

TECHNICAL SKILLS

Development: Python, C++, JavaScript, Java, HTML/CSS, Unix, ARM, MIPS

Tools and Technologies: React, Django, Matplotlib, Bootstrap, Docker, Unity, LaTeX, SFML, JUnit, Git, Linux

Languages: English (Fluent), Mandarin (Native)

WORK EXPERIENCE

University of Florida – Undergraduate Research Assistant, Gainesville FL August 2024 – Present

- Used NewsAPI to obtain metadata on articles about China and stored into Pandas DataFrame.
- Compared methods of Sentiment Analysis (TextBlob, VADER, Transformers). Determined distilbert-base-uncased-finetuned-sst-2-english to be most effective through testing.
- Tokenized news content using NLTK and applied DistilBERT transformer to assign sentiment labels and scores to each sentence.

PROJECTS

Chess AI Website | React, NodeJS, MySQL, Javascript, JSX, CSS

- Developed a secure Login form React component to store users' unique credentials in the MySQL database.
- Created Board using chessboardjsx integrated with chess.js allowing for move validation, engine integration, etc.
- Designed AI brain: Negamax algorithm in Javascript with a table for optimal placement for each piece type.
- Integrated a chat box with the ChatGPT API to provide personalized chess coaching and interactive banter.

Custom Programming Language | Java, Junit

- Built a Lexer, Parser, and Interpreter for a custom programming language. Implemented deterministic finite automata with features such as while loops, switch statements, and if-statements.
- Iteratively processed strings character by character to generate tokens, ensuring adherence to grammar specifications.
- Constructed an abstract syntax tree from token sequences, validating correct grammar and syntax and facilitating compilation.

IMDB Movie Rank | C++, WinForms

- Processed IMDB CSV file of 45,000+ movies extracting title, year, rating, popularity, revenue, and language and stored within a Movies class.
- Engineered a movie ranking system using Quicksort and Merge Sort algorithms in C++, improving search speed and user experience; performance tested with chrono.
- Features user-friendly year input/output via Windows Form.

Minesweeper | C++, SFML

- Employed object-oriented methods to enable dynamic resizing of the game board via a configuration file specifying width, height, and number of bombs.
- Utilized a bitmap to represent bombs as 1’s and empty tiles as 0’s, enabling the calculation of adjacent bombs for empty tiles displayed on the board.
- Developed a recursive depth first search algorithm for revealing empty tiles, halting when encountering a bomb count/flag tile.