Patrick Amerlan

☎ 630-639-8006 ✉ patrickamerlan10@gmail.com linkedin.com/in/patrick-amerlan github.com/amerlanP

Education

University of Illinois at Chicago **2021 – 2024**
Bachelor of Science in Computer Science
GPA: 3.3/4.0

- Data Structures/Advanced Data Structures, Algorithms, Systems Programming, Software Design, Machine Learning, Framework-based Development, Database Systems, Secure Web App Development

College of DuPage **2019 – 2021**
Associate of Engineering
GPA: 3.1/4.0

Technical Skills

Languages/Technologies: C/C++, Python, Java, JavaScript, HTML/CSS, Dart, SQL, ReactJS, Node.js, Express.js, Flutter

Tools: Git, Linux Command Line (CLI), Google Test, JUnit, GDB

Projects

Habere
JavaScript, ReactJS, Google Firebase, Astro

- Habit-tracking web app that utilizes Astro and React frameworks for modern and responsive user experience
- Integrated Firebase authentication and Firestore database to securely store user data and enable data persistence
- Utilized custom REST API endpoints for data retrieval, storage, and other various functionality

Beer Can Collection Mobile App
Dart, Flutter, SQLite

- Cross-platform antique beer can collection app developed using Flutter framework
- Real data imported from BCCA into SQL database enabling advanced search functionality
- Material design used for a simple, clean, and visually pleasing user experience

Morra
Java, Sockets, JavaFX

- Game with GUI that utilizes multi-threading and networking to enable a responsive, multiplayer experience
- Model-View-Controller architecture to allow for modular development and enables sustainable collaboration
- Game logic controlled by server, data passed between server and clients to maintain consistent game state

Public Transit Database Tool
Python, SQLite

- Retrieves public transit data using Python's sqlite3 library based on user's input
- Multiple different functions to retrieve and organize data in a variety of ways, using Matplotlib for graphing
- Uses object, data, and presentation tier architecture to separate code functionalities

Sliding Block Puzzle Solver
C++

- Given input data for starting configuration of a block puzzle, finds the solution in an efficient runtime
- Utilizes custom breadth-first search (BFS) algorithm that tries all valid puzzle configurations
- Always finds shortest solution if there are multiple solutions to the same puzzle

Work Experience

PD1 Solutions, LLC. **Jul. 2020 – Present**
Installation Engineer

- On-site installation of machine vision systems
- Full assembly of vision system controller
- PC software installation and configuration

Data Annotation **Jan. 2024 – Present**
AI Trainer

- Engaged in code-topic conversation with new AI models
- Extensive testing, debugging, and correction of AI generated code
- Judged between multiple AI models to adhere to best coding practices