Alex Simak Projects

High school

- Java (Highschool Project): Hangman
  - File I/O, User Input, String manipulation
- C (APL): Reverse Polish Notation Calculator
  - Basic arithmetic operations, graphing and a variety of formulas

Semester 1

- Python: Music Library
  - Pull data from DB, store based on genre, year, artist and create playlists based on user input
- Python: Conway's Game of Life
  - Two-dimensional lists, mutability, and creating and calling functions
- Python: Maze Solver
  - File I/O, two-dimensional lists, creating and calling functions, and recursion.

Semester 2

- C++: Urban Heat Islands
  - Calling Multiple functions, 2D arrays, File I/O
- C++: Dragon Wars
  - Classes and instantiation of classes using constructors, vectors, File I/O, makefile
- C++: UMBC Trains
  - Classes/ instantiation of constructors, linked list, I/O, Overloaded operators and enums
- C++: UMBC Shipping
  - Inheritance and Polymorphism, File I/O, Driver Class
- C++: Star Wars
  - Templated data structure (queue or stack), overloaded operators

Semester 3

- C++: Circular Buffer of Circular Buffers
  - object-oriented design, dynamic memory allocation (of arrays) and pointer manipulation
- C++: Sally Forth
  - Build a compiler that starts off working with arithmetic operations, then expanding it to handle stack operations, variables, comparison and logical operators, if statements and loops
- C++: Quadtrees for Life
  - Give a basic understanding of a tree-based data structure
- C++: Median Heaps
  - Binary Heaps, Templates and function pointers
- C++: Incremental Rehash
  - Implementation of a Hash Table

Semester 4

- Machine Code: Bomb Lab
  - Principles of Machine-level programs, general debugging and reverse engineering
- SQL/Python: Linking
  - Links a database schema, to a query language, (SQL) using python.

Semester 6

- Artificial Intelligence: Dijkstra and A* (Python)
  - Implemented Dijkstra's and A* to build a pathfinding algorithm for a terrain mapped as a bitmap.
- Artificial Intelligence: Connect 4
  - Implemented a game playing agent that plays connect four with the Minimax algorithm allowing the player to choose the level of difficulty to play the game.
- Artificial Intelligence: Random Forest Classification
  - Used the Random Forest Classifier in the scikitlearn package in Python in order to predict whether or not someone makes more than or less than $50k.
- C: Chess and Tic Tac Toe (Operating Systems)
  - Implemented a kernel module that can be used to play against the user in a game of tic-tac-toe. This module will implement a virtual character device (with a corresponding entry in /dev that the user can open a file descriptor to), which will be interacted with by the user using normal file read and write operations.

Side Projects

- Python: I was searching for a new car, because I was commuting to school and work and I would constantly check the website Cars.com for any new cars that I preferred. Instead of constantly checking, I created a web scraper that automatically scrapes the website using the BeautifulSoup Library and if a new car, with my desires filters appears, it will send me an email with the new car/s.
- Python: I created a simple game using a GUI for the user interface using pygame, which uses the keyboard as the interaction feature, which the player uses the arrows to dodge obstacles and tries to last longer, the game gets faster as the time progresses which makes it harder.
- MERN: MongoDB, Express, React, Node
  - Created a simple website using HTML/CSS and React. The website took in user input (login information) and was able to link the information with the database which either allowed or denied user permission to the application.
  - https://blog.hyperiondev.com/index.php/2018/09/10/everything-need-know-mern-stack/ (for me)
- JavaScript/Unity: The Impossible Game
  - I developed a mobile game for the android phone which resembled the game Mario. The player would use the touch screen components on their device to move the player through a 2D world to avoid the obstacles coming at them. The objective of the game is to get passed all the obstacles and reach the finish line.
- React/AWS:
  - Created a very basic login page using react that is linked with AWS to publish and maintain the website.
- jQuery:
  - Assisted my father to manipulate his website, using jQuery to automatically adjust the CSS animation and event handling to make it mobile application friendly.
- Machine Learning Experience:

- Very basic model which inputs stock data using pandas data reader, takes over 10 years of data (depending the length of time the stock has been around) and based on that builds and trains a model based on the close price for that specific stock, then represented via a graph showing what the actual data shows vs the expected based on the trained model.