Alfonso T Canady Jr

Software Dev | Game Dev | Instructor

Memphis, TN |alfonso.canadyjr@gmail.com | 901-846-2547

Portfolio | Github | LinkedIn

Software developer with 2+ years of software development experience, uses OOP principles and paradigms regularly and uses efficient algorithms to write clean and fast code. Utilized known algorithms to refactor codebase in Cinilope project resulting in a 10x speedup on load times in application. Experience designing and developing software to model various vehicle and air defense capabilities and concepts used in military training.

TECHNICAL SKILLS

Languages: C, C++, HTML, CSS, JavaScript, Java, LUA, C#, Python
Tools: Linux CLI, Unity, Git, Github, PowerShell, Agile

EXPERIENCE

Integration Innovation Inc – Unity Developer
Dec 2021 – Present

- Develop software and systems for use in training military personnel by modeling various military vehicle and air defense capabilities and concepts.
- Work under SCRUM master to produce clean and efficient code.
- Follow storyboards provided by ISDs (Instructional Systems Designers) and work with SMEs (Subject Matter Experts) to develop content in Unity for use in training military personnel.
- Obtain and maintain DoD Interim Secret Clearance.

ProTec - Japan (Yokota Air Force Base)
Oct 2020 – Jan 2021

- IT Training position
- Installed and maintained electrical systems under the supervision of lead engineer

Cinilope – Lead Software Engineer (Part-Time)
June 2020 - Present

- Designed and developed code for behavioral modeling of drones or other UAVs to simulate their performance in a virtual environment using Unity.
- Developed and maintained mobile applications for Android, Apple, and Windows Holo Lens platforms involving drone flight simulations
- Developed backend C# scripts to enhance program capabilities.
- Refactored existing code and implemented appropriate data structures and algorithms to reduce time and space complexity.
- Refactored existing code to follow proper Object Oriented Programming principles.

Code Crew Code School – Lead Instructor
February 2021 - August 2021

- Developed Full Stack Developer curriculum in conjunction with Seattle's Code Fellows Coding Boot Camp

- Trained Teaching Assistants in teaching techniques and software development practices
- Instructed beginner – advanced CS Courses on Full-Stack software development

Code Crew – K-12 Instructor and Curriculum Designer
May 2019 - October 2020

- Designed and Developed Computer Science Curriculum for K-12
- Developed CS and Teacher Training for K-12 Instructors and Teaching Assistants

EDUCATION

Rhodes College- Memphis, TN
B.S Computer Science (Minor in Mathematics) 3.5 GPA – 2019

AWARDS

Memphis Flyer 20 under 30 award recipient: Memphis Flyer | 20 < 30 The Class of 2022

PROJECTS

Ecosystem Simulation *– Unity / C# / Artificial Intelligence /GitHub Repo*

- Developed natural selection simulation modeling relationship between prey and available food in the environment.
- Created a system for genetic variation allowing set attributes a chance to mutate to the next generation resulting in an agent more "fit" to survive in said environment.
- Utilized OOP Principles

BlackJack_Cpp *– Windows Console / C++ / OOP Principles /Github Repo*

- Console application where two users can bet and play against a dealer in a game of blackjack.
- Used OOP principles: inheritance, encapsulation, polymorphism, and abstraction to develop clean and easily readable code minimizing if/switch statements.
- Utilized user input validation in various locations so that only expected values are allowed to be input, an incorrect value will prompt the user for another until a correct one is given.

Coding Challenges (Featured) *– Java / C# / Github Repo*

- Various programming challenges written in Java to both learn the language and challenge myself to write not only working but optimal and fast Java code.
- Merging two sorted arrays (Doing so in place to save on space in O(m+n) time)
- Determine if an array contains any duplicates (Utilizing a Java HashSet to develop a solution with O(n) time complexity)
- Find the first bad version x in a continuous list of length n where at x all versions > x are bad and all versions < x are good. (Solution uses an implementation of Binary Search to run in O log(n) time)
- Also implemented Linked List / Queue/ and Trie in C# and other challenges in Java.

Additional projects and code available at https://acanady.github.io/#projects