ZACHARY A. KONIK

Berkley, Michigan | +1 (248) 550-3388 | zak@zmbg.us

linkedin.com/in/zachary-konik/ | github.com/zakattack02

Objective

Hardworking individual looking to secure a responsible career opportunity in the Computer/ Software Engineering field, to fully utilize my training and skills, while making a significant contribution to the success of the company.

Education

Clarkson University [August 2021 – Present] [Potsdam, New York] [Computer & Software Engineering]

Work Experience

Technology & Development Intern [May 2023 – August 2024] [NHC Commercial Inc.]

- Developed websites on legacy and historical platforms.
- Provided I.T. support for live stream religious services.

Assistant Teacher [August 2020 – August 2024] [Berkley Building Blocks]

- Led groups of up to six children in activities to teach basic motor skills.
- Supervised children in outdoor activities.

Sales Assistant [2015 – January 2024] [KIFS Collectables — Oak Park, MI]

- Assisted in setting up display cases for merchandise at conventions.
- Oversaw sales of merchandise.

Projects

FPGA-accelerated Minecraft World Generation – September 2024 – December 2024

- Developed an FPGA-accelerated system to generate Minecraft worlds, with the primary objective of determining if a block at a given position should be solid or air. This aimed for faster generation than a CPU.
- Implemented procedural generation of a single chunk of terrain (16x16x384 blocks) based on user-provided seeds.
- Designed a host application as a graphical user interface (GUI) in C# compatible with Windows, macOS, and Linux. The GUI included controls to initiate generation, real-time progress indication (with error relay), a preview pane, and a text pane for logging information from both the hardware device and the application.
- Enabled bidirectional TCP/IP over Ethernet communication with a Zybo Z7-20 FPGA hardware device. The host application accepts a user-specified IP address and communicates on port 9055.
- Integrated custom Vivado IP Cores, including a BRAM Arbiter and four parallel Generation Workers, with a full Arithmetic Logic Unit (ALU) for 32-bit fixed-point mathematical operations [12, 17-19]. The ALU could perform operations such as setting constants, absolute value, square, cube, half/quarter negative, addition, multiplication,

minimum, and maximum. Floating-point math was implemented using 32-bit fixed-point numbers with 1 sign bit, 9 integer bits, and 22 decimal bits.
- Achieved terrain generation including ground and caves. However, the generated terrain was not identical to the official game algorithm's output and did not include ore veins due to a late-discovered design flaw that prevented the implementation of interpolate and cache instructions.
- Applied a simple terrain decoration feature to put dirt and grass atop terrain with empty air above it, which improved aesthetic appeal and resembled older Minecraft world generation styles.
- Utilized the Spigot Minecraft modification framework to export generated data into a game-compatible format. Spigot provides an API to a running server, which significantly reduced development time compared to creating a custom NBT (named binary tag) library. The generated world was accessible and viewable directly from the game client.
- The project used Vivado 2023.1 because Vivado 2024.1 lacked support for the Lightweight IP (lwIP) library examples essential for network communication functionalities.
- Performance-wise, generating a full chunk on hardware took approximately 2 minutes, longer than the target of under 30 seconds, which was partly attributed to poor network conditions during tests.

2048-Game-For-CS470 – Dec 2024

github.com/zakattack02/2048-Game-For-CS470

- This repository contains code for a functional 2048 game for CS470(Deep Learning).
- It is designed to be interactive using keyboard arrow keys.
- The project features all standard 2048 game functionality and can be configured to run in headless mode, displaying the game matrix in the terminal.
- It is implemented entirely in Python.
- implemented NEAT-Python

EE316-Project-5: EE316 Project 5 – Spring 2024 – Apr 2024

github.com/zakattack02/EE316-Project-5

- This project, part of the EE 316 Computer Engineering Junior Lab for Spring 2024, focuses on developing a digital system using the Xilinx ZynQ processor.
- The system is designed to sample 8-bit data from analog inputs on Digilent's Cora Z7.
- It controls the angular position of a servo motor, the intensity of PMOD LEDs, and the speed of a DC motor.
- Key design specifications include utilizing Vivado and SDK tools.
- The project controls a servo motor with a PWM signal (ranging −90 to +90 degrees) and manages PMOD LED intensity using an AXI Timer's PWM output.
- It also incorporates a light-sensitive resistor to control a DC motor and a buzzer.
- Onboard buttons with interrupts are used for system control, and system status is displayed on an LCD.
- An ultrasonic sensor is implemented for object detection and alerts, including colored LED and sound alarm based on object distance.
- Languages used in the project include HTML (50.4%), VHDL (25.5%), and C (18.1%) [35].

GitHub - zakattack02/LLM-DiscordBot – Jul 2025

github.com/zakattack02/LLM-DiscordBot

- This is a versatile Discord bot capable of reading messages from a specified channel, responding to user commands, and performing various AI-powered tasks.
- Its features include image captioning using BLIP-2, object detection with YOLOv8, Optical Character Recognition (OCR) using Tesseract, GIF handling, video thumbnail extraction via FFmpeg, and asynchronous processing.
- The bot supports integration with external AI models.
- The project is written entirely in Python.
- It is licensed under the Apache License 2.0.

STAT383-Final-Project: STAT383 – Dec 2024

github.com/zakattack02/STAT383-Final-Project

- This project is the STAT383 Probability and Statistics Final Project.
- It aims to analyze data related to the 2024 election and make predictions based on voter behaviour and demographic factors.
- It involves an observational study using a dataset collected from a class survey, which includes demographic information and polling data.
- The methodology includes statistical methods such as correlation coefficients, linear regression, hypothesis testing (Z-test for proportions), and normality and assumption tests (F-test, Chi-Square).
- The project's code is written entirely in R .
- This project is licensed under the MIT License.

Extracurricular Activities

Capture The Flag Competitions — January 2021 – Present

- Scaled infrastructure using Google Cloud Platform (GCP) and Digital Ocean with Kubernetes and Docker, deploying custom software.

Eagle Scout Rank — May 2020

- Coordinated and raised $3,000 in funds, materials, and labour for a community service project.

Amateur Radio — K2CC

- Licensed amateur radio operator with experience in emergency communications and RF engineering.

Formula SAE Electric

- Team member developing electric race car systems and embedded vehicle controls.

Clarkson Open-Source Institute

- Contributing to open-source projects and promoting collaborative software development.

Skills

- **Programming Languages**: JavaScript, Python, C/C++, HTML/CSS, Java, Bash, R, VHDL.
- **Technologies**: React/Vite, Git, UNIX, Docker, NGINX, Google Cloud Platform, GitHub, PowerShell, Visual Studio Code.
- **Leadership**: Self-Directed, Quick Learner, Detail-oriented.