

Ben Gaudreau

Professor Levine

COS 420

February 12, 2025

Large Language Models (LLMs) are dramatically changing the ways that people work across a wide variety of fields, and no clearer is this change seen than at the amateur level. At what point should these artificially intelligent tools replace human workers? It is a complex question that encroaches on philosophy, economics, art, and many, many more subjects. One of the fields that has been receiving considerable attention is the one in which the models were born in the first place: computer science. It is a well known fact that CS students and tech startups alike have been relying heavily on LLMs to (hopefully) lessen their workload. Are these tools capable of being not one of many tools to solve problems in software development, but rather be *the* solution to every problem?

My first foray into answering this question began by tasking OpenAI's ChatGPT 4 with a programming project from Kettering University given to first-year CS students. The project in question, titled Bulldog, is a modification of the dice game Pig. As such, it features elements of object-oriented programming, control flow handling, randomness, among other introductory concepts. To ensure unbiased results, I completed the project myself beforehand as a baseline to compare the AI-written code against. With my code complete, I then provided the LLM with the exact same materials that I had access to: a requirements specification document and two sample code snippets as a starting point. Both my program and the one generated by AI were written in Java.

To my surprise, the first program returned by ChatGPT was... somewhat correct, but also incredibly wrong. This initial iteration failed to use the code snippets, rewriting one of them and completely ignoring the other. On top of this, the AI-generated code lacked documentation, error handling, and consistency. As I re-read its code once again while writing this essay, I found that

ChatGPT had used the Math and Random libraries interchangeably for the same purpose of rolling dice! In the few parts that it was consistent, it was consistently ugly. Underneath all the garbage, however, was some semblance of a code structure that would light the way for future iterations. As a human instructing the model on what to do, I could always tell ChatGPT to modify the program in a manner that improved the aforementioned shortcomings and brought the overall program closer to completion. And so I did exactly that.

After 24 further iterations upon the first program generated by ChatGPT, the program had reached a point I considered satisfactory to the original requirements of the project. By the end of the process, which stretched out to several days due to rate limiting by OpenAI, not a single line of code in the final program had been written entirely by me. However, the forced delays made the overall time to complete the project slower with AI, although this only applies to individuals, and not within a business application. The real time-saving element of using AI in programming was its efficiency at writing an initial structure. Although it was hardly correct on the first pass, the structure of the code itself hardly changed by the end; more changes were had in the contents of said structure. A human could easily expand upon an LLM's first attempt, whether manually or by further refining it with AI tools. As it stands, however, ChatGPT is certainly not the silver bullet that business executives are hoping it to be.

These results demonstrate that LLMs such as ChatGPT have the capabilities to solve, with some assistance, programming problems that would be reasonable to expect from a first-year CS student. With the trained eye of someone with greater knowledge in the field, I must continue testing its efficacy against more complex programs to see how capable it truly is in computer science. What's more interesting to me, however, is how usable these tools might be to someone with little to no knowledge in the field whatsoever. For now, it appears that there exists some minimum level of skill and knowledge required to take the initial structure offered by a current LLM's first attempt at a programming problem, and take it to a complete product.