Owen Lasbury-Casey

2/27/2025

COMP/MATH 399

Narrative of Exploration – Project 2

AI chatbots can generate code for program developers who may need help correcting or completing their projects. One question that emerged for me was: *Can AI, as we know today, simplify the process of coding GUI-based programs in any given language?* With this project, which focuses on generating Python programs, I developed a way to generate, run, and save AI-generated Python files for future and sample use.

The reason why I chose to create a program that would generate graphical user interface programs is that GUIs have become more widely used than running a program from a command line. Often, developers, especially beginners, find GUI programming intimidating.

From my experiences with AI, I’ve learned that chatbots like OpenAI’s ChatGPT and Google’s Gemini can generate code in virtually any programming language by processing *tokens*. In AI language models, a *token* is a small unit of text, such as a word, part of a word, or punctuation, that the model reads and generates. For example, asking a question like “What is the capital of France?” gets broken down into tokens, and the AI uses those to predict the answer “The capital of France is Paris.” Similarly, code is processed as a sequence of tokens that represent the syntax and structure of a programming language, allowing the AI to understand and generate code effectively.

I also explored the ability to upload existing code that may need corrections or updates. In many cases, AI chatbots can analyze this code and return improved or corrected versions, making them valuable tools for debugging and learning.

Just like classifying images, AI could potentially classify or interpret a GUI program by analyzing the number of pixels and dimensions that define the size of the interface. When it comes to drawing shapes, placing text, or adding buttons, the AI would operate within certain spatial constraints, limited by the resolution or layout dimensions, like how image recognition models process pixel boundaries. This spatial awareness ensures that elements are placed logically and do not overlap or exceed the visible area of the interface.

The finishing result is a tool that empowers students, hobbyists, and even experienced coders to prototype rapidly, learn by example, or generate boilerplate code effortlessly. Looking ahead, features like voice input, more sophisticated layout engines, support for other languages (like JavaScript or Java GUIs), and integration with cloud deployment tools are on the roadmap.

What started as a simple "what if?" became a fully functional AI-assisted GUI generator. And the journey isn't over—it’s just the beginning of blending human creativity with machine intelligence to reshape how we write software.