Course:

BDTF4000 Craft your Prompts – AI magic for all

Assessment Brief:

You are tasked with developing a small software application using AI-powered coding tools (e.g., GitHub Copilot, OpenAI's Codex) to assist in the development process. You may choose from one of the three project options below.

Description:

Tic-Tac-Toe (with GUI)

Description: Develop a GUI-based Tic-Tac-Toe game where users can play against each other (two players). The game should have a 3x3 grid and provide visual feedback when a player wins or if the game results in a draw. You may also include a feature to reset the game.

AI Used:

ChatGPT

GitHub project link:

https://github.com/Crazy-Squirrel-cpu/AI-Assisted-Coding-Project

How AI was used, throughout all stages AI was used in order to provide extra assistance, when it came to debugging or troubleshooting very specific problems I often used stack overflow, for other general problems, AI was used.

Planning

Planning was relatively easy, choosing tic tac toe or my assessment meant that I didn't need to carry out any data analysis or to gather and display information from the provided data sets.

Planning for the game had a heavy focus on presentation and overall flair, the base game was planned to have various quirks and features that add to the game feel and theme. The game itself isn't overly complex and so I assumed that getting the actual game itself to work wouldn't take much time, one or two prompts at most.

Coding

The coding for the game was completely reliant on the AI, the AI was able to provide a working game of Tic Tac Toe in just one prompt. Coding went relatively easy early on, I assume this was due to the low number of lines in the code, as a result there were less errors and there was more chance to branch out and add in many different features without much problem.

The middle of the development was where some errors started to form, as the code itself started to get larger problems outside of the AI started to have an effect. ChatGPT has a limit on your interactions and how much you can use it for reading and writing code. During this period of development, I had hit that limit twice, this didn't help my workflow, I had to use different accounts and copy and pasted the code from another session into this new one. Inevitably, I encountered problems, the AI had the code, but it didn't have the context from the previous session, because of this, the AI would often write out code without integrating it into the already written code, it would also make up new functions or variables without paying attention to the already defined variables.

Toward the end of development there was a certain dread now when it came to using ChatGPT, the code was now over 240 lines long, copy pasting code into the program was no longer feasible without preparing to switch accounts after 4 or 5 prompts to the AI, at this point I had to change how I prompted the AI and instead asked for the solution to be sent on it's own without including the rest of the code, after getting the solution I'd ask for help on placement and then I'd look for errors and then run the code.

This method was a lot more efficient for taking up less of my provided tokens for ChatGPT which meant I could have a lot more time in a session without switching accounts or AI's. A realisation was that it would've been a lot easier to make use of this method from the start and so this is something I've taken onboard and will now make use of in future uses of ChatGPT and AI in general. Another realisation is that the confusion of the AI when I'd started a new session and used the code from a previous session could be solved through more detailed prompts that stress that the AI should stick to previously stated variables and values.

Reflection

Reflecting on AI there's a lot to think about in terms of efficiency and how it affects the productivity of the project. Personally, I consider myself a beginner in Python and do not currently have the skill to create my own version of my AI-assisted project, and even if I had no knowledge of programming or coding at all I would still be very able to create something similar than what I've done here. It's concerning to think that someone with no experience or background in this field could outperform me with this tool.

There are pitfalls in overreliance on AI, whenever I'd gotten errors or problems in the later stages of development, I felt unable to properly troubleshoot problems due to my inexperience and the fact that I did not have a good understanding of the code itself and how it all worked. Handing over all the construction of the code to AI can lead to uncertainty due to being unable to properly plan through all stages of development. You can always be very specific in your prompts but you'll always be at the mercy of what the AI outputs, an example of this would be a problem I experienced when using a background for the game grid, I provided a picture as reference to be used in the

background of the game and instead the AI scaled the image down and put it underneath the game grid itself.

AI is shaping up to have a big impact on the future of programmers and computer scientists in all fields, the ability to have a highly intelligent assistant to help you with problems in a few moments makes AI assistance an incredibly useful tool. AI is great for programming, questions on syntax or functions in situations that are specific to you that you might not be able to find answers for online. The convenience of AI cannot be stated enough, there are now websites and mobile apps that freely provide this service and with the rise of Artificial Intelligence, they will only grow to be more effective.

AI can be a great tool for growth as a student or it can be a crutch that you must constantly lean on in order to finish work, each approach to AI comes with the problem is that it seems to be a very sturdy crutch.