Assignment #2 : Programming for AI

Due November 17th, 8:30am [20% of total grade]

Question 1: Mandelbrot Set (45%)

Part I

Research the Mandelbrot set. If you are unfamiliar with complex numbers, you will need to learn about them as part of your research. After reading and understanding how the Mandelbrot set is defined, write a description **in your own words** explaining how the set is formulated. Additionally, provide pseudocode outlining an algorithm to display the set.

Part II

Write a program to display the Mandelbrot set using any graphics library of your choice. Libraries such as Matplotlib work well for this purpose.

Part III

Implement a feature to allow the user to change the scale at which the set is displayed.

Question 2: Mastermind Game (45%)

For this question, you will implement the classic board game Mastermind in Python. Mastermind is a code-breaking game where the player attempts to guess a secret code generated by the computer. Your program should handle game logic, user input, and provide feedback after each guess.

Game Logic

- The computer randomly generates a secret code consisting of colors (choose from Black, White, Red, Blue, Green, Yellow).
- The player must guess the secret code within a limited number of attempts (default: 15).
- Implement a text-based interface for the player to enter guesses.
- Display prompts and messages to guide the player.
- After each guess, provide feedback:
  - Number of colors in the correct position (exact matches).
  - Number of correct colors in the wrong positions.

Example:
If the secret code is [Red, Blue, Green, Yellow, White, White] and the player's guess is [Red, Yellow, Purple, Green, White, White], your feedback should indicate three exact matches (Red at the beginning, and White, White at the end) and two right colors (Yellow, Green).

End the game when the player correctly guesses the code or uses all attempts, and display appropriate win/lose messages.

Question 3: CHIP-8 (10%)

This question involves research rather than programming. Learn about the CHIP-8 emulator by finding reliable resources online (collaboration encouraged).

- Find and download a Python CHIP-8 implementation.
- Run the emulator with a ROM (even just a test ROM is okay).
- Submit a short video demonstrating the emulator running on your machine.

Submission Instructions

- Upload two .py files (the Mandelbrot and Mastermind programs) in a single .zip file using standard Windows compression (PKZIP), not 7-Zip or other tools.
- Each .py file should include test calls so it can be run as-is to display output.
- For Question 3, submit a short video : describe the resources you found, the emulator you used, and a screenshot of the emulator running.

Rubric

Criterion	Exceptional (90-100%)	Satisfactory (65-89%)	Needs Improvement (<65%)	Weight
Mandelbrot - Research & Pseudocode	Clear, accurate explanation; well-structured pseudocode; shows solid understanding of set and complex numbers	Mostly clear explanation; pseudocode mostly correct but some gaps; shows general understanding	Explanation incomplete or inaccurate; pseudocode missing or incorrect; gaps in fundamental understanding	15%
Mandelbrot - Program Implementation	Correct, efficient program displaying Mandelbrot set; uses graphics well; clean code and comments	Program works with minor issues; some code or style improvements needed	Program limited or incorrect output; poor code structure or documentation	25%

Criterion	Exceptional (90-100%)	Satisfactory (65-89%)	Needs Improvement (<65%)	Weight
Mandelbrot - Interactive Scaling	Smooth, intuitive scale control; user interaction works flawlessly	Scale control implemented with minor usability issues or bugs	Scale feature missing or not functioning	5%
Mastermind - Core Logic & Game Flow	All game rules correctly implemented and enforced; input handling robust; win/loss conditions clear	Most game logic implemented; minor issues or missing edge cases; input mostly handled well	Incomplete game logic; missing features or broken input handling	25%
Mastermind - User Interface & Feedback	Clear prompts and messages; feedback accurate and informative; user experience polished	Generally clear prompts and feedback; minor issues with clarity or correctness	Confusing or missing feedback; poor user prompts and interface	20%
CHIP-8 Research & Demonstration	Finds diverse, reliable resources; emulator runs successfully; clear video and documentation	Finds some useful resources; emulator runs with minor issues; documentation somewhat clear	Limited resources; emulator not demonstrated; poor or incomplete documentation	10%