Generative AI Usage Documentation

Overview

This document details the use of generative AI (GitHub Copilot) in the development of this traffic light simulation project.

AI Assistance Details

Tools Used

- GitHub Copilot

Prompts and Outputs

1. Initial Implementation

Prompt: "Implement a traffic light simulation in Python with these requirements: [list of requirements]"

Output: Initial class structure and basic implementation was generated.

Modifications Made:

- Restructured the code to better separate concerns

- Added comprehensive docstrings and type hints

- Implemented proper state management

2. Test Case Generation

Prompt: "Generate unit tests for the TrafficLight class"

Output: Basic test cases were provided

Modifications Made:

- Enhanced test coverage

- Added edge case testing

- Improved test assertions

3. Bug Fixing

Prompt: "The traffic light transitions are not working correctly when time_remaining is 0"

Output: Suggested fixes for the update() method

Modifications Made:

- Implemented proper state transition logic

- Added immediate transition handling

- Ensured time_remaining is properly updated

Percentage of AI-Generated Code

Approximately 40% of the final code was AI-suggested, with significant modifications and improvements made to:

- Ensure correctness

- Improve code quality

- Add documentation

- Handle edge cases

Learning Outcomes

1. Effective use of AI as a pair programming tool

2. Importance of understanding generated code

3. Need for thorough testing of AI-suggested code

4. Value of code reviews for AI-generated code

Ethical Considerations

- All AI-generated code was reviewed and understood

- Proper attribution is given to AI assistance

- Final implementation represents original work with AI assistance

Conclusion

Generative AI was a valuable tool in accelerating development, but human oversight and testing remained crucial for ensuring a correct and robust implementation.