

Functional Requirements

1. The system shall allow loading BasicML programs from .txt files from any directory.
2. The system shall allow saving BasicML programs .txt files into any directory.
3. The system shall have a solution explorer to select txt files that have been created or loaded in.
4. The system shall load text from selected .txt file into instructions editor
5. The system shall allow editing of loaded programs.
6. The system shall validate code from instructions editor by ensuring it matched BasicML syntax.
7. The system shall provide a Run button to execute BasicML code from editor
8. The system shall display system output such as errors or confirming operations
9. They system shall display selected text file in instructions editor
10. The system shall visually indicate execution status in console (running/waiting for input).
11. The system shall execute all BasicML operations:
 - a. READ (10) for input from the user to specified location
 - b. WRITE (11) for output to the user from specified location
 - c. LOAD (20) for loading specified location value into the accumulator
 - d. STORE (21) for storing accumulator value to specified location
 - e. ADD (30) for adding specified location to accumulator
 - f. SUBTRACT (31) for subtracting specified location from accumulator
 - g. MULTIPLY (32) for multiplying specified location by accumulator
 - h. DIVIDE (33) for dividing accumulator by specified location
 - i. BRANCH (40) for branching to specified location while running
 - j. BRANCHNEG (41) for branching to specified location while running if accumulator is negative
 - k. BRANCHZERO (42) for branching to specified location while running if accumulator is 0
 - l. HALT (43) for stopping the run of the program
12. The system shall handle user input via console for READ operations.
13. The system shall display output from WRITE operations in the console.
14. The system shall maintain and display accumulator state during execution.
15. The system shall highlight the currently executing instruction.
16. The system shall report all errors without crashing.
17. The system shall display memory contents in a table format that updates during execution.
18. The system shall truncate anything above 100 lines of text in the instructions editor
19. The system shall allow user to enter primary and off colors in hex
20. The system shall allow user to invert colors with toggle switch.

Non-Functional Requirements

1. The system shall execute programs in less than 10 seconds.
2. The system shall have an intuitive user interface
3. The system shall be compatible with Windows (.NET 8.0+).
4. The system shall respond to user input within 1 second.