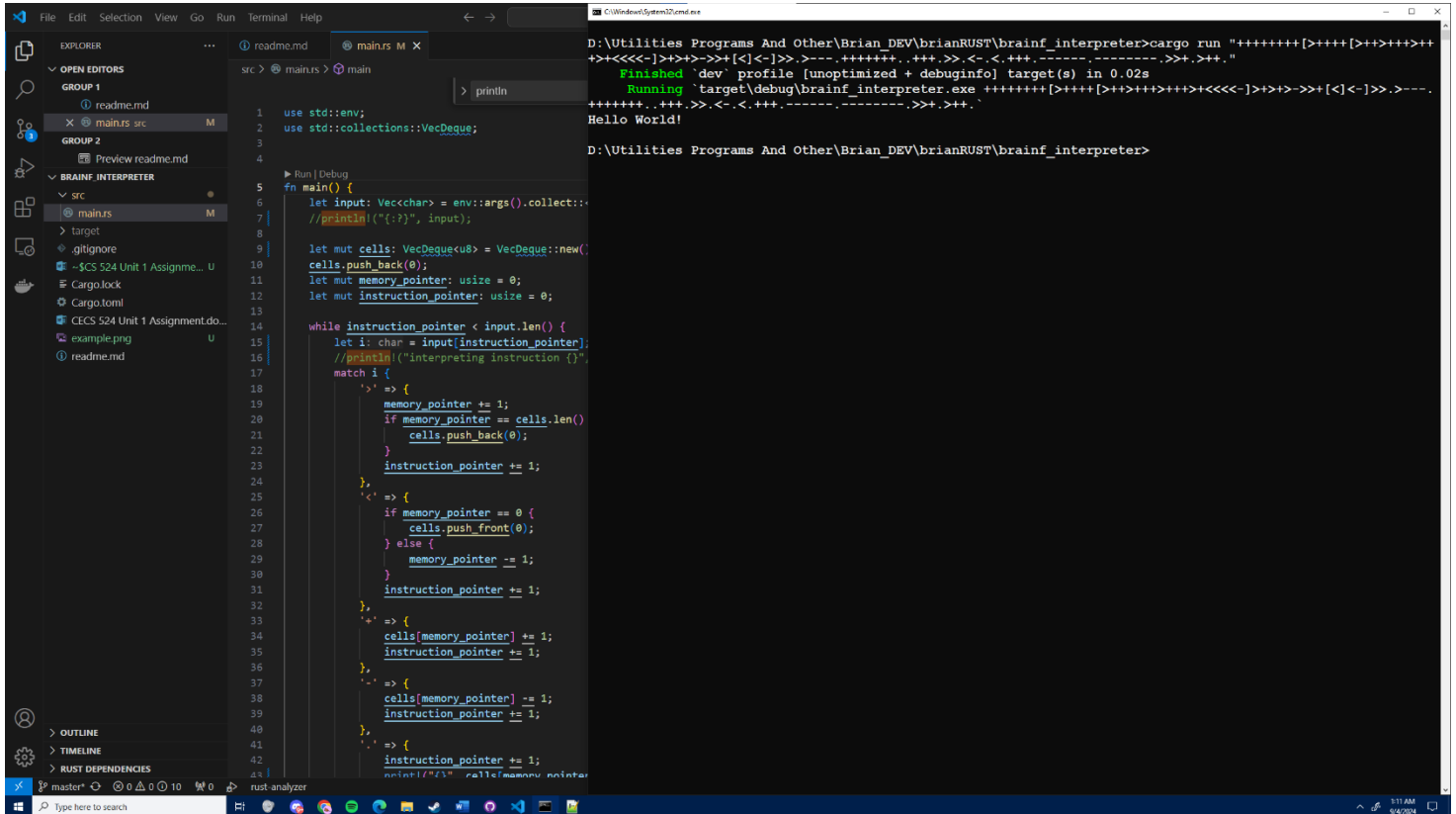


Brian Chilin Stevenson



The screenshot shows a Rust IDE with a project named 'main'. The source file 'main.rs' contains a Brainfuck interpreter implementation. The code defines a memory pointer, an instruction pointer, and a set of cells. It implements the Brainfuck commands: '+', '-', '<', '>', '<.', '>.', '[', and ']'.

```
1 use std::env;
2 use std::collections::VecDeque;
3
4
5 fn main() {
6     let input: Vec<char> = env::args().collect();
7     //println!("{}", input);
8
9     let mut cells: VecDeque<u8> = VecDeque::new();
10    cells.push_back(0);
11    let mut memory_pointer: usize = 0;
12    let mut instruction_pointer: usize = 0;
13
14    while instruction_pointer < input.len() {
15        let i: char = input[instruction_pointer];
16        //println!("{}", "interpreting instruction {}", i);
17        match i {
18            '+' => {
19                memory_pointer += 1;
20                if memory_pointer == cells.len() {
21                    cells.push_back(0);
22                }
23                instruction_pointer += 1;
24            },
25            '-' => {
26                if memory_pointer == 0 {
27                    cells.push_front(0);
28                } else {
29                    memory_pointer -= 1;
30                }
31                instruction_pointer += 1;
32            },
33            '<' => {
34                cells[memory_pointer] += 1;
35                instruction_pointer += 1;
36            },
37            '>' => {
38                cells[memory_pointer] -= 1;
39                instruction_pointer += 1;
40            },
41            '.' => {
42                instruction_pointer += 1;
43                println!("{}", cells[memory_pointer]);
44            },
45            '[' => {
46                // TODO: Implement loop
47            },
48            ']' => {
49                // TODO: Implement loop
50            },
51            _ => {
52                // Invalid command
53            },
54        }
55    }
56 }
```

The terminal output shows the execution of the program with the Brainfuck input: `++++++ [>++++ [>+>++++>++++><<<<-] >+>->+ [<] <-] >> .>--- .+++++++ . .+++ .>> .<- .< .+++ .----- .----- .>> .>+ .`. The output is `Hello World!`.

CECS 524 Unit 1 Assignment

Programming - Due Sep 11, 5pm to Canvas.

Write, in the programming language of your choice, an interpreter for the language Brainfu*k. This is a Turing complete language. It is very simple, on the order of a late first semester of programming assignment. Did mine in Java, about 60 lines of code. Run it with this input below - what is the output?

+++++++ [>++++ [>+>++++>++++><<<<-] >+>->+ [<] <-] >> .>--- .+++++++ . .+++ .>> .<- .< .+++ .----- .----- .>> .>+ .

The Wikipedia page has enough information on the functions of the commands to complete the assignment.

Turn in your source code and a screen shoot of the output and the IDE behind it (or surrounding the output).

Mac users! Please do not submit zipped directories of your program. And do not use Apple based file formats.