Pat Letts

Senior project proposal

F17 CS 480

Therapist: a Brainf\* interpreter

**Language overview**

Brainf\* is a very simple esoteric language created by Urban Müller. Simple beyond the point of usefulness, with only 8, single-character commands: <>-+,.[]; these commands allow for the navigation and changing of an array and its contents, the unofficial standard being at least 30,000 cells in size, all initialized to zero. ‘<’ and ‘>’ respectively decrement and increment a data pointer to track a program’s point in the array. ‘+’ and ‘–‘ respectively increment and decrement the value in the current cell. ‘,’ and ‘.’ allow for single-byte input and output of the current cell. ‘[‘ and ‘]’ allow for looping. All other characters should be ignored.

Here’s a Hello World:

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

I’d like to build a decent interpreter for Brainf\*. Given that it’s especially simple to implement, a good portion of my focus (and code) is going to be focused around particular extra features.

**Challenges and considerations**

The first challenge that comes to mind for an utmost basic implentation of Brainf\* is looping, especially nested loops. My current plan is to use a stack to keep track of loops, storing the locations of left brackets for proper jumping.

There’s a lot of variation between implementations; how long should the array be? Urban Müller’s implementation was 30,000 cells. How many bits in a cell? Some support 8 bits, others would argue for 32. Signed or unsigned? Should I/O always be considered ASCII values instead of raw integers? And so on. Luckily, a number of common conventions have been established regarding these questions and more.

**Goals**

I plan for the majority of my code to use C++, though I may very well end up investing some code in a scripting language for tasks such as compilation and packaging. Naturally, I hope to learn something about interpreter design, and I’m also really interested in learning how to package software for say, a Linux repository, so it can be downloaded and setup directly through a package manager. In a more general sense, I also hope to become more familiar with the intracacies of C++, and other languages I may end up using. I’d also like to implement some graphical features for the debugger, ideally with the GTK library. I’ve never worked with it before and it would be a nice chance to teach myself an entirely new library.

**Features**

Basic interpreter – provide a source file to be interpreted as a Brainf\* program.

Simple options – switch certain options for the interpreter with flags. Un/signed bytes, array size, and

ASCII I/O.

Debug mode – a particular feature that I haven’t seen yet would be a cool graphical debug mode, showing the states of particular stacks, and stepping through code, allowing the user to watch the array change, one command at a time.

REPL – I’d also like to work on a simple REPL shell. Considering the nature of Brainf\*, immediate testing and feedback would be nice to have.

**Grading**

Total points – 100

Basic interpreter setup – 40

-Without nested looping – 30

-With – 40

Debug mode – 50

REPL shell – 20

Simple options – 10 ~ define ASCII I/O, un/signed cells, and cell size

Man page – 10

Auto-install setup – 10

Scale – out of 100 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **F** |
| >90 | >70 | >50 | >=40 | <40 |