DrBrainf*ck – Changes from original proposal

Claudio Maggioni   Tommaso Rodolfo Masera

December 15, 2018

Contents

1 Interpreter 1

2 CLI 1

3 GUI 1

4 Language 1

5 Libraries 2

1 Interpreter

The interpreter has been implemented as planned. All interpreter logic resides in `interpreter.rkt`. This section is entirely functional and uses as the basic data structure the type `ProgState`. The most exotic element in the interpreter is the use of callback functions as arguments in order to handle input asynchronously. The test suite is complete in this file.

2 CLI

A CLI version of the interpreter is available running the file `cli.rkt`. The code uses the `command-line` function from *Racket* in order to parse CLI flags and arguments. Since the entire implementation relies on imperative elements (like `read-line` and `display`) we were unable to build a function-to-function test suite.

3 GUI

The GUI has been implemented with `#lang racket/gui`. The implementation is mostly imperative (since that is the paradigm `racket/gui` is built on), and so the test suite is limited to `bf-lexer` and helper functions. Advanced editing functionality has been implemented mostly through `racket/gui` editors and the library `framework`.

4 Language

The languages `racket/gui` and `racket` have been used instead of ISL due to greater flexibility and functionality.

5 Libraries

The most important libraries used are:

racket/gui used to implement the GUI and the editor functionality;

framework used to add line numbers to the editor and to add syntax highlighting;

rackunit used to replace the *SL `check-…` functions as a testing suite.