Greenfoot-CLI Documentation

This is an attempt at writing a Command Line Interface for Linux with Greenfoot.

Usage

You can clone the repository and open the project.greenfoot file with Greenfoot. Alternatively, you can download the entire Jar and execute it with Java.
After you start the program, press run to be able to use it.
The program can correctly interpret any non-extended US-ASCII input without a modifier key, and with shift as modifier, the input is interpreted as the corresponding character on a US ANSI QWERTY keyboard.

1 Commands

Commands are not case-sensitive, but their parameters might be.

1.1 brainfuck

Usage: `brainfuck [Program]`
This command can interpret a brainfuck program that is given as a parameter.

1.2 push

Usage: `push [string]`
This command pushes a string onto the stack.

1.3 pop

Usage: `pop`
This command pops the current String on the stack and prints it.

1.4 echo/print/println

Usage: `[command] [String]`
This command prints a string.

1.5 exit/stop

Usage: `[command]`
This command stops the execution of the program.

1.6 clear

Usage: `clear`
This commands clears the interface.

2 Development

2.1 cli.java

Contains the World-initialization and Greenfoots act method.

2.2 lib.java

Library of the program. Contains the following static public functions and data structures:

2.2.1 cliMap: `final static char[][]`
Contains the status of each field of the interface.

2.2.2 buffer: `static char[]`

Buffer for keyboard input.

2.2.3 cursorX: `static int`

The position of the cursor.

2.2.4 pwd: `static List<Character>`

Contains the position in the filesystem, but there is no working command to manipulate it at the moment.

2.2.5 pwdA: `static char[]`

pwd as array.

2.2.6 bufferOverflowProtection: `static int`

Number that specifies by how much the buffer must be shorter than the length of the interface to avoid an ArrayIndexOutOfBounds Exception.

2.2.7 prefix: `static List<Character>`

List that contains the prefix of the input line.

2.2.8 println(String):

Prints a String with a newline.

2.2.9 print(String):

Prints a String.

2.2.10 commandCheck():

Splits the buffer into command and parameter and executes Commands.commands with these variables.

2.2.11 newline():

Generates a new line and writes the input buffer onto the interface.

2.3 Util.java:

Utility commands used in the program.

2.3.1 enum OS/getOS():

Checks the OS of the machine running the program.

2.3.2 asciiToInt(char):

returns the US-ASCII value of the input.

2.3.3 brainfuckSyntaxCheck(String):

Checks the input for invalid brainfuck syntax.

2.3.4 isBetween(int, int, int):

Returns wether or not the first int is between the second as the low end and the third as the high end.

2.3.5 spaceBefore(int, char[]):

returns the number of spaces before the next non-space character. The int is the startpoint of the check, the target is the char array.

2.3.6 commandLength(int, char[]):

returns the number of continuus non-space characters after the startpoint. The int the startpoint, and the char[] is the target.

2.3.7 spaceAfter(int, char[]):

returns the number of spaces before the next non-space character. int is the startpoint and char[] is the target.

2.3.8 parameterLength(int, char[]):

returns the number of characters after the startpoint. int is the startpoint and char[] is the target.

2.4 Commands.java:

Contains the commands that can be executed from the interface.

2.4.1 commands(`String, String`):

Executes the interface commands. The first Sring is the command name and the second one is the parameter.

2.5 Stack.java:

Contains the stack and its functions.

2.5.1 stack: `static String[]`

The stack itself.

2.5.2 stackPointer: `static int`

The stack pointer.

2.5.3 push(String):

Pushes the String onto the stack.

2.5.4 pop():

Pops the String and prints it.

2.6 BrainfuckInterpreter.java:

Contains the brainfuck interpreter.

2.6.1 brainfuckInterpreter(String):

The interpretation program.

2.7 Objects:

2.7.1 Cursor.java:

The cursor object.

2.7.2 Font.java:

The font object.