Software Engineering Project

- Bug World Simulator Manual -

P. Baumann, Constructor University, Spring 2023

NAME

sim -- simulate a world populated with bugs

SYNOPSIS

sim [*options*] *file.world red.bug black.bug*

DESCRIPTION

sim simulates a world populated with bugs. The bugs are competing for bits that they are collecting. **sim** loads an environment *file.world* and populates it with red and black bugs, whose descriptions are coming from the files *red.bug* and *black.bug*, respectively. It then starts the simulation for 1000 cycles. The presentation of the simulation is controlled by *options*; without any option, the simulation produces no output.

OPTIONS

-n *int*, **--cycles** *int* Simulate for *int* cycles, rather than the default 1000 cycles.

-x, **--swap** Swap the roles of red and black bugs; this is equivalent to reversing the last two file arguments of the command line. Swapping roles helps to determine the advantage an asymmetrical world gives one bug over the other.

-s, **--stats** Display statistics on stdout after the simulation finished; this includes the winning party (red or black).

-e *int*, **--every** *int* Emit information about the state of the simulation after every *int*'ths cycle to stdout. The first information is the 0-th cycle, after the simulated world was created but before any other action was taken; it is always printed. More formally, the state of the world after cycle I<n> is printed, if I<n mod int = 0>. The default for I<int> is C<1>; the format of the output is controlled by the B<-log> flag.

-l selab|ascii, **--log** B **selab|ascii** Emit information about the state of the simulation in format **log** or **ascii**. The default is to emit no information.

The **ascii** format displays the state of the simulation as a two-dimensional maps on stdout or using a curses interface. The output may resemble the format used for

world files; this format may take keyboard input from the user to control the simulation but the details are implementation specific.

The **selab** format is a line-oriented format that contains the complete state of the world, intended for debugging. This format is defined by example below.

-h, **--help** Emit a short help message on stdout and exit. It is an error to specify any other option or file name together with **-h**.

FILES

sim does not depend on any files except those specified on the command line. The exact formats for the files *file.world* and <file><.bug> are defined elsewhere, but here are two examples:

The first example describes a world of dimension 10 by 10.

```
10
10
# # # # # # # # # #
# 9 9 . . . . 3 3 #
# 9 # . - - - - - #
# . # - - - - - - #
# . . 5 - - - - - #
# + + + + + 5 . . #
# + + + + + + # . #
# + + + + + . # 9 #
# 3 3 . . . . 9 9 #
# # # # # # # # # #
```

The second example describes a bug's behavior as a finite state machine where each line is a state, starting from zero. Everything following a semicolon up to the end of a line is a comment.

```
     sense ahead 1 3 food     ; [ 0]
     move 2 0                 ; [ 1]
     pickup 8 0               ; [ 2]
     flip 3 4 5               ; [ 3]
     turn left 0              ; [ 4]
     flip 2 6 7               ; [ 5]
     turn right 0             ; [ 6]
     move 0 3                 ; [ 7]
     sense ahead 9 11 home    ; [ 8]
     move 10 8                ; [ 9]
     drop 0                   ; [10]
     flip 3 12 13             ; [11]
     turn left 8              ; [12]
     flip 2 14 15             ; [13]
     turn right 8             ; [14]
```

```
    move 8 11                  ; [15]
```

The log file format **selab** is a compact, column-oriented format that lists for each cell of the world its contents; here is a small example, together with a header:

```
After cycle 0...
========== cell ========== ======= bug ======
        b b
        a i                    cbd
pos pos s t  red    black      oii
 x   y  e s  marks  marks  id  ltr state rest
=== === = == ====== ====== === === ===== ====
...
001 007 r 02 543__0 5_3___
002 007 r 03 5___1_ 5_3210
003 007 r 00 __3__0 __3210 010 r_4 00033 0000
004 007 r 00  4  10 54 210 005 rX3 00002 0000
005 007 r 00 __321_ 5___1_
006 007 _ 02 _4_2__ 54__1_ 012 r_4 00019 0000
007 007
008 007 _ 09 5__2__ 5__2_0
...
```

The columns denote: x and y position of the cell, color of the base if the cell belongs to a base, bits stored in cell, marks used by red and black bugs, id of bug on cell, its color, whether it carries a bit, direction, state, and resting time.

Output lines must not contain trailing white space characters. The usage of tabulator characters is not allowed. A single blank line is used to separate world dumps from each other.

EXIT CODE

Upon successful completion, **sim** exits with exit code 0, and a positive exit code otherwise.