1. Week 4: Wireworld

1.1 Wireworld

Wireworld is a cellular automaton due to Brian Silverman, formed from a 2D grid of cells, designed to simulate digital electronics. Each cell can be in one of four states, either ‘empty’, ‘electron head’, ‘electron tail’ or ‘copper’ (or ‘conductor’).

The next generation of the cells follows the rules, where n is the number of electron heads found in the 8-surrounding cells:

- empty $->$ empty
- electron head $->$ electron tail
- electron tail $->$ copper
- copper $->$ electron head if $n == 1$ or $n == 2$
- copper $->$ copper otherwise

See also:

www `https://en.wikipedia.org/wiki/Wireworld`

www `http://www.heise.ws/fourticklogic.html`

Exercise 1.1 Write a program which is run using the `argc` and `argv` parameters to `main`. The usage is as follows :

```
$ wireworld wirefile.txt
```

where `wirefile.txt` is a file specifying the initial state of the board. This file codes empty cells as ‘ ’, heads as ‘H’, tails as ‘t’ and copper as ‘c’. Display the board for 1000 generations using plain text. You may assume that the grid is always 40 cells by 40

Make sure all your code is fully ANSI compliant, and fully follows the house-style guidelines.

80%

1.2 ncurses

C has no inherent functionality to allow printing in colour etc. Therefore, a programming library know a ncurses was created in 1993 to allow terminals to interpret certain control-codes as colours and other effects.

The library itself is somewhat complex, allowing keyboard and mouse events to be captured and a whole range of simple graphics functionality. On the web page is my ‘wrapper’ for the library, along with a program demonstrating its use. This will only work in unix-style terminals.

To compile the code you’ll have to use both my code neillncurses.c and also link in the ncurses library. A typical compile might look like

```
gcc yourcode.c neillncurses.c -Wall -Wfloat-equal -Wextra -O2
    -pedantic -ansi -lncurses -lm
```

Exercise 1.2 Adapt the wireworld code so that the output is displayed using this library, with tails being red, heads being blue, copper being yellow and background being black. The main loop will update the board, display it, and repeat until a quit event occurs (e.g. a mouse click or the ESC key is pressed). ■

20%