

DESIGN

Yifan Zou

January 2022

1 Introduction

This assignment contains `universe.c`, which implements the Universe ADT, and `life.c` which contains `main()` and other functions to complete the implement of the Game of Life.

2 Code Design

2.1 `universe.c`

2.1.1 `*uv_create()`

This function creates a universe, and creates memory for all the elements inside the universe.

```
create a universe
use for loop to create memory for rows
    use for loop to create memory for cols
        create memory for each grid

set the value rows, cols, and toroidal
```

2.1.2 `uv_delete()`

This function frees the created memory.

```
free all the grid
free the universe
```

2.1.3 `uv_rows()/uv_cols()`

These two functions return the numbers of rows and cols in the universe.

```
return rows/cols
```

2.1.4 `uv_live_cell()/uv_dead_cell()`

These functions assign cells as alive or dead.

```
if rows and cols are within valid input
    make the specific grid true/false
```

2.1.5 `uv_get_cell()`

This function checks if the cell alive or dead.

```
if rows and cols are within valid input
    return the value in grid
```

2.1.6 `uv_populate()`

This function imports data from infile to assign the cells in universe survive or not.

```
scan all the data in file
    if row or col is invalid
        return false
return true when all the data are valid
```

2.1.7 `uv_census()`

This function checks the number of alive neighbors of the selected cell.

```
check if the grid exists
    if the universe is toroidal
        top = (r+rows-1)% rows
        bot = (r+rows+1)% rows
        left = (c+cols-1)% cols
        right = (c+cols+1)% cols

        list_rs = [top, r, bot]
        list_cs = [left, c, right]

        loop through all the neighbor, exclude (r,c)
            if the neighbor is alive
                num += 1
        return the number of alive neighbor

    if the universe is flat
        check all the neighbor value, exclude the value out of bound or grid itself
        num += 1
        return the number of alive neighbor
if the grid doesn't exist, error
```

2.1.8 uv_print()

This function prints out the universe.

```
for all the grid
    if the grid is alive
        print("o")
    if the grid is dead
        print(".")
```

2.2 life.c

use getopt() to receive the input from user

-i: read the first line of the file to get the numbers of rows and cols.

create two universe, u_A and u_B

-t: mark the flag toroidal as true

populate u_A, set all the grid included in the file as alive

-s: set a ncurses screen

generate for 100 times

 // Display u_A

 loop through all the grids

 if the cell is alive

 print "o"

 refresh the screen

 sleep for 50000 microsecond

 // Perform next generation

 loop through all the cells and check if the cell survives

 if survive, check number of alive neighbor

 if live_neighbor == 2 || 3

 record at u_B that the cell at (row, col) is alive in next generation

 else

 record as dead

 if dead

 if live_neighbor == 3

 alive

 else

 dead

 swap u_A and u_B

close screen

```
print out the result  
  
free the memory  
  
close infile and outfile  
  
end
```