

Intro to C Programming

Summer 2011

Lab 2: Game of Life Notes

Overview

In this lab, we will extend Lab 1, by implementing Conway's Game of Life, with fewer restrictions. In the first version, the code used static arrays and array indexing. This version will use dynamic memory allocation with `malloc()` and pointer arithmetic `*(ptr+i)` instead of `ptr[i]`.

- Ignore the stuff about the struct. It would be a waste of time. The current code does not lend itself well to this data structure.
- All of my changes were made in `lifegame.c`
- The world, nextstates, worldheight and worldwidth become ptrs and integers:

```
static int worldwidth;
static int worldheight;

/* current cell states of the world */
static int *world;

/* next generation cell states */
static int *nextstates;
```

- To make life easier, I defined four short functions that contained all of the pointer arithmetic:

```
int getWorld(int x,int y);
void setWorld(int s, int x, int y);
int getNext(int x,int y);
void setNext(int s, int x, int y);
```

- When reading from a file, you must read the first line to know the X and Y dimensions, and then malloc `world` and `nextstats` to the correction dimensions.
- Don't iterate with `i` and `j`, trying to set values with `x` and `y`:

```
for (i = 0; i < worldwidth; i++)
    for (j = 0; j < worldheight; j++) {
        setWorld(DEAD,x,y);
        setNext(DEAD,x,y);
    }
```

- `output_world()` must dynamically allocate a string that is the width of the current size of the world.

```
void output_world(void) {
    char *worldstr; // [2*WORLDWIDTH+2];
    int i, j;

    worldstr = (char *)malloc(2*worldwidth+2);

    worldstr[2*worldwidth+1] = '\\0';
    worldstr[0] = '+';
    for (i = 1; i < 2*worldwidth; i++)
        worldstr[i] = '-';
    worldstr[2*worldwidth] = '+';
    puts(worldstr);
    for (i = 0; i <= 2*worldwidth; i+=2)
        worldstr[i] = '|';
    for (i = 0; i < worldheight; i++) {
        for (j = 0; j < worldwidth; j++)
            worldstr[2*j+1] = getWorld(j,i) == ALIVE ? CHAR_ALIVE : CHAR_DEAD;
        puts(worldstr);
    }
    worldstr[0] = '+';
    for (i = 1; i < 2*worldwidth; i++)
        worldstr[i] = '-';
    worldstr[2*worldwidth] = '+';
    puts(worldstr);
    free(worldstr);
}
```