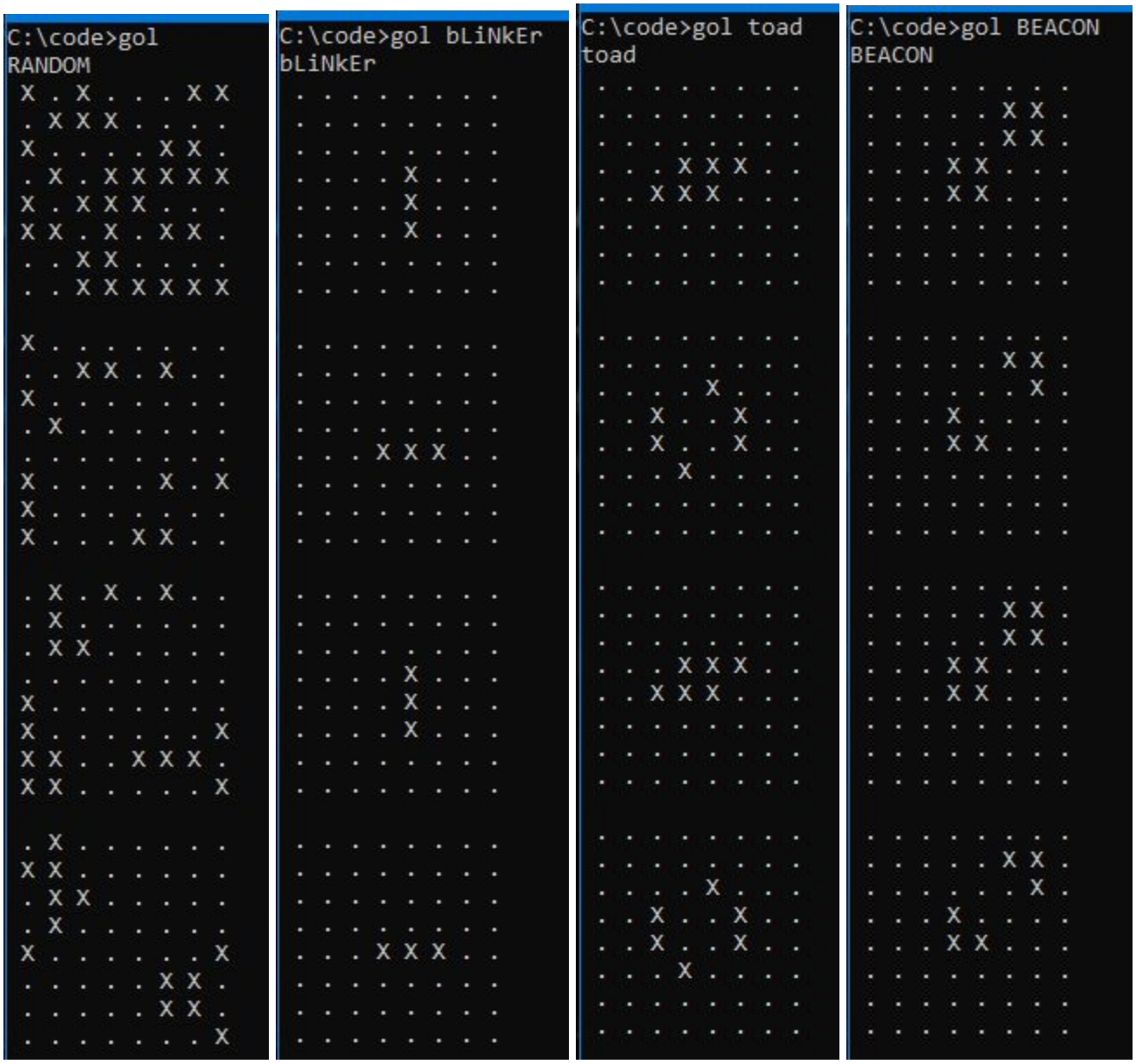
Conway's Game of Life

Read description: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Implement command line version of it on 8x8 toroidal grid on a platform of your choice.
Expected behaviour based on the command line argument is demonstrated below.



It alway print exactly 4 generations of the game steps.

Next page shows our simplistic implementation...


gol - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Release x64 puts
Toolbox
gol.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
// https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
// Any live cell with fewer than two live neighbours dies, as if by underpopulation.
// Any live cell with more than three live neighbours dies, as if by overpopulation.
// Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.
// Any live cell with two or three live neighbours lives on to the next generation.
enum { N = 8 };
typedef board_t // Square board exactly 8x8 with toroidal topology.
#define life_cell(b, x, y)
#define set_cell(b, x, y, v)
board_t BLINKER
board_t TOAD
board_t BEACON
static board_t b;
static void print() {
}
static void random() {
}
static void generation() {
}
static void init(const char* pattern) {
}
int main(int argc, const char* argv[]) {
return 0;
}

Extra bonus challenge. This is the size of our release executables. Can you make yours smaller?

```
C:\code\gol\bin>find . -name "*.exe" | xargs ls -l
-rwxrwxrwx   1 user     group        4048 May 31 18:30 ./x64/gol.exe
-rwxrwxrwx   1 user     group        3328 May 31 18:30 ./x86/gol.exe
```

(with still human readable source code)... 🙂
https://github.com/konmik/Life