/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Conways Games Of Life Simple Implementation

show changes in each generation by follow rules we explained in file Readme…

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#define Rows 10

#define Col 10

#define get\_col(c) (c%Col)

#define get\_row(c) (c/Col)

#define c\_left(c) ((get\_col(c) == 0) ? (Col-1) : -1)

#define c\_right(c) ((get\_col(c) == Col-1) ? (-Col+1) : 1)

#define c\_top(c) ((get\_row(c) == 0) ? ((Rows-1) \* Col) : -Col)

#define c\_bottom(c) ((get\_row(c) == Rows-1) ? (-(Rows-1) \* Col) : Col)

typedef struct \_cell

{

struct \_cell\* n[8];

char current\_state;

char next\_state;

} cell;

typedef struct

{

int rows;

int cols;

cell\* cells;

} world;

void evolve(cell\* c)

{

int count=0, i;

for (i=0; i<8; i++)

{

if (c->n[i]->current\_state) count++;

}

if (count == 3 || (c->current\_state && count == 2)) c->next\_state = 1;

else c->next\_state = 0;

}

void update(world\* w)

{

int nrcells = w->rows \* w->cols, i;

for (i=0; i<nrcells; i++)

{

evolve(w->cells+i);

}

for (i=0; i<nrcells; i++)

{

w->cells[i].curr\_state = w->cells[i].next\_state;

if (!(i%Col)) printf("\n");

printf("%c",w->cells[i].curr\_state ? '\*' : ' ');

}

}

world\* init\_world()

{

world\* result = (world\*)malloc(sizeof(world));

result->rows = Rows;

result->cols = Col;

result->cells = (cell\*)malloc(sizeof(cell) \* Col \* Rows);

int nrcells = result->rows \* result->cols, i;

for (i = 0; i < nrcells; i++)

{

cell\* c = result->cells + i;

c->n[0] = c+c\_left(i);

c->n[1] = c+c\_right(i);

c->n[2] = c+c\_top(i);

c->n[3] = c+c\_bottom(i);

c->n[4] = c+c\_left(i) + c\_top(i);

c->n[5] = c+c\_left(i) + c\_bottom(i);

c->n[6] = c+c\_right(i) + c\_top(i);

c->n[7] = c+c\_right(i) + c\_bottom(i);

c->current\_state = rand() % 2;

}

return result;

}

int main()

{

srand(3);

world\* w = init\_world();

while (1)

{

update(w);

getchar();

}

}

