\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW6

Author: F14076083 魏湧致 <zxc33991177@gmail.com>

Class: 資訊111 (乙班)

Description:  
這題要寫踩地雷的程式，且要用pointer的方式來控制array。在程式中有3個function，第一個是guess，用來讓使用者輸入要猜的是哪個格子，若輸入的格子已經不是未知就print ’You have clicked!’，第二是num\_of\_mine，用來計算所選的格子附近3\*3的格子有多少的地雷，如果地雷數是0則再將這8個格子放到num\_of\_mine中計算，往外擴展到地雷數不是0為止，第三是print\_out，用來print出整個網格，如果是未知或是還沒擴展到的格子print’?’，其餘print出周圍的地雷數，地雷數為0 print ’\_’，而當整個網格’?’的數量與一開始設定的地雷數相同時print ‘You win’。再主程式中設定grid和out這2個array，grid是原始的array，內容1代表該格子有地雷，out是用來print的array。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Code:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define Nmax 30

void guess(int \*m, int \*n, char out[][Nmax])

{

scanf("%d%d", m, n);

if(out[\*m][\*n]!='?')

{

printf("You have clicked!\n");

guess(m, n, out);

}

}

void num\_of\_mine(int grid[][Nmax+1], char out[][Nmax], int m, int n, int size)

{

int \*p, c=0, num=0, i=-1, j=-1;

if(m<0 || m>size-1 || n<0 || n>size-1 || out[m][n]!='?')

return;

for(p=&grid[m][n]-Nmax-2; p<=&grid[m][n]+Nmax+2; p++)

{

if(c==3){

p=p+Nmax-2; c=0;

}

if(m!=0 || m!=size-1 || n!=0 || n!=size-1)

;

else{

c++; continue;

}

if(\*p==1)

num++;

c++;

}

out[m][n]=num+'0'; c=0;

if(num==0)

{

for(i=-1; i<=1; i++){

for(j=-1; j<=1; j++){

if(i==0 && j==0)

continue;

num\_of\_mine(grid, out, m+i, n+j, size);

}

}

}

}

void print\_out(char out[][Nmax], int size, int m, int n, int mine)

{

char \*c;

int count=0, correct=0;

for(c=&out[0][0]; c<=&out[size-1][size-1]; c++)

{

if(\*c=='0')

printf("\_ ");

else

printf("%c ",\*c);

if(\*c=='?')

correct++;

count++;

if(count==size)

{

c=c+Nmax-size;

count=0;

printf("\n");

}

}

printf("\n");

if(correct==mine){

printf("You win"); exit(1);

}

}

int main(int argc, char \*argv[])

{

srand(time(NULL)); rand();

int size=atoi(argv[1]), mine=atoi(argv[2]);

int grid[Nmax+1][Nmax+1], \*p;

char out[Nmax][Nmax], \*c;

int m, n, num;

if(mine>size\*size)

{

printf("Error!");

return 0;

}

for(p=&grid[0][0]; p<&grid[size][size]; p++)

\*p=0;

for(c=&out[0][0]; c<&out[size][size]; c++)

\*c='?';

for (int i=0; i<mine; i++)

{

p=&grid[0][0];

int x=(int) ((double)(rand()) / (RAND\_MAX+1.0) \* (size\*size + 0.0));

int u=x%size;

if(\*(p+(x/(size))\*(Nmax+1)+u) == 1)

i--;

\*(p+(x/(size))\*(Nmax+1)+u) = 1;

}

print\_out(out, size, m, n, mine);

while(1)

{

guess(&m, &n, out);

system("clear");

if(grid[m][n]==1){

printf("You lose."); break;

}

num\_of\_mine(grid, out, m, n, size);

print\_out(out, size, m, n, mine);

}

return 0;

}

Compilation:

gcc -o mine mine.c

Execution:

./mine 4 2

Output:

? ? ? ?

? ? ? ?

? ? ? ?

? ? ? ?

0 0

\_ 1 ? ?

\_ 1 2 2

\_ \_ \_ \_

\_ \_ \_ \_

You win