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

Report: HW6

Author: F74086381 蘇恩質 <grgy078033@gmail.com>

Class: 112資訊系 (乙班)

Description:

用#define Nmax 30來定義Nmax = 30，設一迴圈使N > Nmax時跳出warning並讓玩家重新輸入一N值。

使用system(“clear”)讓玩家開始遊戲前先清空螢幕。

使用<time.h>函式庫並用srand(time(NULL))讓地雷位置能夠更隨機，並用rand()%的方式決定地雷隨機的位置。

以 \*a1作為二維陣列a2的指標，用以產生一個N\*N的grid，設安全且尚未被猜或展開的座標點為10、地雷為9，用以隨機放入M個炸彈

另設一個\*a3[N]當作a2[N][0]的位置、\*\*map透過迴圈使得\*\*map作為a2的指標的指標。

另寫一recursive函式mine，檢查map[i][j]周遭是否有地雷，若有地雷，則return map[i][j] = 周遭地雷數；若無地雷，則recursive map[i][j]周遭8格並使map[i][j] = 0。

地圖若有一個問號(10 or 9)，則x++，當x = M(預設地雷數)時，跳出勝利訊息並跳出遊戲迴圈結束遊戲。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define Nmax 30

int mine(int \*\*map, int i, int j, int N);

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

{

system("clear");

srand(time(NULL));

int bomb = 0, i, j, m, n, x, y;

int game = 1;

int N = atoi(argv[1]);

int M = atoi(argv[2]);

while(N > Nmax || N <= 0)

{

printf("your N should lower or equal 30 and bigger than 0\n");

printf("input a new N:\n");

scanf("%d", &N);

printf("\n");

}

int \*a1, a2[N][N], \*a3[N], \*\*map;

a1 = (int \*)a2;

map = (int \*\*)a3;

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

map[i] = (int \*)a2[i];

for(a1 = &a2[0][0]; bomb < M; a1++)

{

i = rand()%N;

j = rand()%N;

if(\*(\*(a2+i)+j) == 9)

continue;

\*(\*(a2+i)+j) = 9;

bomb++;

}

for(a1 = &a2[0][0]; a1 <= &a2[N - 1][N - 1]; a1++)

{

if(\*a1 == 9)

continue;

else

\*a1 = 10;

}

for(y = N + 2, a1 = &a2[0][0]; a1 <= &a2[N - 1][N - 1]; a1++, y++)

{

printf("? ");

if(y != N + 2 && y % N == 1)

printf("\n");

}

while(game)

{

int k = 0;

printf("\nguess:");

scanf("%d%d", &i, &j);

printf("\n");

if(i < 0 || i > N || j < 0 || i > N)

{

printf("wrong position.\nchoose another.\n");

continue;

}

if(\*(\*(a2+i)+j) != 9 && \*(\*(a2+i)+j) != 10)

{

printf("you have guessed this position.\nchoose another.\n");

continue;

}

if(\*(\*(a2+i)+j) == 9)

{

printf("You Are Dead!!\n\n");

break;

}

mine(map, i, j, N);

x = 0;

for(y = N + 2, a1 = &a2[0][0]; a1 <= &a2[N - 1][N - 1]; a1++, y++)

{

if(\*a1 != 10 && \*a1 != 9)

{

if(\*a1 == 0)

printf("\_ ");

else

printf("%d ", \*a1);

}

else

{

x++;

printf("? ");

}

if(y != N + 2 && y % N == 1)

printf("\n");

}

if(x \* 9 == M \* 9)

{

printf("\nYou Win!!\n\n");

break;

}

}

return 0;

}

int mine(int \*\*map, int i, int j, int N)

{

int m, n, k = 0, p, q;

if(map[i][j] == 9 || map[i][j] == 10)

{

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

{

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

{

if(m == i && n == j)

continue;

if(m < 0 || m >= N)

continue;

if(n < 0 || n >= N)

continue;

if(map[m][n] == 9)

k++;

}

if(m == i + 1)

{

map[i][j] = k;

if(map[i][j] == 0)

{

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

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

{

if(p == i && q == j)

continue;

if(p < 0 || p >= N)

continue;

if(q < 0 || q >= N)

continue;

mine(map, p, q, N);

}

}

else

return map[i][j];

}

}

}

else

return map[i][j];

}

Compilation:

gcc -o mine mine.c

Execution:

./mine 3 1

Output:

? ? ?

? ? ?

? ? ?

guess:0 0

\_ 1 ?

\_ 1 ?

\_ 1 ?

guess:0 2

\_ 1 1

\_ 1 ?

\_ 1 ?

guess:2 2

\_ 1 1

\_ 1 ?

\_ 1 1

You Win!!