代码:

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <cmath>

#include <cstdlib>

#include <ctime>

using namespace std;

int inline rand(int Min, int Max)

{

return rand() % (Max - Min) + Min;

}

int inline rand(int Max)

{

return rand() % Max;

}

enum BlockType

{

Space = 0,

Block = -1,

};

void Restart(BlockType\* m, int w,int h,int a,int b) {

for (int y = 0; y < h ; y++) {

for (int x = 0; x < w; x++)

{

if (m[y \* w + x] == a)

m[y \* w + x] = (BlockType)b;

}

}

}

void Connect(BlockType \* m,int w,int h,int x,int y) {

int dx = 1,dy = 1;

int nx = 1, ny = 1;

for (;;) {

switch (rand(4)) {

case 0:

nx = x + 2;

ny = y;

dx = x + 1;

dy = y;

break;

case 1:

nx = x - 2;

ny = y;

dx = x - 1;

dy = y;

break;

case 2:

nx = x;

ny = y + 2;

dx = x;

dy = y + 1;

break;

case 3:

nx = x;

ny = y - 2;

dx = x;

dy = y - 1;

break;

}

if (nx == 0 || ny == 0||dy==0||dx==0||nx == w-1 || ny == h-1 || dy == h-1 || dx == w-1)continue;

else break;

}

if ((m[ny \* w + nx] != m[y \* w + x])&&dx > 0 && dy > 0 && dx < w - 1 && dy < h - 1 && (dx + dy) % 2 == 1) {

int a= m[y \* w + x], b= m[ny \* w + nx];

if (m[ny \* w + nx] >= m[y \* w + x]) {

Restart(m, w, h, b, a);

}

else {

Restart(m, w, h, a, b);

}

m[dy \* w + dx] = BlockType::Space;

}

}

void color(int color)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color);

}

bool All(BlockType\* m, int w, int h) {

for (int y = 0; y < h; ++y)

for (int x = 0; x < w; ++x)

{

if (m[y \* w + x] != BlockType::Space && m[y \* w + x] != BlockType::Block) {

if(m[y \* w + x] >1)return 0;

}

}

return 1;

}

void InitMaze(BlockType\* m, int w, int h)

{

for (int y = 0; y < h; ++y)

for (int x = 0; x < w; ++x)

m[y \* w + x] = BlockType::Block;

for (int y = 1; y < h; y += 2)

for (int x = 1; x < w; x += 2)

m[y \* w + x] = BlockType::Space;

int id = 0;

for (int y = 0; y < h; ++y)

for (int x = 0; x < w; ++x)

{

if (m[y \* w + x] == BlockType::Space)

m[y \* w + x] = (BlockType)++id;

}

/\*

\*/

for (;;) {

int x = rand(1,w-1);

int y = rand(1,h-1);

Connect(m, w, h, x, y);

if (All(m, w, h) == 1)break;

}

for (int i=0;i<2;) {

if (rand(2) == 1) {

int x = rand(1, w - 1);

if (i < 2&&m[0 \* w + x] == BlockType::Block&& m[1 \* w + x] == BlockType::Space) {

m[0 \* w + x] = BlockType::Space;

i++;

}

if (i < 2&&m[(h-1) \* w + x] == BlockType::Block&& m[(h - 2) \* w + x] == BlockType::Space) {

m[(h - 1) \* w + x] = BlockType::Space;

i++;

}

}else {

int y = rand(1, h - 1);

if (i < 2&&m[y\* w + 0] == BlockType::Block && m[y \* w + 1] == BlockType::Space) {

m[y \* w + 0] =BlockType::Space;

i++;

}

if (i < 2&&m[y \* w + w-1] == BlockType::Block && m[y \* w + w - 2] == BlockType::Space){

m[y \* w + w - 1] = BlockType::Space;

i++;

}

}

}

}

void PrintMaze(BlockType\* m, int w, int h)

{

for (int y = 0; y < h; ++y)

{

for (int x = 0; x < w; ++x)

{

switch (m[y \* w + x])

{

case BlockType::Space:

color(2);

cout << " ";

break;

case BlockType::Block:

if (x == 0 || y == 0

|| x == w - 1 || y == h - 1)

color(6);

else if (x % 2 == 0 && y % 2 == 0)

color(4);

else

color(3);

cout << "ǽ";

break;

default:

color(15);/\*setw(2) << hex <<m[y \* w + x] \*/

cout <<" ";

break;

}

}

cout << endl;

}

color(15);

}

int main()

{

srand(time(0));

int w = 39;

int h = 21;

BlockType\* maze = new BlockType[w \* h];

InitMaze(maze, w, h);

PrintMaze(maze, w, h);

delete[]maze;

}

结果：



