（1）configuration.h

//configuration.h

#ifndef \_\_CONFIGURATION\_H

#define \_\_CONFIGURATION\_H

#include <iostream>

#include <string>

#include <cmath>

#include <iomanip>

#include <ctime>

#include <vector>

#include <algorithm>

#include <stdio.h>

#include <cstdlib>

#include <map>

#include <fstream>

#include <sstream>

#include <ctime>

using namespace std;

class Configuration

{

private:

vector<vector<char> > board;

vector<vector<int> > count;

public:

void iniBoard(int, int);

void iniFromFile(string&);

void iniFromKeyboard(int, int);

void boardUpdate();

void boardPrint()const;

};

#endif

（2）configuration.cpp

//configuration.cpp

#include "configuration.h"

#include <iostream>

#include <string>

#include <cmath>

#include <iomanip>

#include <ctime>

#include <vector>

#include <algorithm>

#include <stdio.h>

#include <cstdlib>

#include <map>

#include <fstream>

#include <sstream>

#include <ctime>

using namespace std;

vector<vector<int> > count1;

void Configuration::iniBoard(int x, int y)

{

vector<vector<char> > board1(x + 2, vector<char>(y + 2, ' '));

vector<vector<int> > temp(x + 2, vector<int>(y + 2, 0));

count = temp;

board = board1;

count1 = count;

}

void Configuration::iniFromKeyboard(int x, int y)

{

board[x][y] = '\*';

}

void Configuration::iniFromFile(string &filename)

{

ifstream ifs;

ifs.open(filename, ios::in);

char ct[5];

while( ifs.getline(ct, 5) )

{

int x = int(ct[0]) - 48;

int y = int(ct[2]) - 48;

board[x][y] = '\*';

}

ifs.close();

}

void Configuration::boardUpdate()

{

for(int i = 1; i <= board.size() - 2; i ++)

{

for(int j = 1; j <= board[0].size() - 2; j ++)

{

if(board[i][j] == '\*')

{

count[i - 1][j - 1] ++;

count[i - 1][j] ++;

count[i - 1][j + 1] ++;

count[i][j - 1] ++;

count[i][j + 1] ++;

count[i + 1][j - 1] ++;

count[i + 1][j] ++;

count[i + 1][j + 1] ++;

}

}

}

for(int i = 1; i <= board.size() - 2; i ++)

{

for(int j = 1; j <= board[0].size() - 2; j ++)

{

if(board[i][j] == '\*')

{

if(count[i][j] != 2 && count[i][j] != 3)

board[i][j] = ' ';

}

else

{

if(count[i][j] == 3)

board[i][j] = '\*';

}

}

}

count = count1;

}

void Configuration::boardPrint() const

{

for(int i = 1; i <= board.size() - 2; i ++)

{

for(int j = 1; j <= board[0].size() - 2; j ++)

{

cout << board[i][j] << ' ';

}

cout << endl;

}

}

（3）main.cpp

//main.cpp

#include "configuration.h"

#include <iostream>

#include <string>

#include <cmath>

#include <iomanip>

#include <ctime>

#include <vector>

#include <algorithm>

#include <stdio.h>

#include <cstdlib>

#include <map>

#include <fstream>

#include <sstream>

#include <ctime>

using namespace std;

int main()

{

bool flag; string filename;

int x, y, number, x1, y1, turns;

Configuration c1;

cout << "This is a lifegame, so you should make sure you've already understand its rules.\n";

cout << "Please input 0 or 1. The number 1 represents you will input data from keyboard, and number 2 from file.\n";

cin >> flag;

cout << "Please input the number of rows and columns of the board.\n";

cin >> x >> y;

c1.iniBoard(x, y);

if(flag == 1)

{

cout << "Then please input the number of living cells.\n";

cin >> number;

cout << "and input the location of each cell.\n";

while(number --)

{

cin >> x1 >> y1;

c1.iniFromKeyboard(x1, y1);

}

}

else

{

cout << "Please input the name of data file.\n";

cin >> filename;

c1.iniFromFile(filename);

}

cout << "How many turns do you want to see the changes?\n";

cin >> turns;

for(int i = 1; i <= turns; i ++)

{

cout << "turn " << i << ":\n";

c1.boardUpdate();

c1.boardPrint();

}

return 0;

}