第一周工作报告

专业班级：软件工程1班

学号：202430550672

姓名：黄泓源

实验内容：将people类、mananger类、game类基本实现。

代码：

#ifndef GAME\_H

#define GAME\_H

#include <iostream>

enum Direction{UP,DOWN,LEFT,RIGHT};

class people

{

public:

people() :x(0), y(0) {}

void DisplayMessage(const people&)const;

void SetPeopleBeginPos(const int&,const int&);

void Movepeople(const Direction& heading);

int& getx();

int& gety();

int returnx()const;

int returny()const;

private:

int x;

int y;

};

class game

{

public:

~game();

void DisplayRelativeMove( people\*&, people\*&)const;

bool IfYes( int, people\*&);

void setrun(const int&);

int getrun()const;

people& getp1();

people& getp2();

people\*& getp1array();

people\*& getp2array();

people\*& getp2relative();

void setp2relative(people\*&, people\*&, people\*&,const int& run);

void setzihzhen();

void getanswer( people \*&)const;

bool ifpass( people\*&, const int&)const;

bool ifpass1(people\*&, const int&)const;

void gettip( int &, people\*&)const;

private:

people p1, p2;

int run;

people\* p1array=nullptr;

people\* p2array = nullptr;

people\* p2relative = nullptr;

};

#endif // !GAME\_H

#ifndef MANANGER\_H

#define MANANGER\_H

#include <iostream>

#include "player.h"

#include"game.h"

class Mananger

{

public:

void ChangeChoose();

void ShowMenu()const;

void ShowRule()const;

void Exitgame();

void BeginGame();

void ChooseGame();

void RunChoose();

int getlevel()const;

void CheckMessage(player&);

void GameShowmess(game& g1);

void GameShowmess1(game& g1);

player& getplayer();

void creatgame1();

void creatgame2();

void creatgame3();

void creatgame4();

void creatgame5();

void randomcreatgame();

void PlayerGreatgame();

void challengegame();

void randomcreatgame(int number1);

void PlayerGameShowmess(game& g1);

void noendchallenge();

private:

int choose;

int level;

player p1;

};

#endif // !MANANGER\_H

#include "game.h"

#include <iostream>

#include <iomanip>

using namespace std;

void people::DisplayMessage(const people& p1)const

{

cout <<"("<< p1.returnx() << "," << p1.returny()<<")" << " ";

}

void people::SetPeopleBeginPos(const int& x1, const int& y1)

{

this->x = x1;

this->y = y1;

}

void people::Movepeople(const Direction& heading)

{

switch (heading)

{

case UP:

this->y--;

break;

case DOWN:

this->y++;

break;

case LEFT:

this->x--;

break;

case RIGHT:

this->x++;

break;

}

}

int& people::getx()

{

return x;

}

int& people::gety()

{

return y;

}

int people::returnx() const

{

return x;

}

int people::returny()const

{

return y;

}

game::~game()

{

delete[]p1array;

delete[]p2array;

delete[]p2relative;

p1array = nullptr;

p2array = nullptr;

p2relative = nullptr;

}

void game::setrun(const int& num1)

{

this->run = num1;

}

void game::setzihzhen()

{

p1array = new people[run];

p2array = new people[run];

p2relative = new people[run];

}

void game::setp2relative(people\*& p1, people\*&p2, people\*&p3,const int& run)

{

p1[0].getx() = p2->returnx();

p1[0].gety() = p2->returny();

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

{

p1[i].getx() = p1[i - 1].getx() + p2[i].returnx()-p2[i-1].returnx()- p3[i].returnx() + p3[i - 1].returnx();

p1[i].gety() = p1[i - 1].gety() + p2[i].returny() - p2[i - 1].returny() - p3[i].returny() + p3[i - 1].returny();

}

}

void game::DisplayRelativeMove( people\*& p1array, people\*&p2relative)const

{

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

{

if (i == 0)

{

cout << "你与别人的起点：" << endl;

p1array[i].DisplayMessage(p1array[i]);

p2relative[i].DisplayMessage(p2relative[i]);

cout << endl;

cout << "步数 你与别人的坐标" << endl;

}

else

{

cout <<setw(3)<< i << setw(8);

p1array[i].DisplayMessage(p1array[i]);

p2relative[i].DisplayMessage(p2relative[i]);

cout << endl;

}

}

}

bool game::ifpass(people\*& p1, const int& num)const

{

for (int i = this->getrun() - 1; i > num; i--)

{

if ((p1[i].returnx() == p1[num].returnx()) && p1[i].returny() == p1[num].returny())

{

return false;

}

}

return true;

}

int game::getrun() const

{

return run;

}

people& game::getp1()

{

return this->p1;

}

people& game::getp2()

{

return this->p2;

}

people\*&game::getp1array()

{

return this->p1array;

}

people\*& game::getp2array()

{

return this->p2array;

}

people\*& game::getp2relative()

{

return this->p2relative;

}

void Mananger::ChooseGame()

{

cout << "请输入您要挑战的关卡:" << endl;

int num;

cin >> num;

this->level = num;

switch (getlevel())

{

case 1:

{

creatgame1();

break;

}

case 2:

{

creatgame2();

break;

}

case 3:

{

creatgame3();

break;

}

case 4:

{

creatgame4();

break;

}

case 5:

{

creatgame5();

break;

}

case 6:

{

randomcreatgame();

break;

}

default:

cout << "当前关卡不存在，请重新输入!"<<endl;

ChooseGame();

}

}

#include"mananger.h"

#include"game.h"

#include "player.h"

#include <iostream>

#include <time.h>

#include <random>

#include <ctime>

using namespace std;

bool again()

{

int num1 = 0;

cin >> num1;

if (num1 == 1) { return true; }

else if (num1 == 0) { return false; }

else

{

cout << "输入不符合要求，请重新输入:" << endl;

again();

}

}

void Mananger::RunChoose()

{

ShowMenu();

ChangeChoose();

switch (choose)

{

case 1:

{

system("cls");

BeginGame();

break;

}

case 2:

{

system("cls");

ShowRule();

break;

}

case 0:

{

system("cls");

Exitgame();

break;

}

case 3:

{

system("cls");

CheckMessage(p1);

break;

}

case 4:

{

system("cls");

PlayerGreatgame();

break;

}

case 5:

{

system("cls");

challengegame();

break;

}

case 6:

{

system("cls");

noendchallenge();

break;

}

default:

cout << "输入的选择不符合要求，请重新输入：" << endl;

}

}

void Mananger::ChangeChoose()

{

int choose1;

cin >> choose1;

this->choose = choose1;

}

void Mananger::ShowMenu()const

{

cout << " 请输入你的选择：" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*1开始游戏\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*2了解游戏规则\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*3展示用户消息\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*4创造模式\*\*\*\*\*\*\*\*\*\*"<<endl;

cout << "\*\*\*\*\*\*\*\*5挑战模式\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*6无尽模式\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*0退出游戏\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

};

void Mananger::ShowRule()const

{

cout << "您将会获得一条通往某一目的地的航行路线，同时，以你为参照物，您能获得另一人的相对路径，请推理出另一人的实际路径。" << endl;

}

void Mananger::Exitgame()

{

cout << "游戏已退出" << endl;

exit(0);

}

void Mananger::BeginGame()

{

ChooseGame();

}

int Mananger::getlevel()const

{

return level;

}

player& Mananger::getplayer()

{

return this->p1;

}