#include <conio.h>

#include <iostream>

#include <ctime>

#define размерX 16

#define размерY 16

#define РазмерZ 6

#define Монеты 8

#define Имя 2

//могу поспорить при долгой игре эта х-ня вызовет эпилепсию)) но зато она красивая XD

//2- отделить статику от динкамики (разбиение на типы)-гениально : разбите на статику и динамику.

//статика (лесницы,стены,пол)-т.е то что задает движение.(1н массив)

//динамика (монетки,мобы,персонаж)- т.е то что постоянно меняеца... (по масиву на каждый тип..содержаший положение данного обьекта)

//+разбиваем обьекты по связям кпримеру монетка может лежать над пропастью... как и..

//1- большая карта + спрайты.

//четко делим 3 функции вывод карты... карта...положение обьектов на карте...

using namespace std;

//Создаем структуру для параметров движения.(необязательно но удобно)

struct devil{

int X[Имя+1];

int Y[Имя+1];

int Z[Имя+1];

int pover[Имя];

};

struct MoveIm {

int Z;

int X;

int Y;

};

struct golden{

int X[Монеты+1];

int Y[Монеты+1];

int Z[Монеты+1];

};

//для меню.

struct lolca {

int XP;

int op;

int S;

bool tip;

bool lift;

};

struct Tipiz{

int tippolya[РазмерZ+1][размерX+1][размерY+1];// Маассив типа карты (на 1цу больше фактически из вежливости=)

};

//void devilmove(devil&,MoveIm& , MoveIm& , Tipiz&);

void generick(MoveIm&, Tipiz&);

void postpole(MoveIm&, Tipiz&, MoveIm&, golden&);//исправить

void menyushka(lolca&);

void movein(lolca&, MoveIm&, Tipiz&, char&, MoveIm& , golden& );

void niz(MoveIm&, Tipiz&, MoveIm&, golden&);

int main()

{

srand(time(0));

setlocale(LC\_ALL, "Russian"); //Для вывода Русских букв.

MoveIm ime, final11, pole;// Структуры положений.

Tipiz tip;

lolca lol; //х-ня

golden gold;

devil demon;

char hod;

pole = { РазмерZ , размерX, размерY, }; // Размер поля.

ime = { 1 , 10, 10 }; // Начальное положение игрока.

generick(pole, tip); // Генерация карты

final11.X = (rand() % (pole.X - 1)) + 2;

final11.Y = (rand() % (pole.Y - 1)) + 2;

final11.Z = РазмерZ; //Этаж (всегда верхний)

tip.tippolya[final11.Z][final11.X][final11.Y] = 5;

//Генерация 1х монет

for (int i(0); i < Монеты ; i++){

gold.X[i] = (rand() % (pole.X - 2)) + 2;

gold.Y[i]= (rand() % (pole.Y - 2)) + 2;

gold.Z[i]= ime.Z;//Этаж

}

//генерация 1х врагов

for (int i(0); i < Монеты; i++){

gold.X[i] = (rand() % (pole.X - 2)) + 2;

gold.Y[i] = (rand() % (pole.Y - 2)) + 2;

gold.Z[i] = ime.Z;//Этаж

}

lol.XP = 100;// Начальный опыт.

lol.op = 10; // Опыт для апа х-ки.

lol.S = 1; // Скорость.

lol.lift = true;//тип движения

bool Vin = false;

while (Vin == false)

{

//Меню.

cout << "\n Ваши координаты " << ime.X << "." << ime.Y << "\t Этаж:\_" << ime.Z << "\_";

cout << "\n Выберите направление движения" << "\n W - вверх." << "\t S - низ." << "\n D - право." << "\t A - лево." << "\n Y - Характеристики ." << "\t E - Посмотреть вниз .";

cout << "\n Координаты победы " << final11.X << "." << final11.Y << " Этаж:"<< final11.Z << " \_\_\_\_Счет:" << lol.XP << "\_\_\_" << endl; //Для теста.

postpole(pole, tip, ime, gold);// <----- Создание кривой но всеже карты мира =)

cout << "Введите команду\_\_";

hod = \_getche();

movein(lol, ime, tip, hod, pole, gold );//движение

//devilmove(demon, pole, ime, tip);//движение демонов

// финиш.

if (ime.Z == final11.Z && ime.X == final11.X && ime.Y == final11.Y)

{

Vin = true;

for (int i = 0; i < 100; i++) {

cout << " \_\_\_\_END\_\_\_\_";

}

}

}

\_getch();

return 0;

}

//---------------------Построение поля-------------------------//

//Необьяснимый косяк с передачей поля(был решен также неожиданно как и появился..но кто знает...=) ))))

void postpole(MoveIm& pole, Tipiz& tip, MoveIm& ime, golden& gold) {

//цвет уровня(карты (1-цвет окна 2-цвет букв))

if (ime.Z == 0) { system("color 40 "); };

if (ime.Z == 1) { system("color 20 "); };

if (ime.Z == 2) { system("color 80 "); };

if (ime.Z == 3) { system("color 04 "); };

//построение карты.

for (int i = 0; i <= размерY ; i++)

{

for (int j = 0; j <= размерX ; j++)

{

if (ime.X == i && ime.Y == j) { cout << "[V]"; }//акуратно врисовываем игрока

else

{

for (int k(0); k < Монеты; k++){//похоже на костыль но им не является =)

if (gold.Z[k] == ime.Z && gold.X[k] == i && gold.Y[k] == j){ cout << "[0]"; j++; }

}

switch (tip.tippolya[ime.Z][i][j]){

case 1 :cout << "[\_]"; break;

case 2 :cout << "[X]"; break;

case 3 :/\*cout << "[0]";\*/ break;

case 4 :/\*cout << "[V]";\*/ break;

case 5 :cout << "[F]"; break;

case 6 :cout << "[/]"; break;

case 7 :cout << " "; break;

default: cout << "Ошибка";

}

}

}

cout << "\n";

}

}

//-------------------меню--------------------------------//

void menyushka(lolca& lol)

{

lol.tip = true;

while (lol.tip == true) {

char menyu = '3';

cout << "Настройка параметров персонажа :" << endl;

cout << "Скорость:" << lol.S << endl;

cout << "Опыт:" << lol.XP << endl;

cout << "Направление движения :"; if (lol.lift == true){ cout << "\_Вверх\_" << endl; } else { cout << "\_Вниз\_" << endl; };

cout << "Для увеличения скорости требуется " << lol.op << "опыта." << endl;

cout << "Выберите действие:" << "\n 1 - увеличить скорость." << "\n 2 - уменьшить скорость." << "\n 3 - Сменить направление движения по лестнице." << "\n Y - вернутся к игре ." << endl;

menyu = \_getche();

switch (menyu)

{

case '1': if (lol.XP > lol.op) { lol.S++; lol.XP -= lol.op; lol.op = lol.op \* 2; system("cls"); }

else { system("cls"); } break;

case '2': lol.S--; system("cls"); break;

case '3': if (lol.lift == true){ lol.lift = false; } else{ lol.lift = true; }; system("cls"); break;

case 'y': system("cls"); lol.tip = false; break;

}

}

}

//---------------------Генерация карты----------------------------//

void generick(MoveIm& pole, Tipiz& tip)

{

//желательно отделить карту от игрока(разделение карты на статичную и динамичную...)

//использовать модели генерации как спрайты (прибавляя координаты массива спрайта с обозначениями координатами обшей карты)

MoveIm lestnica;

int HHH;

for (int i = 0; i <= pole.Z ; i++)

{

lestnica.X = (rand() % (pole.X - 2))+1 ;

lestnica.Y = (rand() % (pole.Y - 2))+1 ;

for (int j = 0; j <= pole.X ; j++)

{

for (int k = 0; k <= pole.Y ; k++)

{

HHH = (rand() % (pole.Y - 2)) + 1;//рандомизация

tip.tippolya[i][j][k] = 1 ;

if (lestnica.X == j && lestnica.Y == k && i <pole.Z){ tip.tippolya[i][j][k] = 6; };//Строим лесницы в здании дабы передвигатся вверх вниз.

if (j == 0 || k == 0 || j == (pole.X ) || k == (pole.Y )) { tip.tippolya[i][j][k] = 2; };//Строим стены на границе карты(дабы невыходить за ее границы

//строим местападения=)

if (j > 0 && k > 0 && j < (pole.X) && k < (pole.Y) && i > 1 && HHH == k){ tip.tippolya[i][j][k] = 7; };

//Нужно еше коечто продумать

//1 - можно ити.

//2 - стена.

//3 - монетка

//4 - персонаж

//5- финиш

//6 - лесница

//7 - дыра(в полу)

//8-(моб)

}

}

}

}

//-------------------------Движение --------------------------------//

//Исправить тип локации (проблема-затирание активных обьектов..решение дублирование памяти об типе локации)

void movein(lolca& lol, MoveIm& ime, Tipiz& tip, char& hod, MoveIm& pole, golden& gold) {

switch (hod)

{

case 'ц':

case 'W':

case 'w': if (tip.tippolya[ime.Z][ime.X - lol.S][ime.Y] != 2) { ime.X -= lol.S; system("cls"); }

else { system("cls"); cout << "Вы уперлись в стену..увы предется искать другой путь..."; }

break;

case 'ы':

case 'S':

case 's': if (tip.tippolya[ime.Z][ime.X + lol.S][ime.Y] != 2) { ime.X += lol.S; system("cls"); }

else { system("cls"); cout << "Вы уперлись в стену..увы предется искать другой путь..."; }

break;

case 'ф':

case 'A':

case 'a': if (tip.tippolya[ime.Z][ime.X][ime.Y - lol.S] != 2) { ime.Y -= lol.S; system("cls"); }

else { system("cls"); cout << "Вы уперлись в стену..увы предется искать другой путь..."; }

break;

case 'D':

case 'в':

case 'd': if (tip.tippolya[ime.Z][ime.X][ime.Y + lol.S] != 2) { ime.Y += lol.S; system("cls"); }

else { system("cls"); cout << "Вы уперлись в стену..увы предется искать другой путь..."; };

break;

case 'y': system("cls"); menyushka(lol); break; // меню.

case 'e': system("cls"); niz( pole , tip, ime, gold); break; // виденье

default: cout << "Неверная команда :" << hod;

}

//есле мы на леснице

if (tip.tippolya[ime.Z][ime.X][ime.Y] == 6){

if (lol.lift == true) { if (ime.Z < РазмерZ ){ ime.Z += 1; }; }

else

{

if (ime.Z > 0){ ime.Z -= 1; }

}

};

//Генерация монетки(для возможности брать монетку налету)

//множитель монет =)

for (int i(0); i < Монеты; i++){

while (ime.Z == gold.Z[i] && ime.X == gold.X[i] && ime.Y == gold.Y[i]) {

gold.X[i] = ((rand() % (pole.X - 1)) + 1);

gold.Y[i] = ((rand() % (pole.Y - 1)) + 1);

gold.Z[i] = ((rand() % (pole.Z - 1)) + 1);//Этаж

lol.XP += 100;

}

};

//падение.

while (tip.tippolya[ime.Z][ime.X][ime.Y] == 7){

ime.Z -= 1;

};

}

void niz(MoveIm& pole, Tipiz& tip, MoveIm& ime, golden& gold){

if ((tip.tippolya[ime.Z][ime.X - 1][ime.Y] == 7) || (tip.tippolya[ime.Z][ime.X + 1][ime.Y] == 7) || (tip.tippolya[ime.Z][ime.X][ime.Y - 1] == 7) || (tip.tippolya[ime.Z][ime.X][ime.Y + 1] == 7))

{

bool ziryu = true;

char k;

while (ziryu == true)

{

cout << "Для выхода нажмите \_e\_" << endl;

//цвет уровня(карты (1-цвет окна 2-цвет букв))

system("color F0 ");

//построение карты.

for (int i = 0; i <= размерY; i++)

{

for (int j = 0; j <= размерX; j++)

{

if (ime.X == i && ime.Y == j) { cout << "[@]"; }//акуратно врисовываем игрока

else

{

for (int k(0); k < Монеты; k++){//похоже на костыль но им не является =)

if (gold.Z[k] == ime.Z-1 && gold.X[k] == i && gold.Y[k] == j){ cout << "[0]"; j++; }

}

switch (tip.tippolya[ime.Z - 1][i][j]){

case 1:cout << "[\_]"; break;

case 2:cout << "[X]"; break;

case 3:/\*cout << "[0]";\*/ break;

case 4:/\*cout << "[V]";\*/ break;

case 5:cout << "[F]"; break;

case 6:cout << "[/]"; break;

case 7:cout << " "; break;

default: cout << "Ошибка";

}

}

}

cout << "\n";

}

k = \_getche();

switch (k)

{

case 'E':

case 'e': ziryu = false; break;

default: break;

}

system("cls");

};

};

};

/\*void devilmove(devil& demon, MoveIm& pole, MoveIm& ime, Tipiz& tip){

int y1;

int y2;

for (int i(0); i < Имя ; i++)

{

//есле демон над игроком

if (ime.Z < demon.Z[i]){

if (demon.X[i] > ime.X && tip.tippolya[demon.Z[i]][demon.X[i]-1 ][demon.Y[i]] != 2){ demon.X[i]--; }

else{

if (demon.Y[i] > ime.Y && tip.tippolya[demon.Z[i]][demon.X[i]][demon.Y[i]- 1] != 2){ demon.Y[i]--; }

else {

if (demon.X[i] < ime.X && tip.tippolya[demon.Z[i]][demon.X[i] + 1][demon.Y[i]] != 2){ demon.X[i]++; }

else{

if (demon.Y[i] < ime.Y && tip.tippolya[demon.Z[i]][demon.X[i]][demon.Y[i] + 1] != 2){ demon.Y[i]++; }

}

}

}

}

//если демон под игроком

else

{

for (int j(0); j <; j++){

for (int k(0); k <; k++){

if (tip.tippolya[demon.Z[i]][j][k] == 7){};

}

}

if (ime.Z < demon.Z[i]){

if (demon.X[i] > ime.X){ demon.X[i]--; }

else{

if (demon.Y[i] > ime.Y){ demon.Y[i]--; }

else {

if (demon.X[i] < ime.X){ demon.X[i]++; }

else{

if (demon.Y[i] < ime.Y){ demon.Y[i]++; }

}

}

}

}

}

//падение демонов.

while (tip.tippolya[demon.Z[i]][demon.X[i]][demon.Y[i]] == 7){

demon.Z[i] -= 1;

};

while (){

};

}

}

\*/