Морський бій. Спрощена версія (немає бази)

Створюємо випадковим чином певну кількість кораблів, які можуть переміщуватися між пострілами.

--------- Ship.h ----------

#pragma once

#include <string>

using namespace std;

class Ship

{

public:

//Описуємо поля

int x, y;

int speed;

double health;

int projetailCount; //Кількість снарядів

//Методи доступу

//Конструктори

Ship(int x, int y, int speed, double health, int projetailCount);

Ship(int x, int y, int speed, int projetailCount);

Ship(int x, int y);

Ship(int projetailCount);

Ship();

//Інші методи

// -1 мертвий, 0-мимо, 1-ранили, 2-убили

int shoot(int shootX, int shootY, double damage);

void move();

void move(int newX, int newY);

//toString()

string toString();

};

-------------------- Ship.cpp ----------------------

#include "stdafx.h"

#include "Ship.h"

#include<stdlib.h>

Ship::Ship(int x, int y, int speed, double health, int projetailCount):x(x),y(y),speed(speed),

health(health),projetailCount(projetailCount)

{}

Ship::Ship(int x, int y, int speed, int projetailCount):Ship(x, y, speed, 100, projetailCount)

{

}

Ship::Ship(int x, int y):Ship(x, y, 1+rand() % 5, 100, 10)

{

}

Ship::Ship(int projetailCount): Ship(rand() % 5, rand() % 5, 1+rand() % 5, 100, projetailCount)

{

}

Ship::Ship(): Ship(10)

{

}

int Ship::shoot(int shootX, int shootY, double damage)

{

if (shootX != x || shootY != y)

return 0;

else

{

if (health <= 0)

return -1;

if (health > damage)

{

health -= damage;

return 1;

}

else

{

health = 0;

return 2;

}

}

return 0;

}

void Ship::move()

{

int shiftX = rand() % speed;

int shiftY = rand() % speed;

move(x + shiftX, y + shiftY);

}

void Ship::move(int newX, int newY)

{

x = newX % 5;

y = newY % 5;

}

string Ship::toString()

{

return "Position ("+to\_string(x)+", "+to\_string(y)+"), health = "+to\_string(health);

}

----------------------- Game.h -----------------

#pragma once

#include "Ship.h"

class Game

{

public:

int shipCount;

int projetileCount;

Ship\* ships;

//Base\* base;

Game(int shipCount, int projetileCount);

Game();

void play();

~Game();

};

------------------ Game.cpp ------------------

#include "stdafx.h"

#include "Game.h"

#include <iostream>

using namespace std;

Game::Game(int shipCount, int projetileCount):shipCount(shipCount),projetileCount(projetileCount)

{

ships = new Ship[shipCount]; //Викликається конструкор без параметрів для кожного корабля (неявно)

//base = new Base(....);

}

Game::Game():Game(3,5)

{

}

void Game::play()

{

int diedShipCount = 0;

do

{

//Вводимо координати пострілу

int shootPositionX;

int shootPositionY;

printf("Shoot position x:");

cin >> shootPositionX;

printf("Shoot position y:");

cin >> shootPositionY;

projetileCount--;

//Аналізуємо

int woundedCount = 0;

int diedCount = 0;

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

{

switch (ships[i].shoot(shootPositionX, shootPositionY, 35))

{

case 1: woundedCount++;

case 0: ships[i].move();

printf("%s\n", ships[i].toString().data());

break;

case 2: diedCount++;

break;

}

}

diedShipCount += diedCount;

printf("-----------------------------\n");

printf("Died :%d, wounded: %d \n",diedCount, woundedCount);

} while (diedShipCount<shipCount && projetileCount>0);

if (diedShipCount < shipCount)

{

printf("Loooser!\n");

}

else

printf("You won!!!!");

}

Game::~Game()

{

delete[] ships;

}

================= main =================

// ConsoleApplication29.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include "Game.h"

using namespace std;

int main()

{

Game\* game = new Game(10, 17);

game->play();

system("pause");

return 0;

}