**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Преподаватель, профессор,  департамент программной инженерии,  док. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. И. Легалов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

**Микропроект №2**

**Пояснительная записка**

Исполнитель:  
студент группы БПИ195  
\_\_\_\_\_\_\_\_\_\_ / К. В. Жуковский /  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Москва 2020**

1. Текст задания

Военная задача. Анчуария и Тарантерия – два крохотных латиноамериканских государства, затерянных в южных Андах. Диктатор Анчуарии, дон Федерико, объявил войну диктатору Тарантерии, дону Эрнандо. У обоих диктаторов очень мало солдат, но очень много снарядовдля минометов, привезенных с последней американской гуманитарной помощью. Поэтому армии обеих сторон просто обстреливают наугад территорию противника, надеясь поразить что-нибудь ценное. Стрельба ведется по очереди до тех пор, пока либо не будут уничтожены все цели, либо стоимость потраченных снарядов не превысит суммарную стоимость всего того, что ими можно уничтожить. Создать многопоточное приложение, моделирующее военные действия.

1. Описание алгоритма
2. У пользователя запрашиваются размеры государств(для баланса один размер используется для обоих государств) и цена боеприпаса.
3. Потом две стороны начинают обстрел друг друга в паралельных потоках:
4. Проверятся состояние на соответсвие условиям стрельбы
5. Наугад выбирается точка для выстрела(с использованием rand())
6. Если одна страна полностью уничтожена, то выводится сообщение о победе ее соперника и прорамма прекращает работу
7. Если одна из сторон прекратила стрельбу из-за убыточных результатов, то программа дожидается результата стрельбы второго государства
8. Если оба государства не победили, то выводится сообщение о перемирии

3. Текст программы

#include <iostream>

#include <vector>

#include <thread>

#include <random>

using namespace std;

/// <summary>

/// Класс ячейки

/// У каждой ячейки своя ценность

/// </summary>

class Cell

{

public:

Cell(){

cost = rand() % 10000;

is\_alive = true;

}

int get\_cost(){

return cost;

}

bool get\_is\_alive(){

return is\_alive;

}

void set\_is\_alive() {

is\_alive = false;

}

private:

int cost;

bool is\_alive;

};

/// <summary>

/// Класс поля страны

/// </summary>

class Field

{

public:

Field(int high, int wight, int cost){

sum\_cost = 0;

spended\_money = 0;

cost\_of\_ammo = cost;

this->high = high;

this->wight = wight;

field.resize(high);

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

field[i].resize(wight);

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

field[i][j] = Cell();

sum\_cost += field[i][j].get\_cost();

}

}

}

int get\_high(){

return high;

}

int get\_widht(){

return wight;

}

int get\_sum\_cost(){

return sum\_cost;

}

void take\_gamage(int x, int y){

field[x][y].set\_is\_alive();

}

bool spend(int max){

spended\_money += cost\_of\_ammo;

return (max > spended\_money);

}

bool destroyed(){

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

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

if (field[i][j].get\_is\_alive())

return false;

}

}

return true;

}

private:

vector<vector<Cell>> field;

int cost\_of\_ammo;

int spended\_money;

int sum\_cost;

int wight, high;

};

/// <summary>

/// Метод атаки, выполняется пока соблюдаются условия

/// Ведения огня, если кто-то победил, то программа завершается сразу

/// С информирующим сообщением

/// </summary>

/// <param name="attacker">Атакующая сторона</param>

/// <param name="defender">Атакуемая сторона</param>

/// <param name="name">Название атакующей страны</param>

void attack(Field attacker, Field defender, string name){

while (attacker.spend(defender.get\_sum\_cost()) && !defender.destroyed())

{

int x = rand() % defender.get\_high(); //Случайно выбирается цель

int y = rand() % defender.get\_widht();

defender.take\_gamage(x, y);

}

if(defender.destroyed()){

cout << name << " победила " << endl; //В случае победы

exit(0);

}

cout << name << " прекратила стрельбу, так как это стало невыгодно" << endl; //В случае прекращения стрельбы

}

int main()

{

setlocale(LC\_ALL, "Russian");

int x, y, cost;

cout << "Добро пожаловать в симулятор войны:\nВам будет предложено задать размеры государств\n" <<

"Для баланса они будут использованы для обоих стран\nИ цену снарядов" << endl;

cout << "Введите высоту страны(целое число)" << endl;

cin >> x;

cout << "Введите ширину страны(целое число)" << endl;

cin >> y;

cout << "Введите цену снаряда(целое число)" << endl;

cin >> cost;

Field Anch = Field(x, y, cost);

Field Tar = Field(x, y, cost);

cout << "Война началась" << endl;

thread th1 = thread(attack, Anch, Tar, "Анчуария");

thread th2 = thread(attack, Tar, Anch, "Tарантерия");

th1.join();

th2.join();

cout << "Диктаторы пришли к перемирию" << endl;

}