**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»**

Факультет компьютерных наук

Департамент программной инженерии

**Отчет к домашнему заданию**

**По дисциплине**

**«Архитектура вычислительных систем»**

Работу выполнил:

Студент группы БПИ-194 Горбач И.В.

Вариант 7

**Москва 2020**

**Задание**

7. 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 снарядов для 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. **Модель**

Итеративный параллелизм используется для реализации нескольких потоков (часто идентичных), каждый из которых содержит циклы. Потоки программы, описываются итеративными функциями и работают совместно над решением одной задачи.

**Решение**

Для реализации данной задачи была использовано 2 библиотеки: threads и mutex

Ввод: входных данных нет, можно в самом коде менять площадь территории городов, стоимость одного снаряда.

using namespace std;  
  
mutex mtx;  
const unsigned int dim\_1=3; //количество клеток по вертикали  
const unsigned int dim\_2=3; //количество клеток по горизонтали  
  
///  
/// \param min нижнее значение  
/// \param max верхнее значение  
/// \return рандомное число между мин и макс  
int getRandomNumber(int min, int max)  
{  
 static const double fraction = 1.0 / (static\_cast<double>(RAND\_MAX) + 1.0);  
 // Равномерно распределяем рандомное число в нашем диапазоне  
 return static\_cast<int>(rand() \* fraction \* (max - min + 1) + min);  
}

Площадь территории заполняется рандомайзером

**КЛАСС CITY**

Поля в классе City:

class City{  
public:  
 string name;//Название города  
 string name\_of\_dictator; //имя диктатор  
 int territory[dim\_1][dim\_2];//территория города - матрица по дефолту 10 на 10  
 int cost\_summ=0; //общая сумма клеток города  
 //конструктор класса с параметром имени диктатора

Методы класса city:

City(string city\_name,string dict\_name);//инициализатор конструктора  
 ~City();//иниц деструктора  
  
 void Create\_Territory(){  
 for (auto & i : territory) {  
 for (int & j : i) {  
 int cost=getRandomNumber(0,100);  
 j=cost;// rand mod 100 for creating normal values  
 cost\_summ+=cost; //и в сумму тоже добавляем ранд  
 }  
 }  
 Print\_Territory();  
 }  
 void Print\_Territory(){  
 cout<< "territory: "<< this->name<<endl << "Current cost of territory= "<< cost\_summ << endl;  
 for (auto & i : territory) {  
 for (int j : i) {  
 cout << j << " ";  
 }  
 cout << endl;  
 }  
 }  
};

Метод Create\_Territory отвечает за инициализацию поля territory-что являтся территорией города. Метод Print\_Territory отвечает за вывод на экран территории города.

КОНСТРУТКОР И ДЕСТРУКТОР

///  
/// \param city\_name название города  
/// \param dict\_name имя диктатора  
City::City(string city\_name, string dict\_name) {  
 this->name\_of\_dictator=dict\_name;  
 this->name=city\_name;  
 this->Create\_Territory();  
}  
  
City::~City() {  
}

МЕТОД MakeMove()  
Этот метод, метод совершения выстрела по другому городу

void MakeMove(City& other, int first\_rand,int second\_rand,thread& thread){  
 mtx.lock();//блокируем мьютекс  
 cout <<"----------------------------------------------------------------------------------" << endl;  
 cout <<"id потоковой пушки, стреляющей в : "<<other.name<<" равен: "<<thread.get\_id()<<endl;//выводим значение текущего потока  
 mtx.unlock();//разблокируем мьютекс  
 mtx.lock();//блокируем мьютекс  
 cout<<other.name << " под огнем! Атакована территория: "<<first\_rand+1<<" "<<second\_rand+1 << endl;//вывод атакованного города  
 if (other.territory[first\_rand][second\_rand]==0){  
 cout << "Здесь было пусто, возможно сюда уже стреляли"<< endl;  
 }  
 else {  
 cout << "Территория была обстрелена, стоимость ущерба составила: "<<other.territory[first\_rand][second\_rand]<<endl;  
 other.cost\_summ-=other.territory[first\_rand][second\_rand];  
 other.territory[first\_rand][second\_rand] = 0;  
 }  
 this\_thread::sleep\_for(chrono::milliseconds(200));//засыпаем на 200 миллисекунд  
 other.Print\_Territory();  
 cout <<"Охлаждаем пушки!" << endl;  
 mtx.unlock();//разблокируем мьютекс  
}

Сначала мьютекс закрывается, чтобы правильно вывести разделительные полосы и номер текущего выполняемого потока, потом мьютекс разблокируется и блокируется заново, происходит обстрел вражеской территории, обстреливаемая территория выбирается случайно, если раньше значение стоимости вражеской территории было равно 0, то выводится: «Здесь было пусто, возможно сюда уже стреляли», если раньше там было что то, то выводится «"Территория была обстрелена, стоимость ущерба составила: "<<other.territory[first\_rand][second\_rand]», из общей стоимости всей территории вычитается стоимость разбомбленной территории и стоимость разбомбленной территории становится равно 0. Затем текущий поток отправляется спать на 200 миллисекунд, имитируя перезарядку, происходит вывод обстрелянного города в консоль и мьютекс разблокирывается.

МЕТОД main:

int main() {  
 cout<<"MAIN TREAD ID:"<<this\_thread::get\_id()<<endl;//номер мэйн потока  
 system("chcp 65001");  
 int bullet\_cost = 5;//стоимость 1 пули  
 City Anchyuria = \*new City("Anchyuria","Don Federico");//создание анчурии  
 City Taranteria = \*new City("Taranteria","Don Hernando");//создание тарантерии  
 //создание треда анчурии и треда тарантерии, на онове MakeMove  
 while (Anchyuria.cost\_summ > bullet\_cost&&Anchyuria.cost\_summ>0&& Taranteria.cost\_summ>0 && Taranteria.cost\_summ > bullet\_cost) {  
 thread anch(MakeMove,ref(Taranteria),getRandomNumber(0,dim\_1-1),getRandomNumber(0,dim\_2-1),ref(anch));  
 thread taran(MakeMove, ref(Anchyuria),getRandomNumber(0,dim\_1-1),getRandomNumber(0,dim\_2-1),ref(taran));  
 anch.join();  
 taran.join();  
 }  
 if (Anchyuria.cost\_summ<0||Anchyuria.cost\_summ<bullet\_cost){  
 cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ ПРОИГРАЛА:"<<Anchyuria.name<<endl;  
 }  
 else if (Taranteria.cost\_summ<0||Taranteria.cost\_summ<bullet\_cost){  
 cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ ПРОИГРАЛА:"<<Taranteria.name<<endl;  
 }  
 system("exit");  
 return 0;  
}

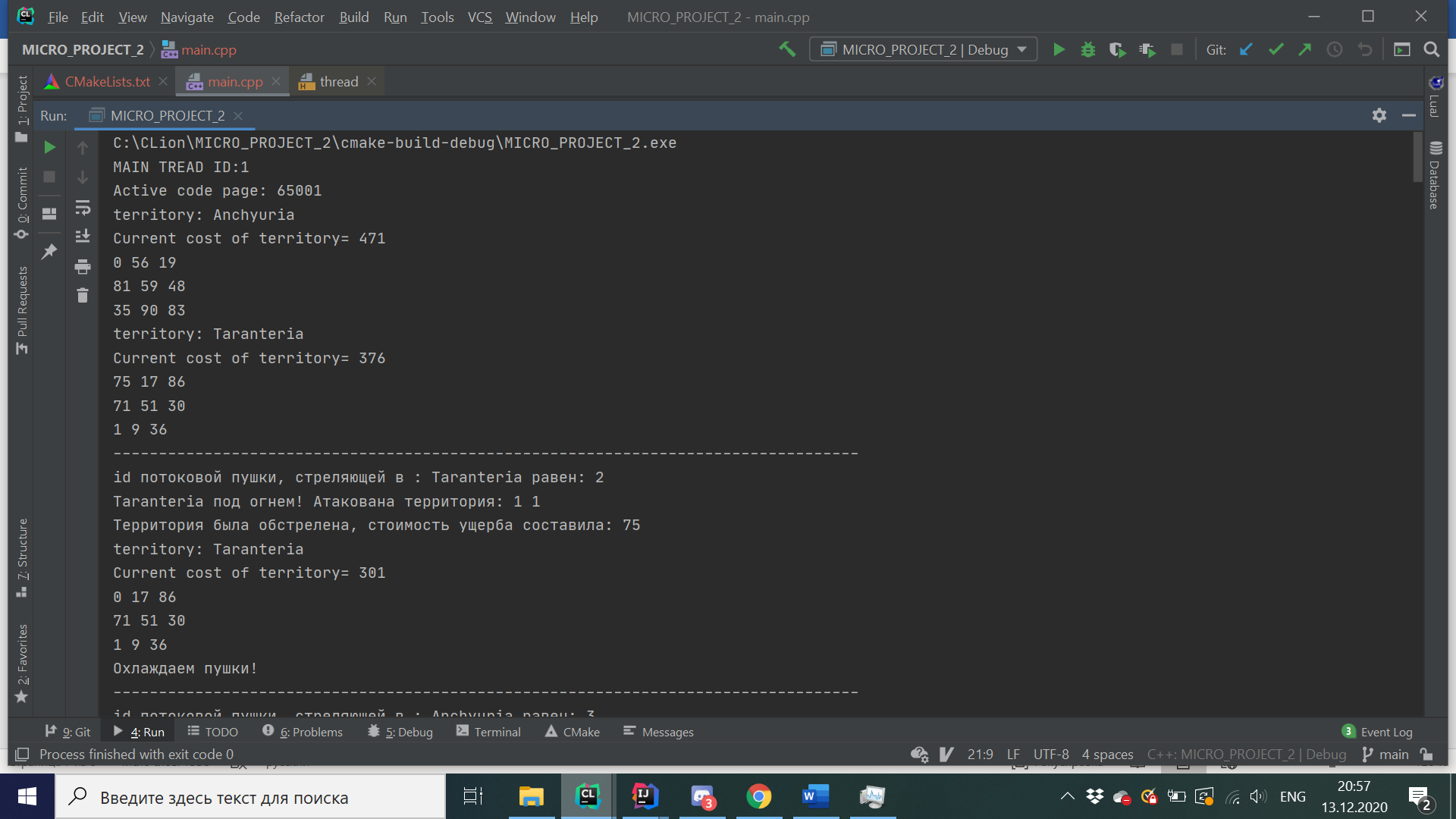
Сначала в консоль выводится айди основного потока программы, затем ставится русская локаль. Создается переменная для стоимости одного патрона. Создается 2 экземпляра класса City, цикл while выполняется пока стоимость территории больше чем 0 или чем стоимость 1 пули. В теле цикла создается 2 потока(костыль, не знал как сделать иначе), один поток имитирует обстрел Тарантерии Анчурией, другой поток имитирует обстрел Анчурии Тарантерией. Параметрами являются обстреливаемые города, два интовых значения, отвечающие за строку и столбец в матрице вражеской территории и поток, для вывода id текущего потока. Затем у обоих потоков вызывается метод join(). После выхода из тела цикла, осуществляется проверка на проигравшего, и завершение программы.

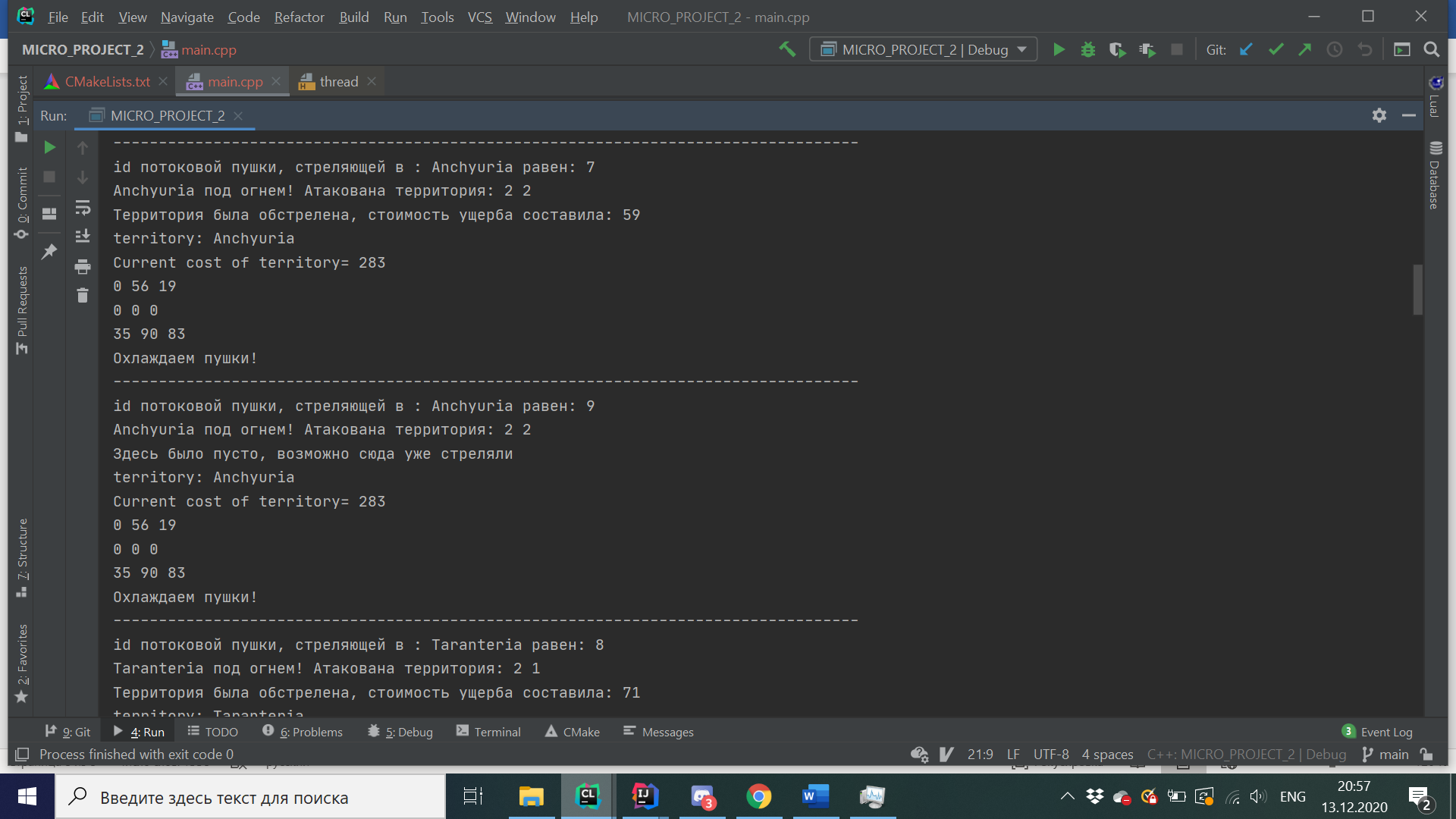
**Код программы**

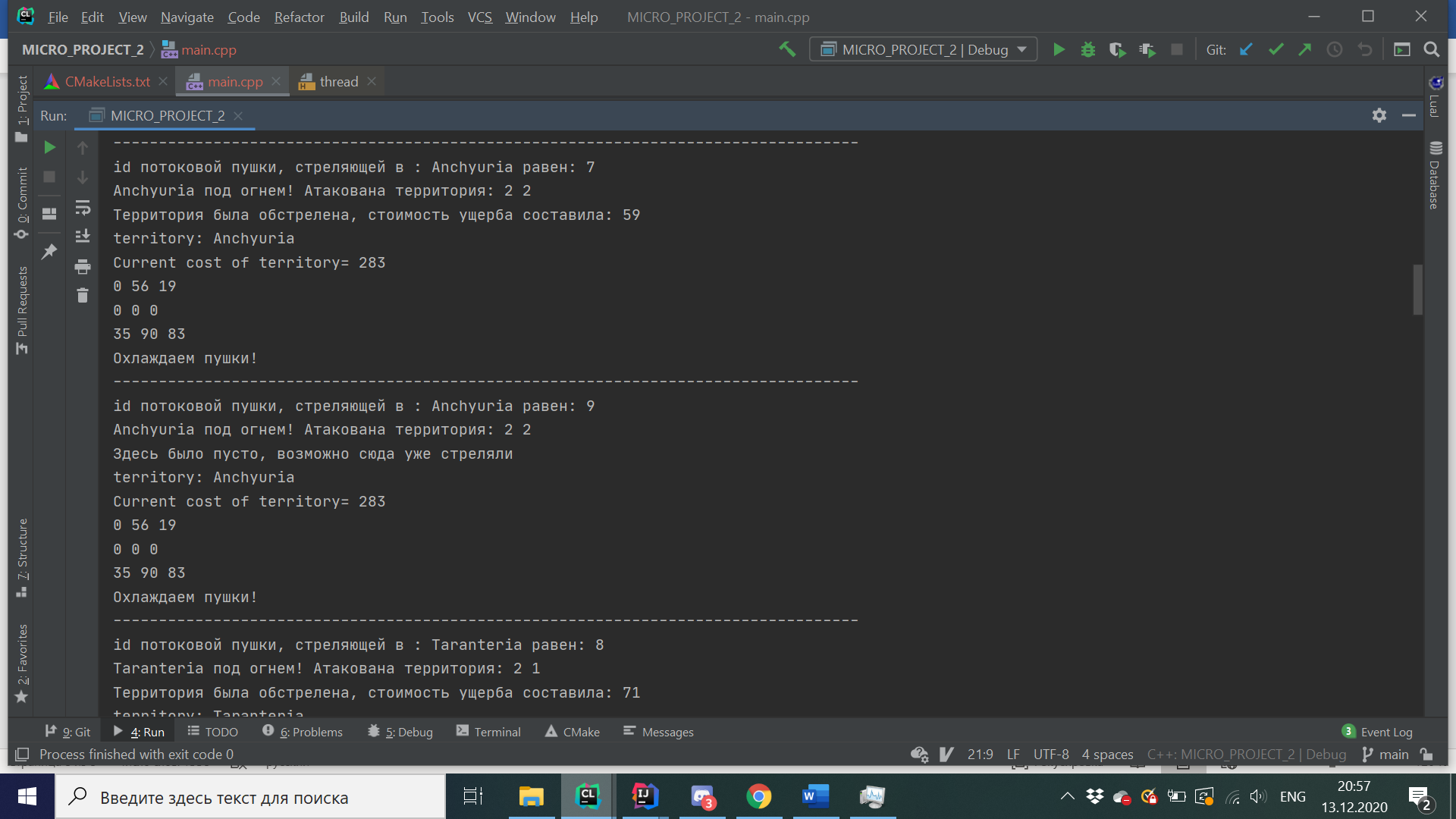
#include <iostream>  
#include "string"  
#include <thread>  
#include <mutex>  
  
/\*  
 \* Горбач Илья Вадимович  
 \* БПИ 194  
 \* Вариант №7  
 \* 7. Военная задача. Анчуария и Тарантерия – два крохотных  
латиноамериканских государства, затерянных в южных Андах. Диктатор  
Анчуарии, дон Федерико, объявил войну диктатору Тарантерии, дону  
Эрнандо. У обоих диктаторов очень мало солдат, но очень много снарядов  
для 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.  
  
 \*/  
using namespace std;  
  
mutex mtx;  
const unsigned int dim\_1=3; //количество клеток по вертикали  
const unsigned int dim\_2=3; //количество клеток по горизонтали  
  
///  
/// \param min нижнее значение  
/// \param max верхнее значение  
/// \return рандомное число между мин и макс  
int getRandomNumber(int min, int max)  
{  
 static const double fraction = 1.0 / (static\_cast<double>(RAND\_MAX) + 1.0);  
 // Равномерно распределяем рандомное число в нашем диапазоне  
 return static\_cast<int>(rand() \* fraction \* (max - min + 1) + min);  
}  
//Класс город  
class City{  
public:  
 string name;//Название города  
 string name\_of\_dictator; //имя диктатор  
 int territory[dim\_1][dim\_2];//территория города - матрица по дефолту 10 на 10  
 int cost\_summ=0; //общая сумма клеток города  
 //конструктор класса с параметром имени диктатора  
 City(string city\_name,string dict\_name);//инициализатор конструктора  
 ~City();//иниц деструктора  
  
 void Create\_Territory(){  
 for (auto & i : territory) {  
 for (int & j : i) {  
 int cost=getRandomNumber(0,100);  
 j=cost;// rand mod 100 for creating normal values  
 cost\_summ+=cost; //и в сумму тоже добавляем ранд  
 }  
 }  
 Print\_Territory();  
 }  
 void Print\_Territory(){  
 cout<< "territory: "<< this->name<<endl << "Current cost of territory= "<< cost\_summ << endl;  
 for (auto & i : territory) {  
 for (int j : i) {  
 cout << j << " ";  
 }  
 cout << endl;  
 }  
 }  
};  
///  
/// \param city\_name название города  
/// \param dict\_name имя диктатора  
City::City(string city\_name, string dict\_name) {  
 this->name\_of\_dictator=dict\_name;  
 this->name=city\_name;  
 this->Create\_Territory();  
}  
  
City::~City() {  
}  
void MakeMove(City& other, int first\_rand,int second\_rand,thread& thread){  
 mtx.lock();//блокируем мьютекс  
 cout <<"----------------------------------------------------------------------------------" << endl;  
 cout <<"id потоковой пушки, стреляющей в : "<<other.name<<" равен: "<<thread.get\_id()<<endl;//выводим значение текущего потока  
 mtx.unlock();//разблокируем мьютекс  
 mtx.lock();//блокируем мьютекс  
 cout<<other.name << " под огнем! Атакована территория: "<<first\_rand+1<<" "<<second\_rand+1 << endl;//вывод атакованного города  
 if (other.territory[first\_rand][second\_rand]==0){  
 cout << "Здесь было пусто, возможно сюда уже стреляли"<< endl;  
 }  
 else {  
 cout << "Территория была обстрелена, стоимость ущерба составила: "<<other.territory[first\_rand][second\_rand]<<endl;  
 other.cost\_summ-=other.territory[first\_rand][second\_rand];  
 other.territory[first\_rand][second\_rand] = 0;  
 }  
 this\_thread::sleep\_for(chrono::milliseconds(200));//засыпаем на 200 миллисекунд  
 other.Print\_Territory();  
 cout <<"Охлаждаем пушки!" << endl;  
 mtx.unlock();//разблокируем мьютекс  
}  
  
int main() {  
 cout<<"MAIN TREAD ID:"<<this\_thread::get\_id()<<endl;//номер мэйн потока  
 system("chcp 65001");  
 int bullet\_cost = 5;//стоимость 1 пули  
 City Anchyuria = \*new City("Anchyuria","Don Federico");//создание анчурии  
 City Taranteria = \*new City("Taranteria","Don Hernando");//создание тарантерии  
 //создание треда анчурии и треда тарантерии, на онове MakeMove  
 while (Anchyuria.cost\_summ > bullet\_cost&&Anchyuria.cost\_summ>0&& Taranteria.cost\_summ>0 && Taranteria.cost\_summ > bullet\_cost) {  
 thread anch(MakeMove,ref(Taranteria),getRandomNumber(0,dim\_1-1),getRandomNumber(0,dim\_2-1),ref(anch));  
 thread taran(MakeMove, ref(Anchyuria),getRandomNumber(0,dim\_1-1),getRandomNumber(0,dim\_2-1),ref(taran));  
 anch.join();  
 taran.join();  
 }  
 if (Anchyuria.cost\_summ<0||Anchyuria.cost\_summ<bullet\_cost){  
 cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ ПРОИГРАЛА:"<<Anchyuria.name<<endl;  
 }  
 else if (Taranteria.cost\_summ<0||Taranteria.cost\_summ<bullet\_cost){  
 cout<<"В ВОЙНЕ НЕ БЫВАЕТ ПОБЕДИТЕЛЕЙ, НО В ЭТОЙ ВОЙНЕ ПРОИГРАЛА:"<<Taranteria.name<<endl;  
 }  
 system("exit");  
 return 0;  
}

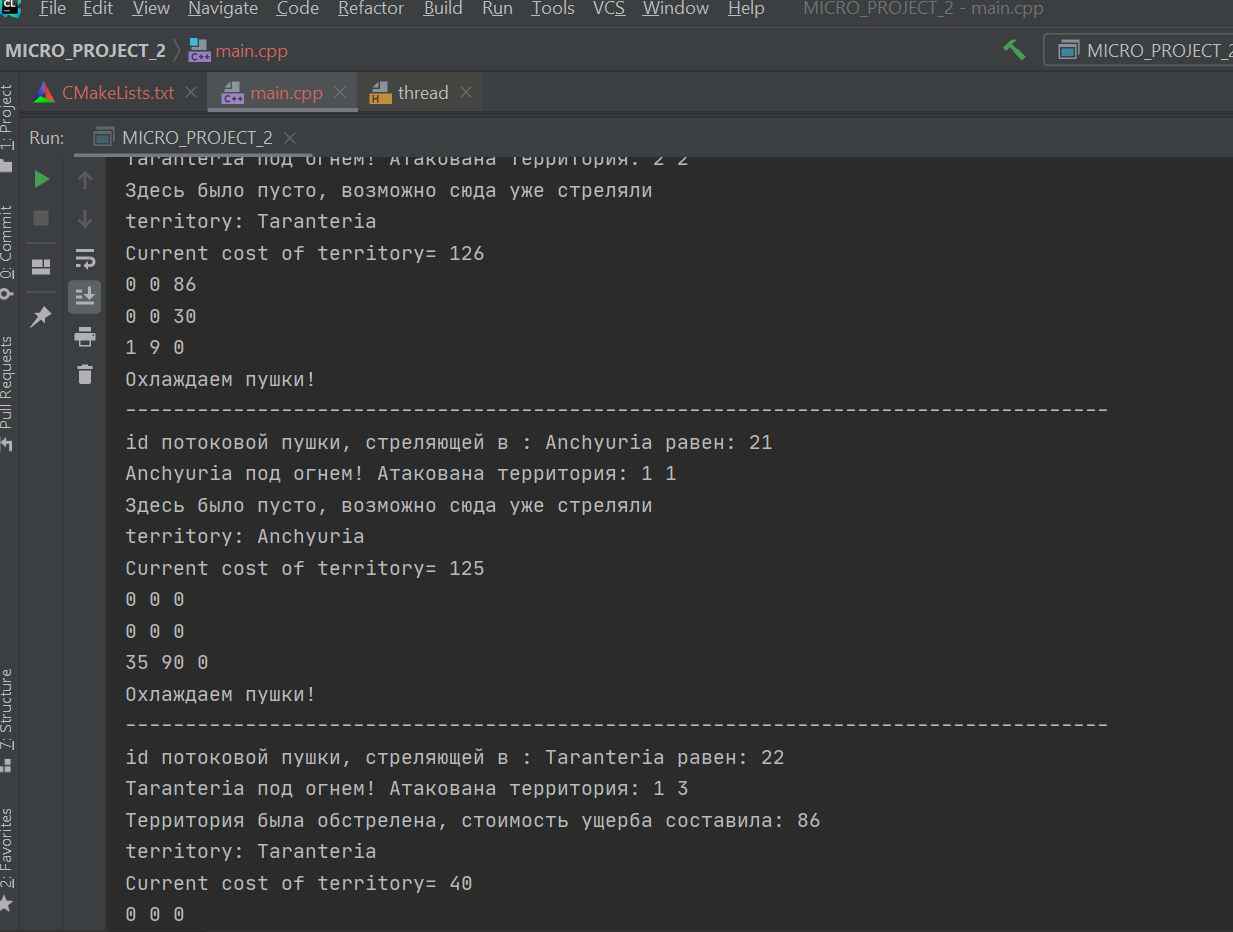
**Тестирование**

**Пример 1:**

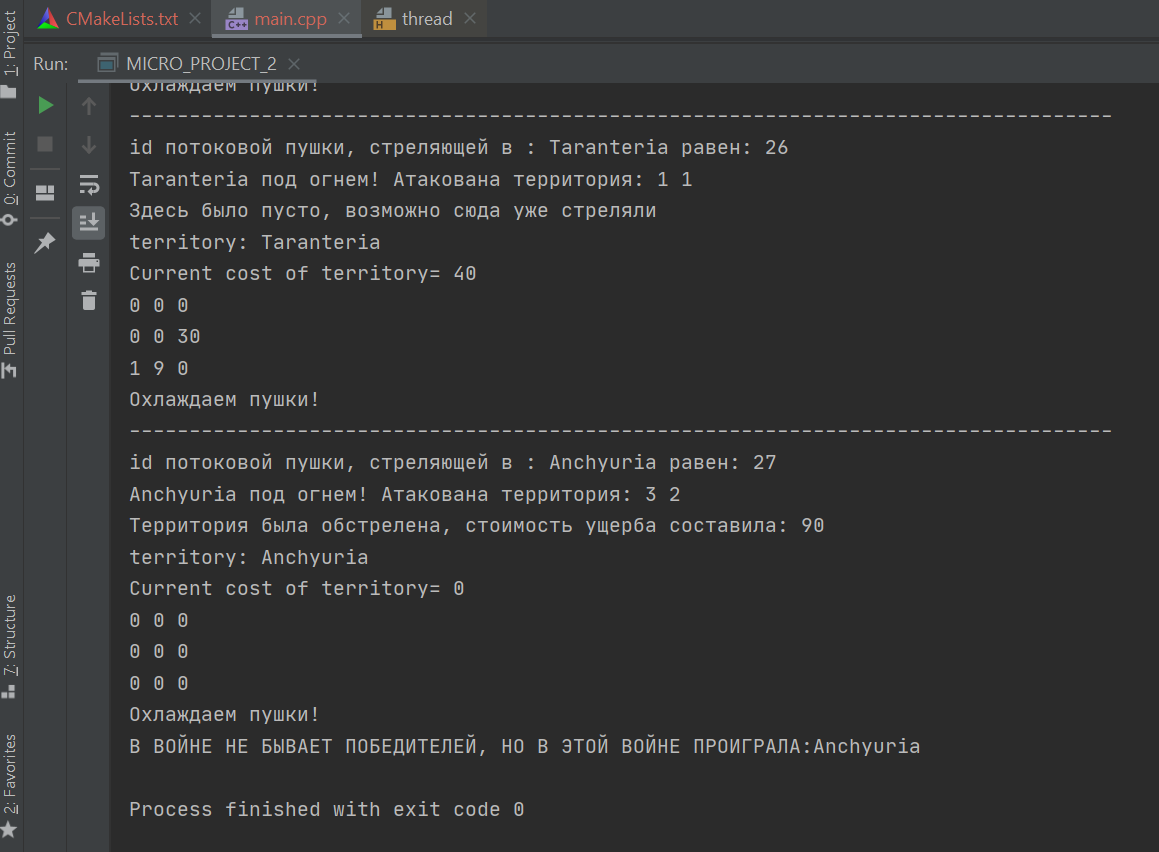








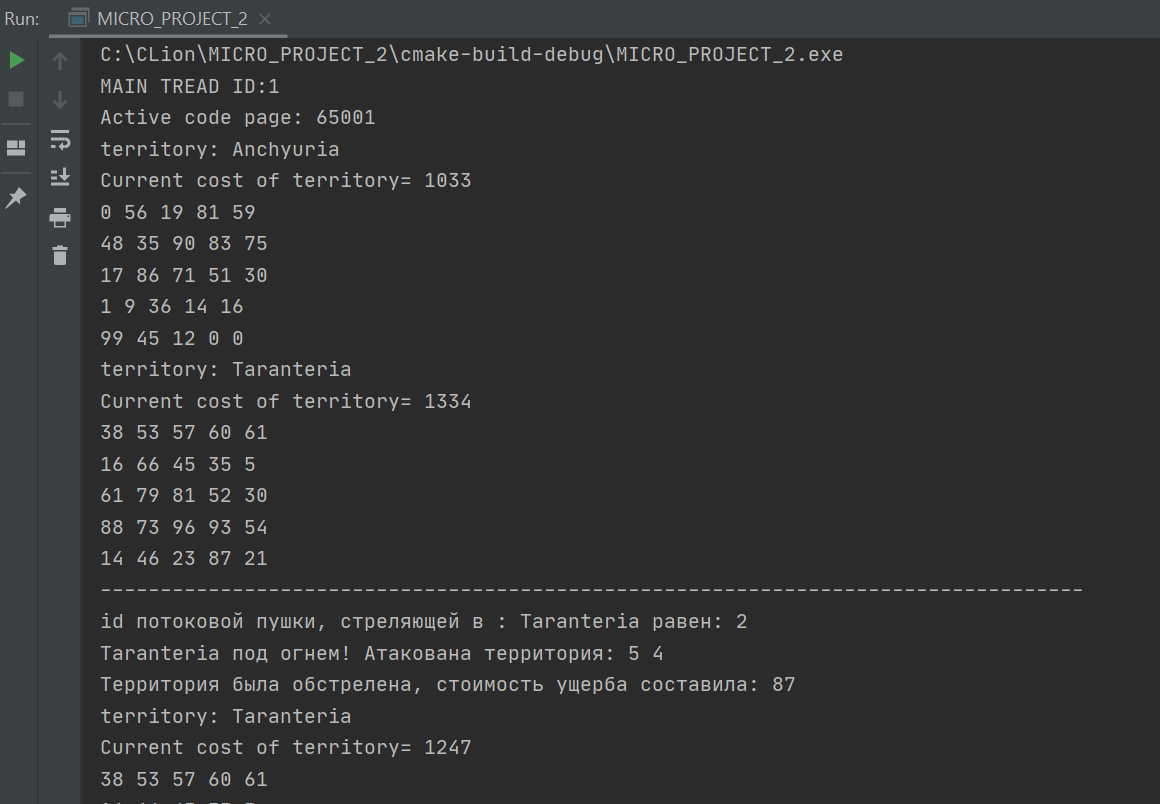
Конец примера 1:



Пример был выполнен для размера территории 3:3

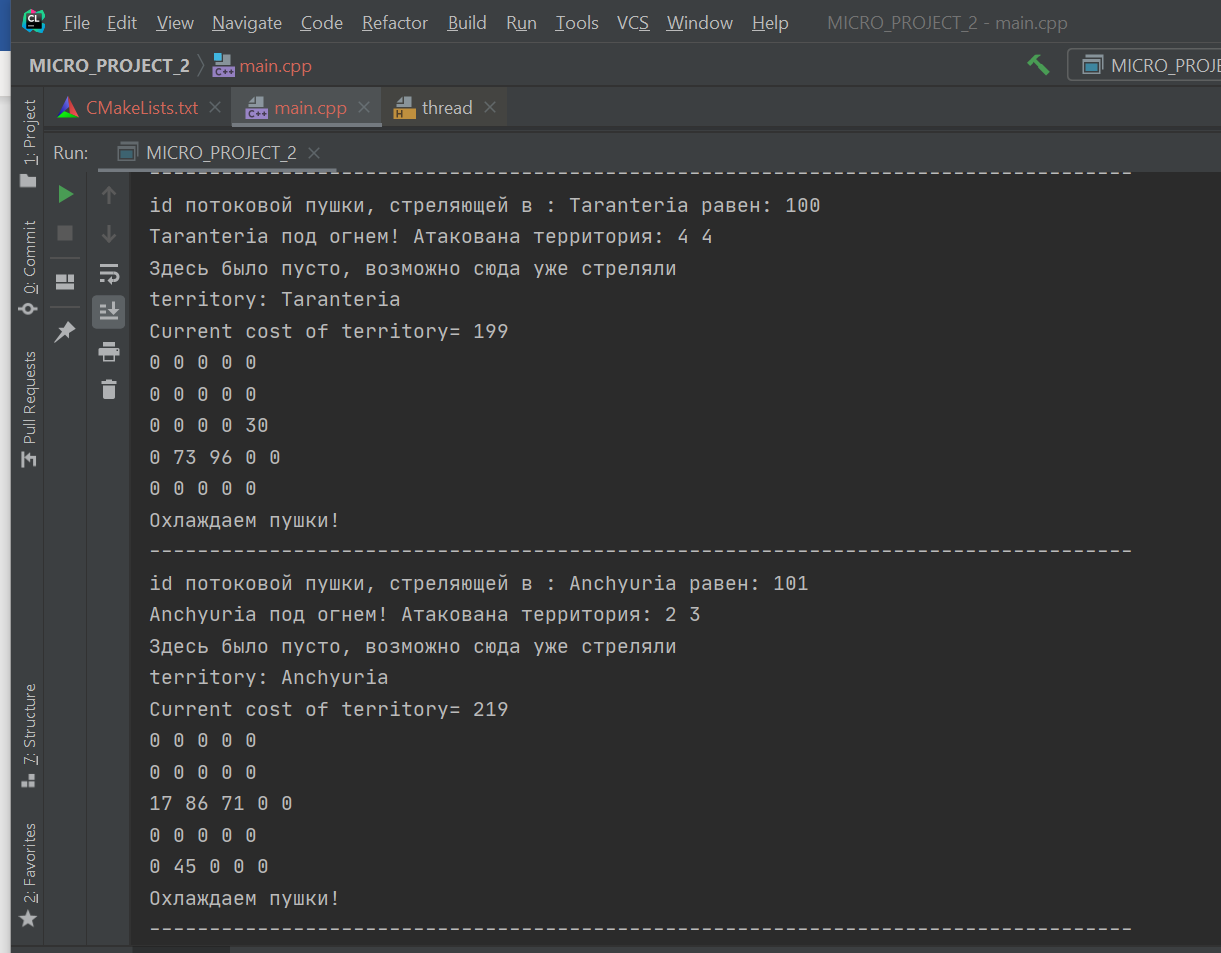
Пример 2 для территории 5 на 5:

Начало:

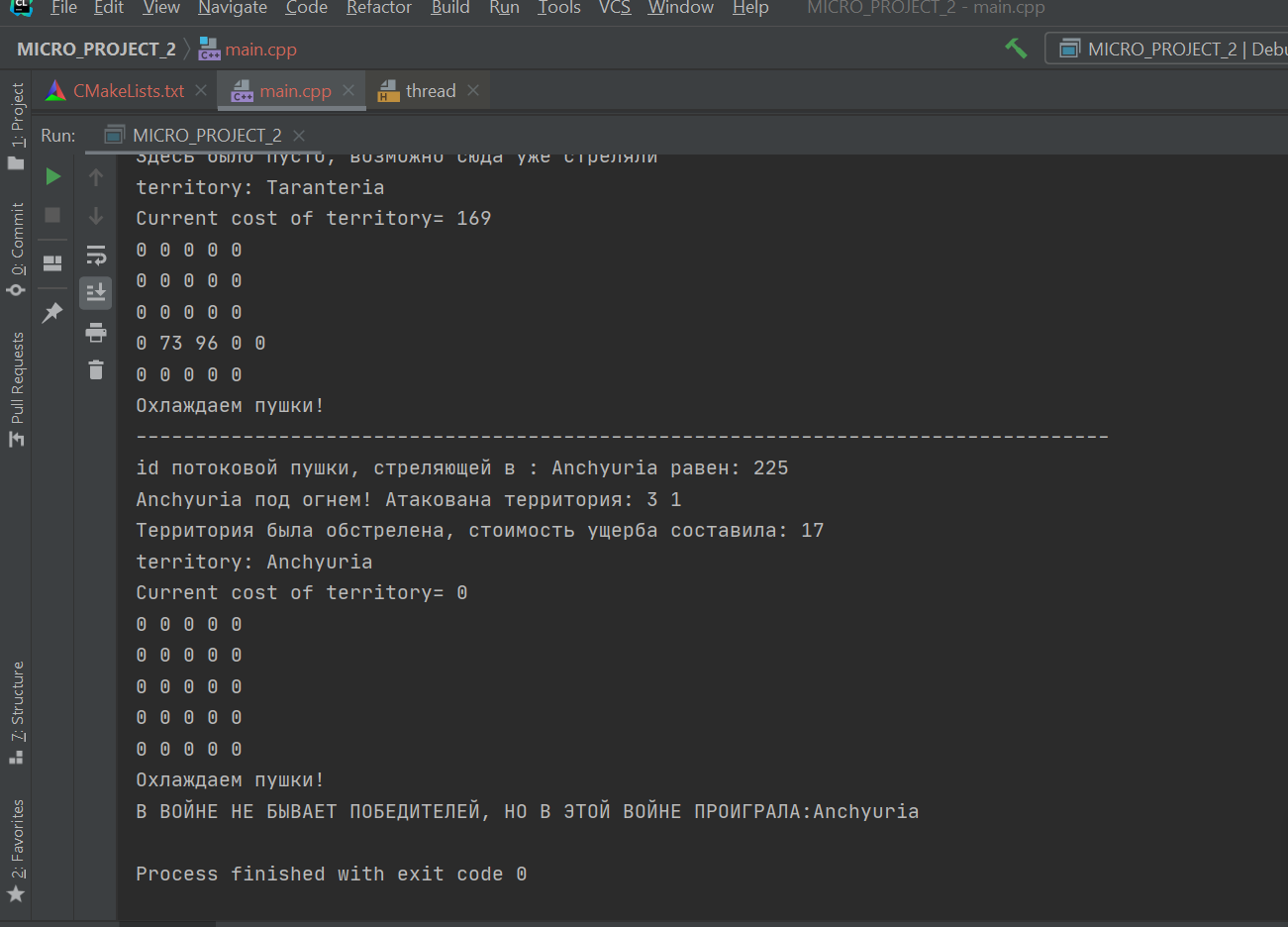


Выполнение идет довольно долго, в связи с засыпаниями на 200 миллисекунд и попытками попасть рандомайзером в ранее необстрелянную клетку города

Середина программы:



Конец программы:



**Список используемых источников**

1. https://plugins.jetbrains.com/plugin/8352-c-c--single-file-execution
2. https://coderoad.ru/32030935/
3. <https://ravesli.com/urok-71-generatsiya-sluchajnyh-chisel-funktsii-srand-i-rand/#toc-1>
4. <https://www.cyberforum.ru/cpp-beginners/thread956351.html>
5. <https://www.youtube.com/watch?v=DFgtADi0Mjo>
6. http://www.softcraft.ru/