

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

# Институт Информационных Технологий

# Кафедра инструментального и прикладного программного обеспечения (ИиППО)

**КУРСОВАЯ РАБОТА**

### по дисциплине

### «Объектно-ориентированное программирование»

### Тема курсовой работы:

**«**Баллистическая игра**»**

|  |  |  |
| --- | --- | --- |
| Студент группы | ИНБО-03-18 | Исмаилов И. Н. |
| Руководитель курсовой работы | Ассистент кафедры ИиППО | Хлебникова В.Л. |
| Работа представлена к защите | « » 2018 г. |  |
|  |  |  |
| «Допущен к защите» | « » 2018 г. |  |
|  |  |  |

Москва 2019

|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

# Институт Информационных Технологий

# Кафедра инструментального и прикладного программного обеспечения (ИиППО)

**Протокол заседания комиссии по защите курсовой работы**

### От « » 2019 г №

### Состав комиссии:

### Утверждена распоряжением заведующего кафедрой **ИиППО** от « » 2019 г. №

Слушали защиту курсовой работы: **«**Разработка программы по поддержанию личной коллекции в актуальном состоянии»

### по дисциплине: «Объектно-ориентированное программирование» студентки группы: ИНБО-03-18 Исмаилов И. Н.

### Во время защиты курсовой работы были заданы следующие вопросы:

### 1.

### 2.

### 3.

### Итоговая оценка выполнения и защиты курсовой работы: Члены комиссии:

|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

# Институт Информационных Технологий

# Кафедра инструментального и прикладного программного обеспечения (ИиППО)

**ЗАДАНИЕ**

**на курсовую работу**

### по дисциплине

## «Объектно-ориентированное программирование»

1. **Исполнитель:** студент Исмаилов И. Н. группа: ИНБО-03-18
2. **Тема работы: «**Разработка баллистической игры»
3. **Исходные данные:**

1) Индивидуальное задание на курсовую работу.

2) Среда разработки: С++.

3) Комплект разработчика приложений на языке C ++, руководство по языку программирования C ++, стандартные библиотеки классов C ++ .

4) Интернет-ресурсы и источники по теме курсовой работы.

1. **Перечень вопросов к разработке:**

## Разработать техническое задание на программный продукт.

## Изучить и научиться применять принципы поэтапной разработки и отладки программ средней сложности

## Проанализировать предметную область согласно заданию и разработать собственные классы, разработать реализацию классов, а также разработать алгоритмы, необходимые для функционирования программы.

## Разработать интерфейс для программы.

## Разработать тесты для проверки работоспособности программного продукта.

## Разработать руководство по использованию разработанного программного продукта.

## Перечень обязательных графических материалов: UML-диаграмма(ы), Листинги программы.

1. **Календарный план-график работы над проектом:**

## получение задания, поиск решения;

## разработка программы;

## создание программного кода;

## тестирование работы программы;

## оформление курсовой работы и сдача ее на проверку;

## защита курсовой работы;

1. **Срок предоставления к защите курсовой работы:** до « »

## 2019 г.

|  |  |
| --- | --- |
| Задание на курсовую работу | Руководитель курсовой работы: |
| выдано: « » 2019 г. | Хлебникова В.Л. |
|  | Подпись: |
| Задание на курсовую работу | Исполнитель курсовой работы: |
| получено: « » 2019 г. | Исмаилов И.Н. |
|  | Подпись: |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_bookmark0)

1. [ОБЩИЕ СВЕДЕНИЯ О ЗАДАНИИ 7](#_bookmark1)
2. [ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ 7](#_bookmark2)
   1. [Поэтапная разработка программы 7](#_bookmark3)
   2. [Проектирование классов 7](#_bookmark4)
   3. [UML диаграмма 13](#_bookmark5)
3. [ТЕСТИРОВАНИЕ ПРОГРАММЫ 15](#_bookmark6)
4. [РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 16](#_bookmark7)
   1. [Процедура установки 16](#_bookmark8)
   2. [Процедура удаления 16](#_bookmark9)
   3. [Запуск программы 16](#_bookmark10)
   4. [Эксплуатация программы 16](#_bookmark11)

[ЗАКЛЮЧЕНИЕ 2](#_bookmark12)2

[Список используемых источников 23](#_bookmark13)

[Приложение А. Исходный код программы](#_bookmark14) 24

# ВВЕДЕНИЕ

В ходе разработки проекта, будут использованы знания по Объективно-Ориентированному Программированию на языке С++.С++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования, этот язык достаточно большой и удобный для работы в нём.

“Баллистическая игра” является программой, в которой двум игрокам предстоит посоревноваться между собой в точности. Целью игры является уничтожение баллисты противника.

# ОБЩИЕ СВЕДЕНИЯ

# 1.1 Наименование программы:

# “Баллистическая игра”.

# 1.2 Назначение программы и ее основные функции

# 1) Программа рассчитывает траекторию полета камня.

# 2) Генерирует ход игрока.

# 3) Фиксирует попадание.

# 1.3 Функциональные ограничения

# 1)Программа работает только на кроме Windows.

# 2.ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ

# Поэтапная разработка программы

## Разработка программы состоит из следующих этапов:

## придумал идею;

## создание абстрактного класса bali;

## унаследование двух классов от абстрактного класса;

## описание методов класса(функции);

## написание main;

## 

## **Проектирование программы**

## Программа должна выполнять поставленные задачи:

## Игроки ходят по очереди;

## 2) Ход игрока заключается в выборе угла и массы камня;

## 3) Программа производит расчет траектории полета камня и рисует ее на экране;

## 4)Фиксирует попадание игрока в баллисту;

## **2.3 Проектирование классов**

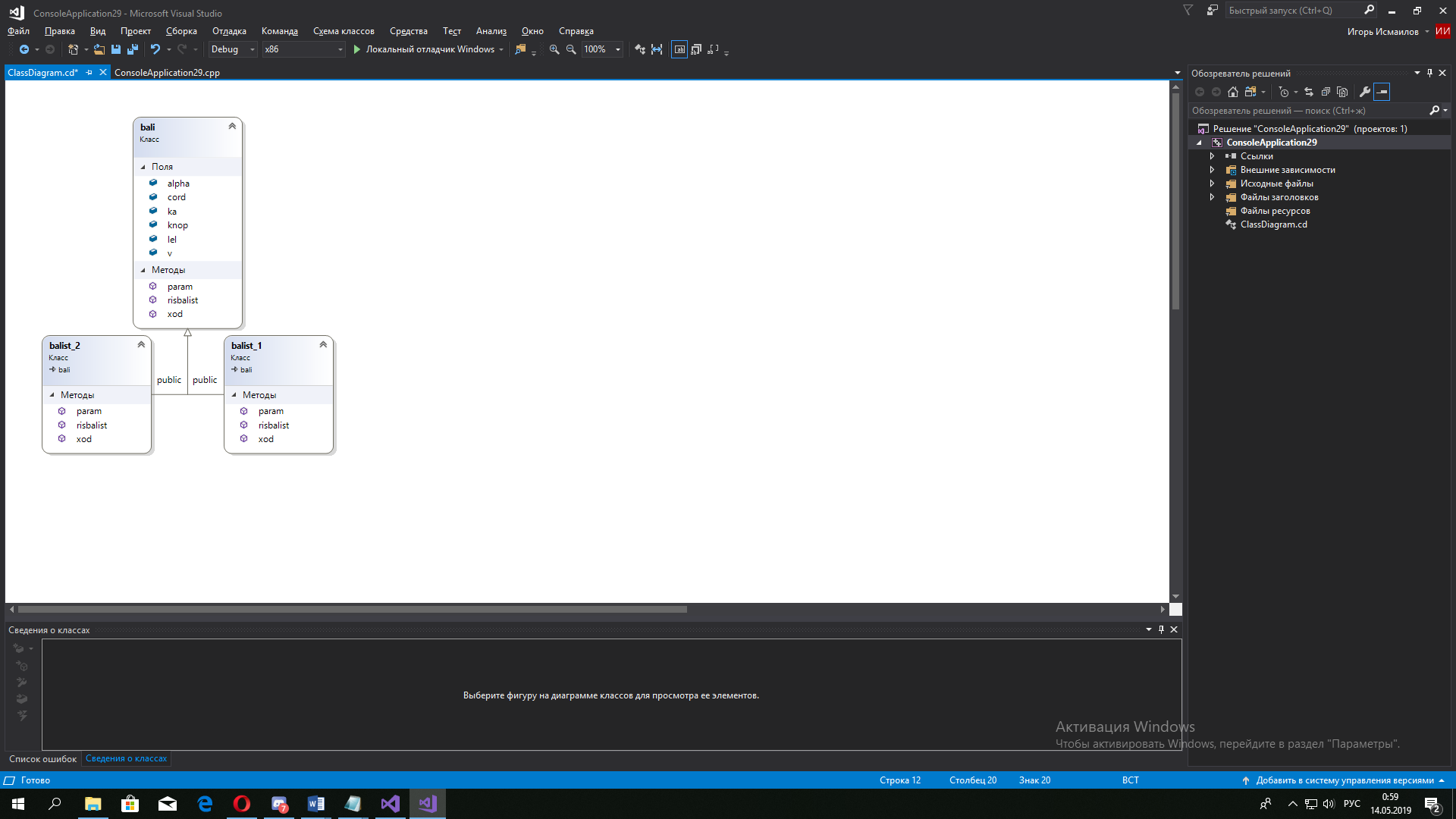


Рис 1.Общая UML диаграмма.

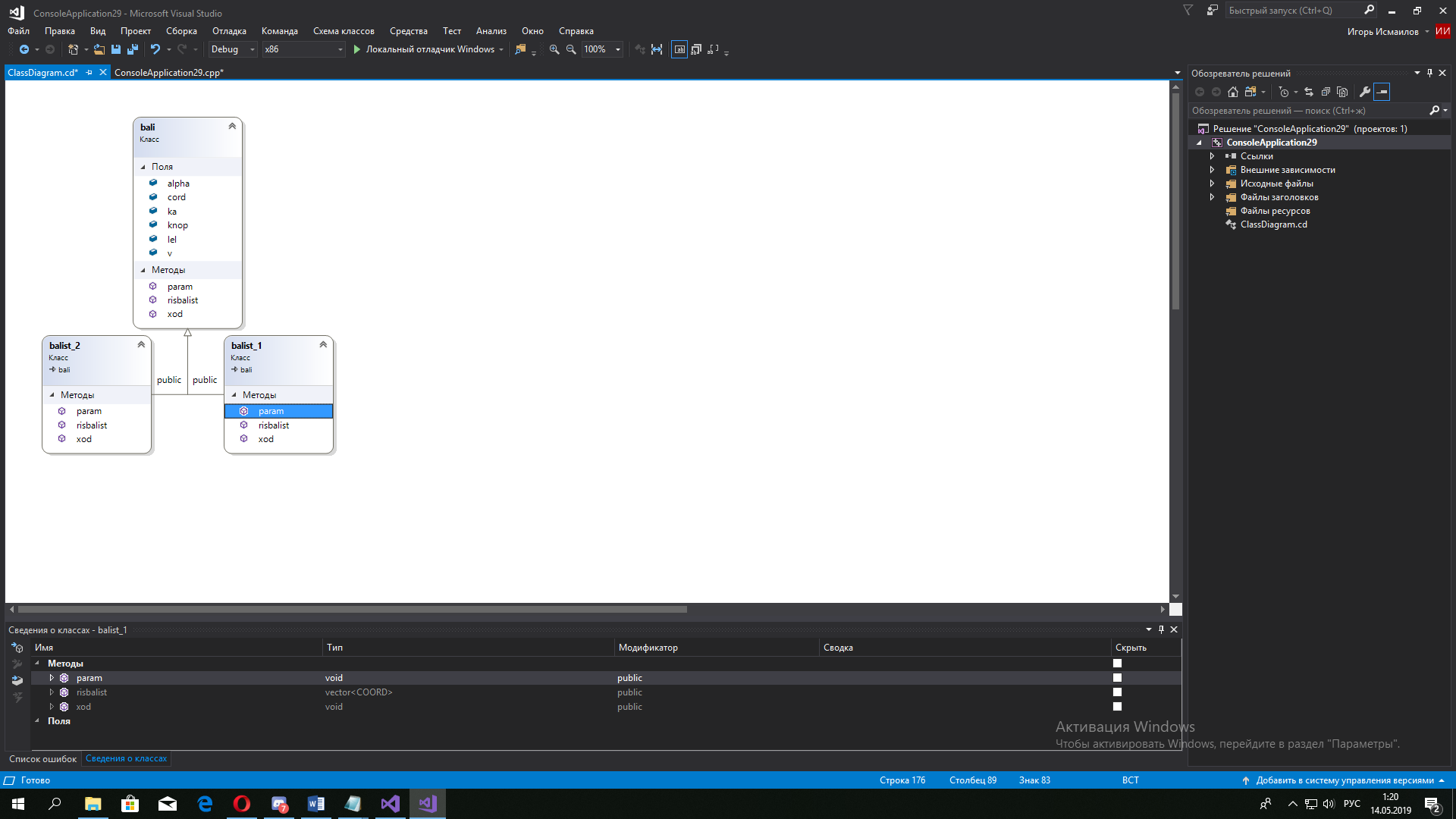


Рис 2.UML Диаграмма класса bali.

Абстрактный класс, от которого наследуют balist1,balist2, методы и поля.

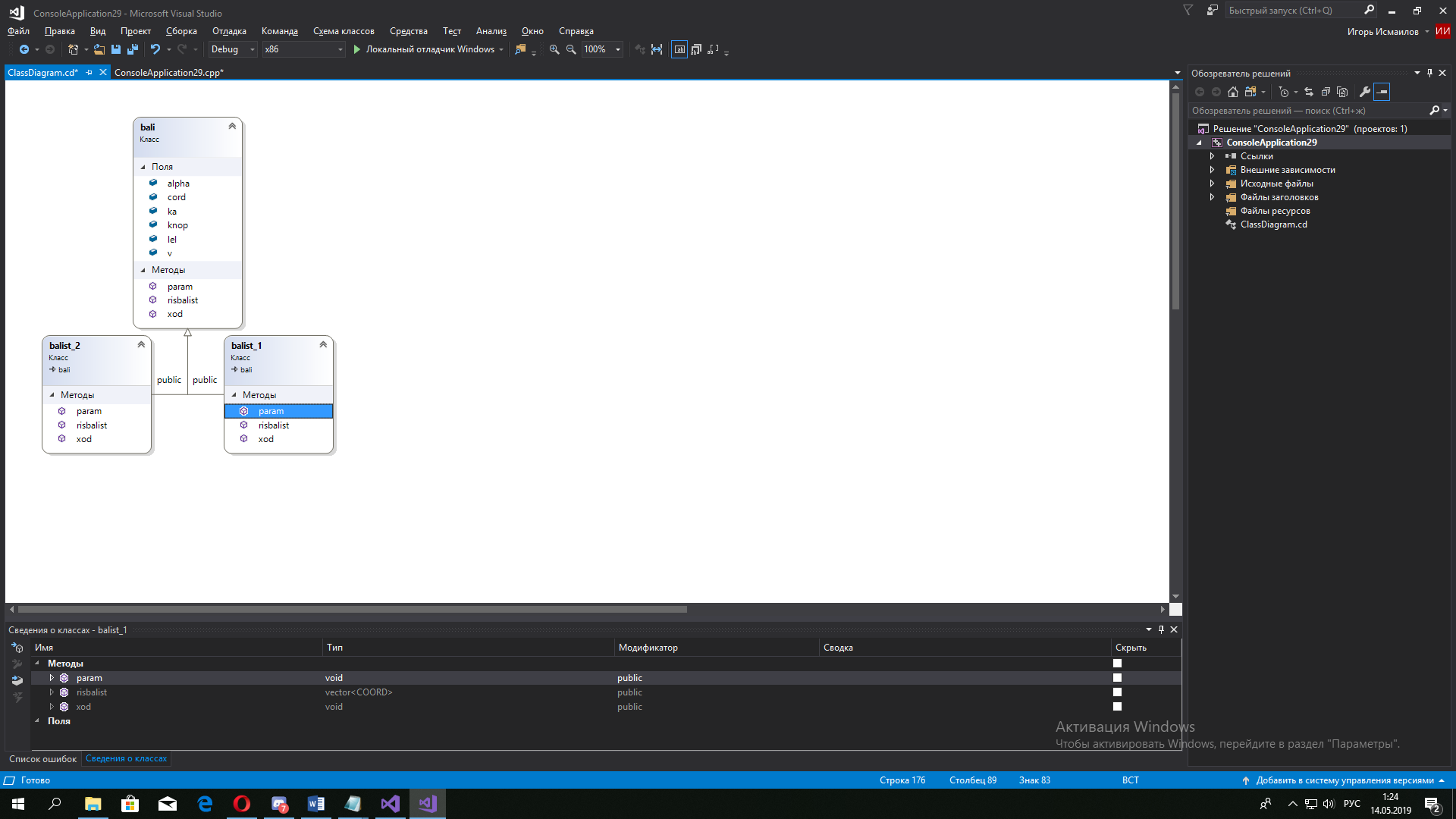


Рис 3.UML диаграмма класса balist1.

Класс имеющий метод xod, который отвечает на восприятие программой нажатия кнопок.Также имеет метод risbalist, который рисует баллисту и метод param, отвечающий за параметры при нажатии кнопок.

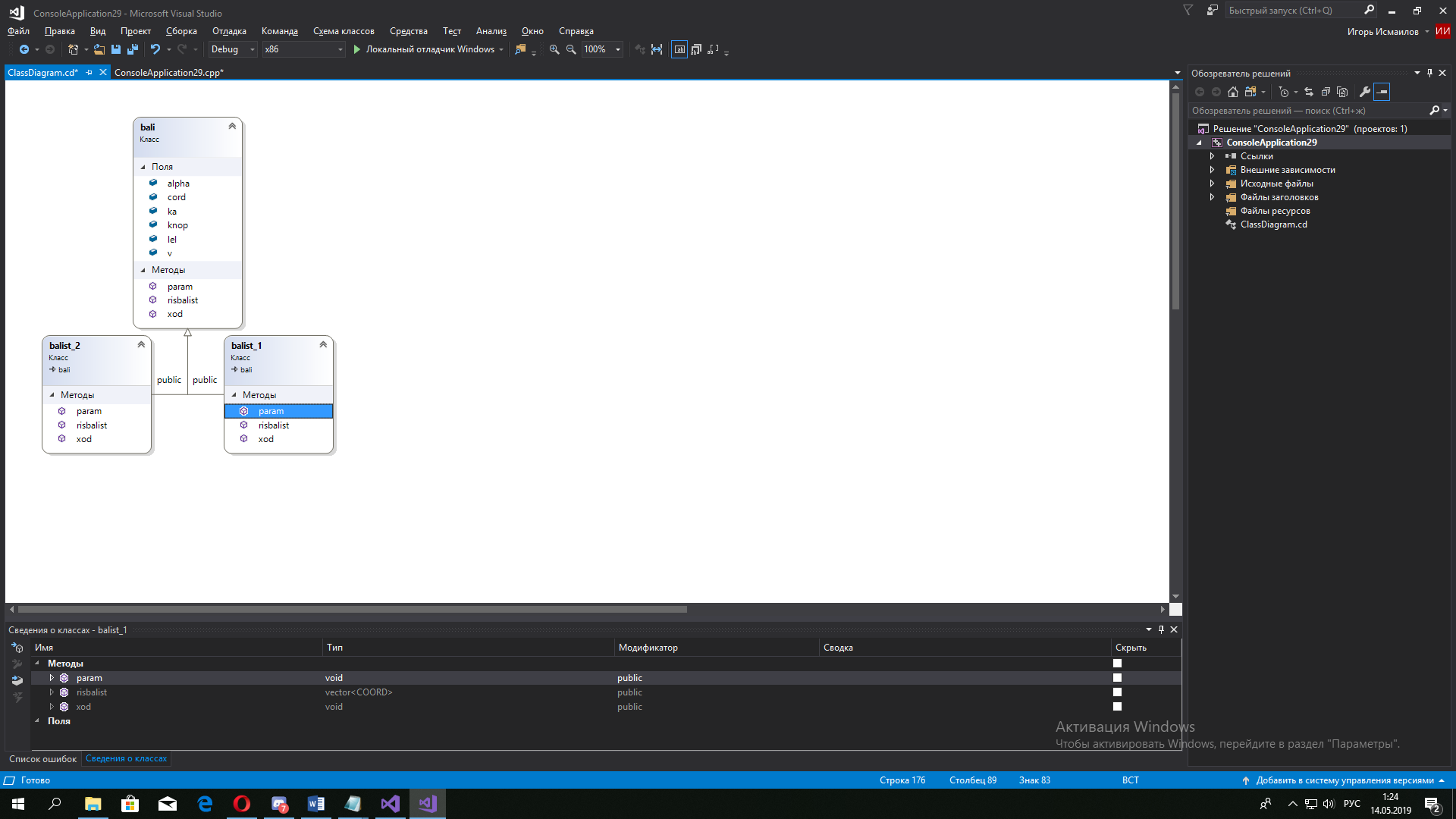


Рис 4.UML диаграмма класса balist2**.**

Класс имеющий отрицательное значение по сравнению с balist1.

**3.ТЕСТИРОВАНИЕ ПРОГРАММЫ**

Тестирование игровой программы производилось попытками сломать ее, вывести из строя,проверками углов наклона баллист,скорости камня.В нормальных условиях программа работает стабильно, поставленные задачи выполняются без ошибок.

**4.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**4.1 Процедура установки**

Установка игры осуществляется копированием игры в каталог на жестком диске компьютера.

**4.2 Процедура удаления**

Удаление игры может быть произведено путём удаления её с жёсткого диска компьютера

**4.3 Запуск программы**

Запуск программы осуществляется через exe файл или компилятором Visual Studio.

**4.4 Эксплуатация программой**

1)После запуска программы появится меню с кратким сводом информации об управлении игрой.

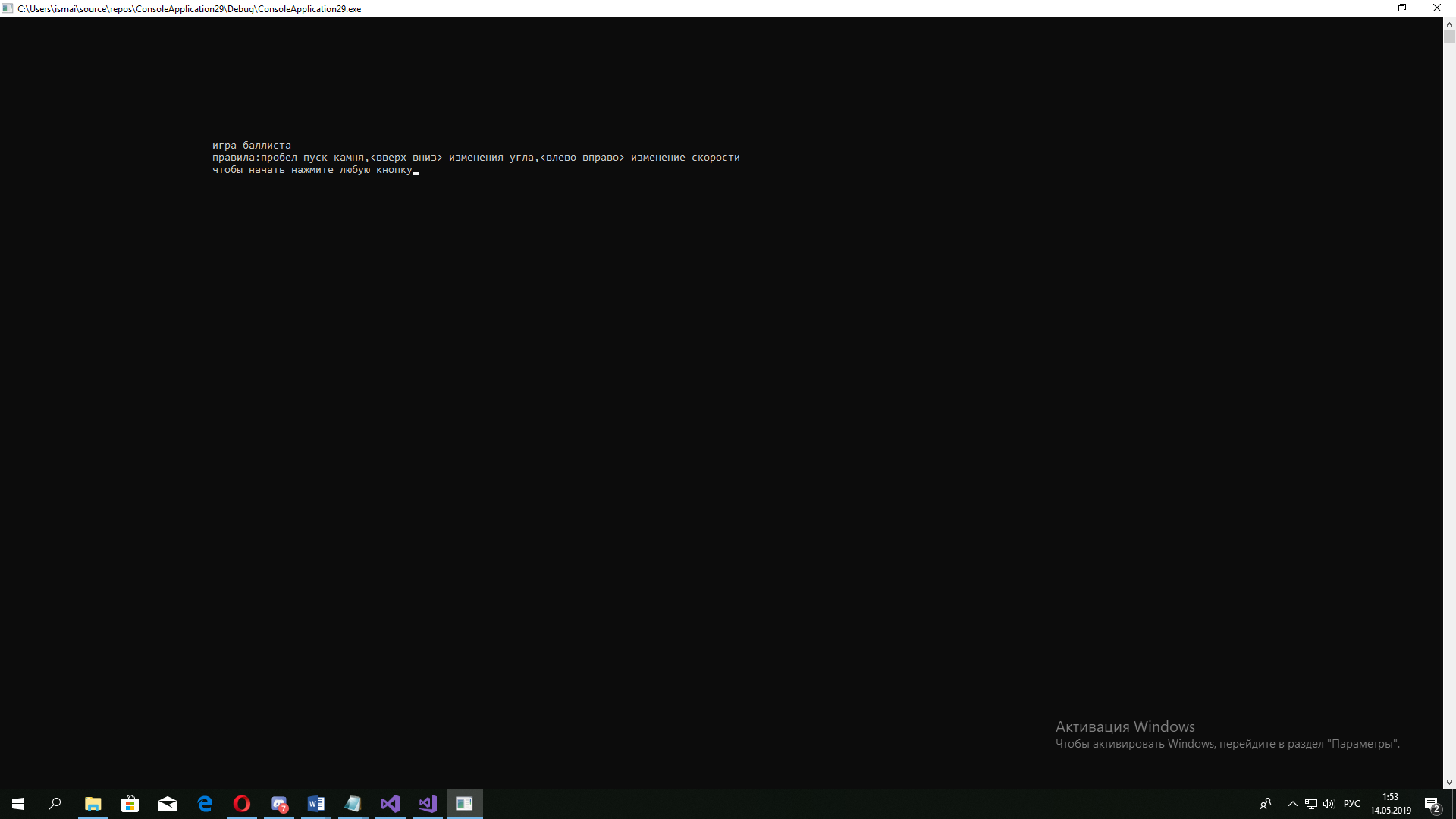


Рисунок 4.

4.5

2)После нажатия любой кнопки, пользователю будут показаны две баллисты, стоящие напротив друг друга. Каждый игрок ходит по очереди.

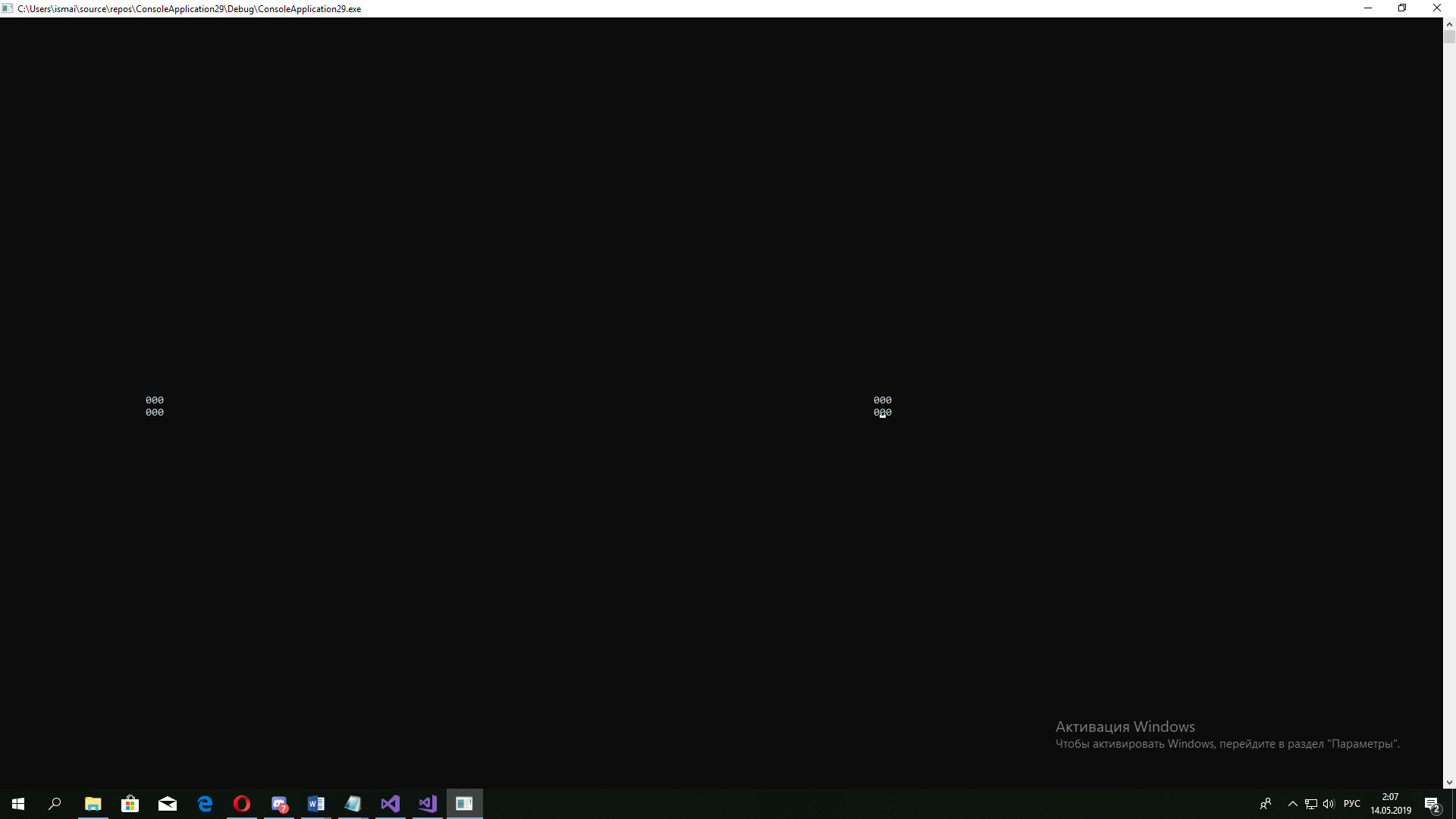


Рисунок 4.1

3)Управление осуществляется с помощью стрелочек на клавиатуре.

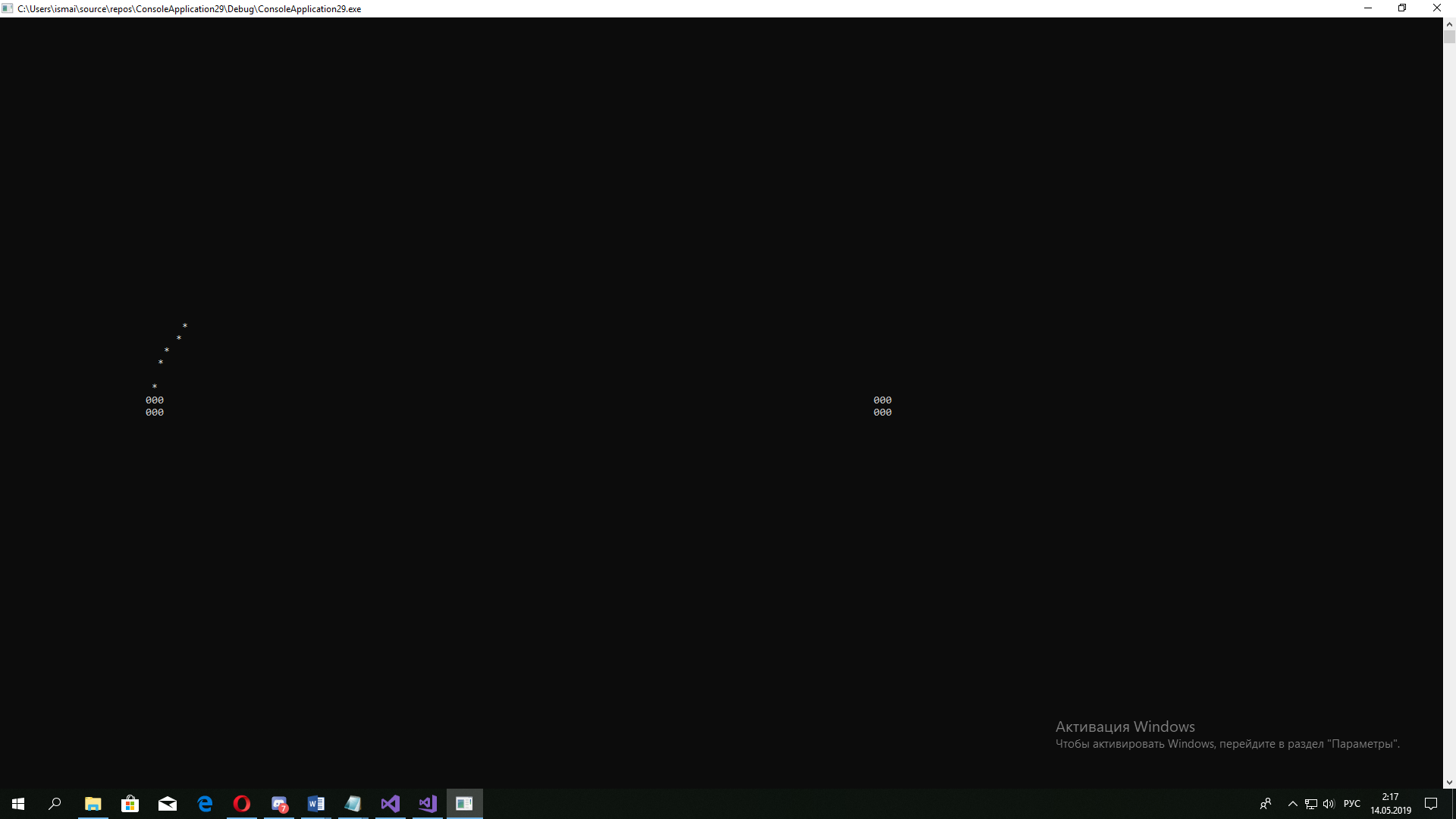


Рисунок 4.2

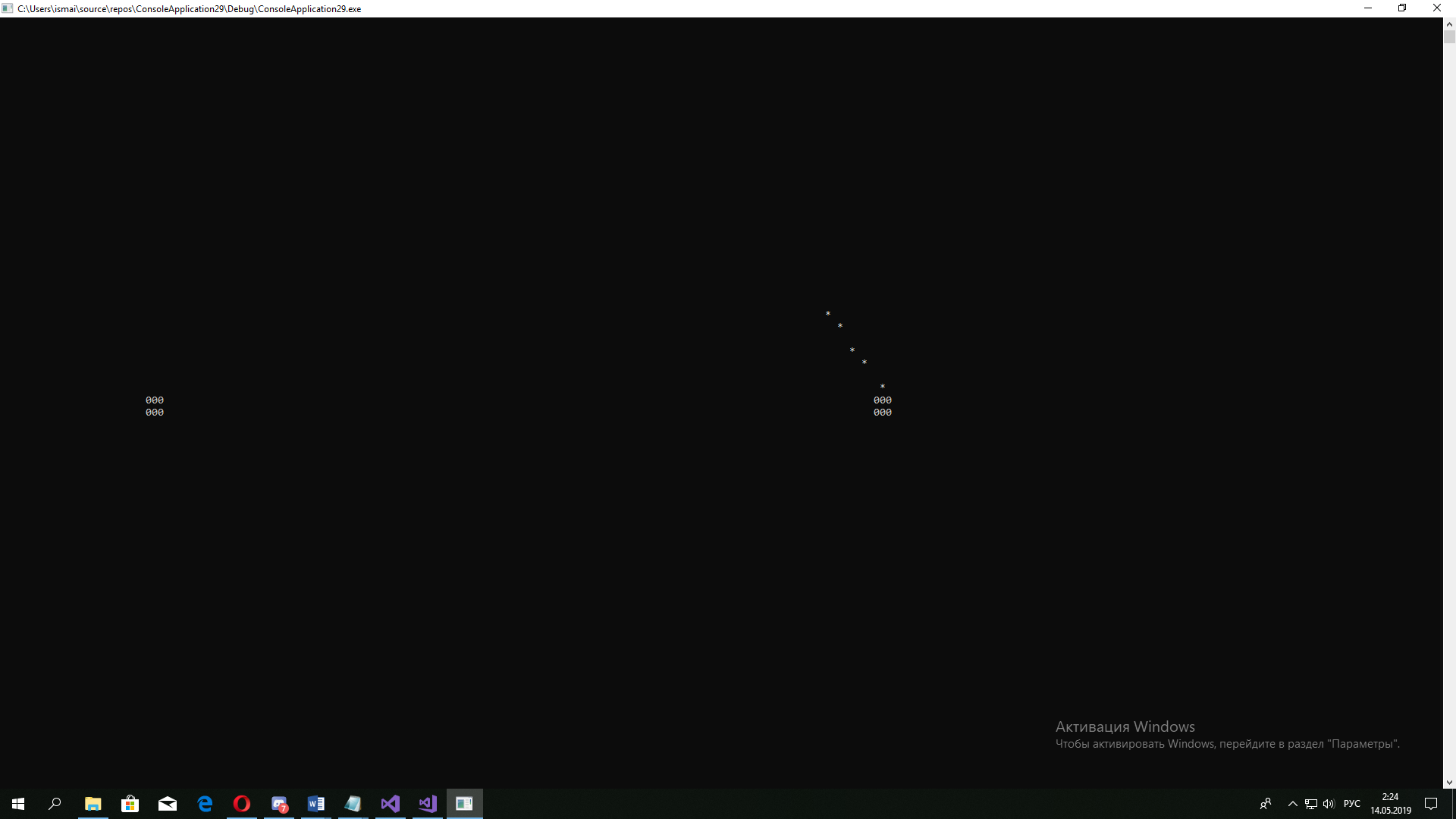
4)Для изменения угла выстрела, следует нажимать стрелочки вверх(для увеличения угла) или вниз(для уменьшения угла).Углы наклона от Пи/20 до 90 градусов. 

Рисунок 4.3

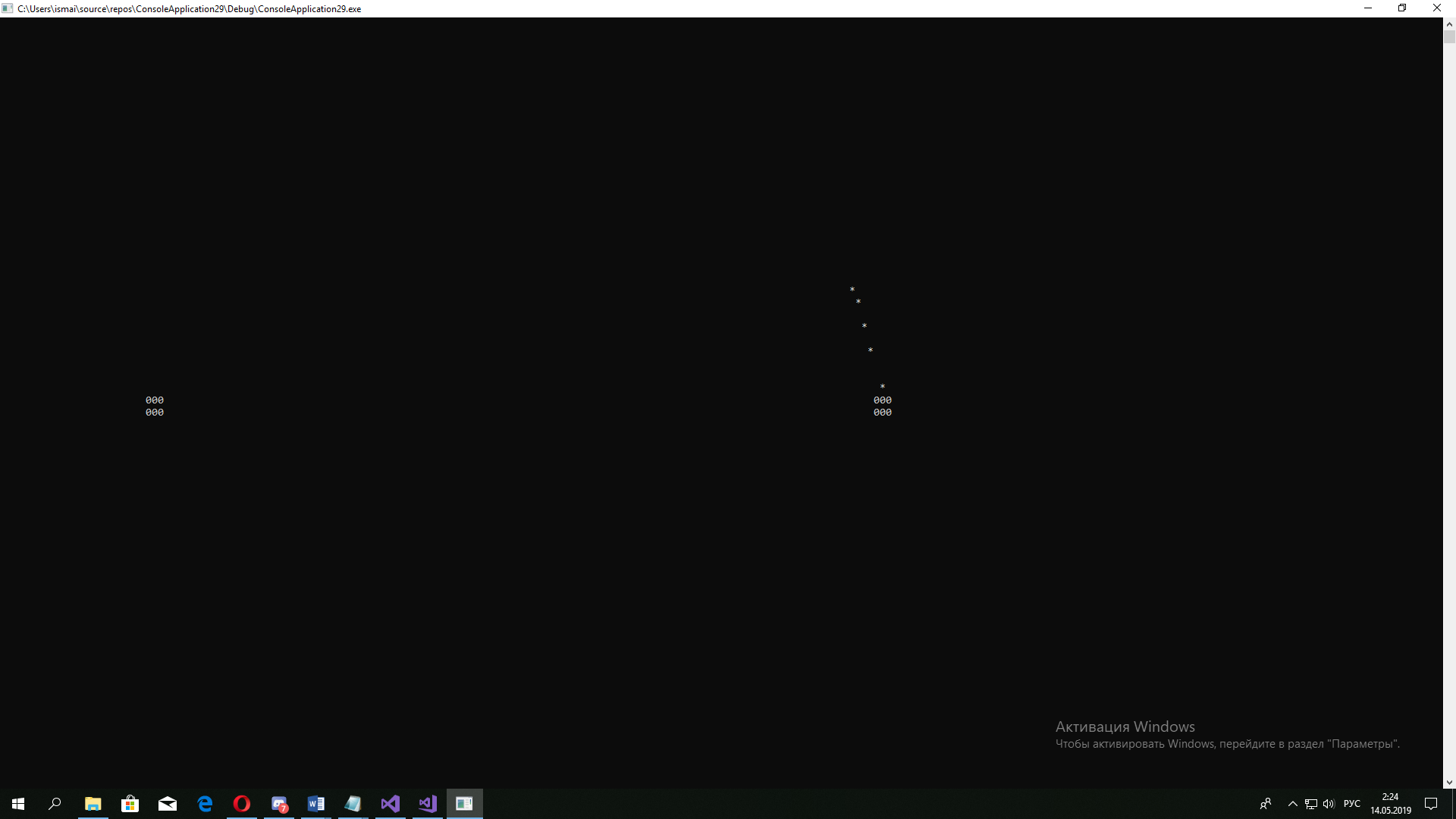


Рисунок 4.3.1

5)Выстрел осуществляется клавишей ”пробел”, после нажатия данной клавиши, вылетает камень из баллисты игрока, которого был ход.

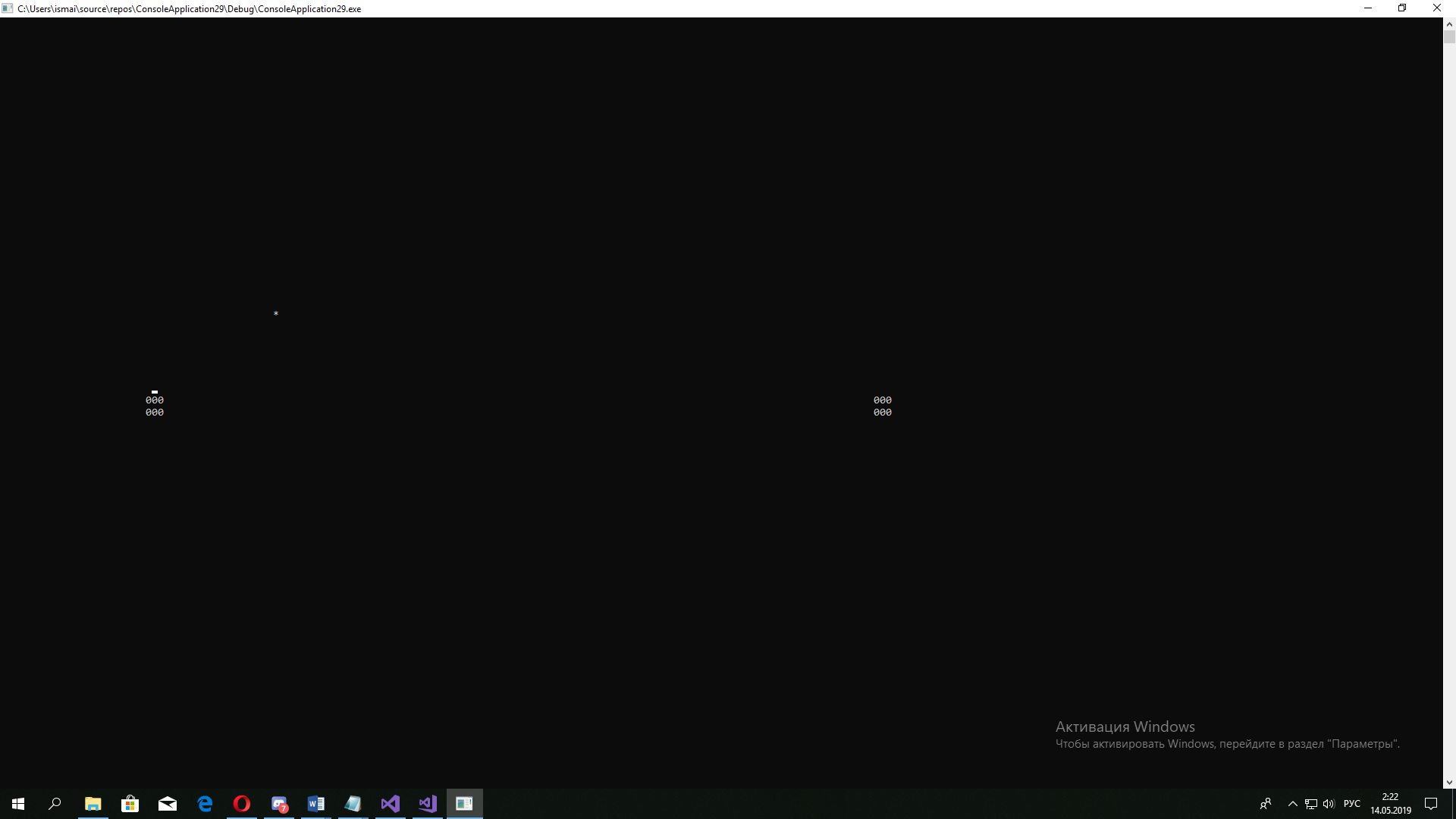


Рисунок 4.4

6)После неудачного выстрела, для передачи хода противнику, необходимо нажать на клавишу “пробел”.

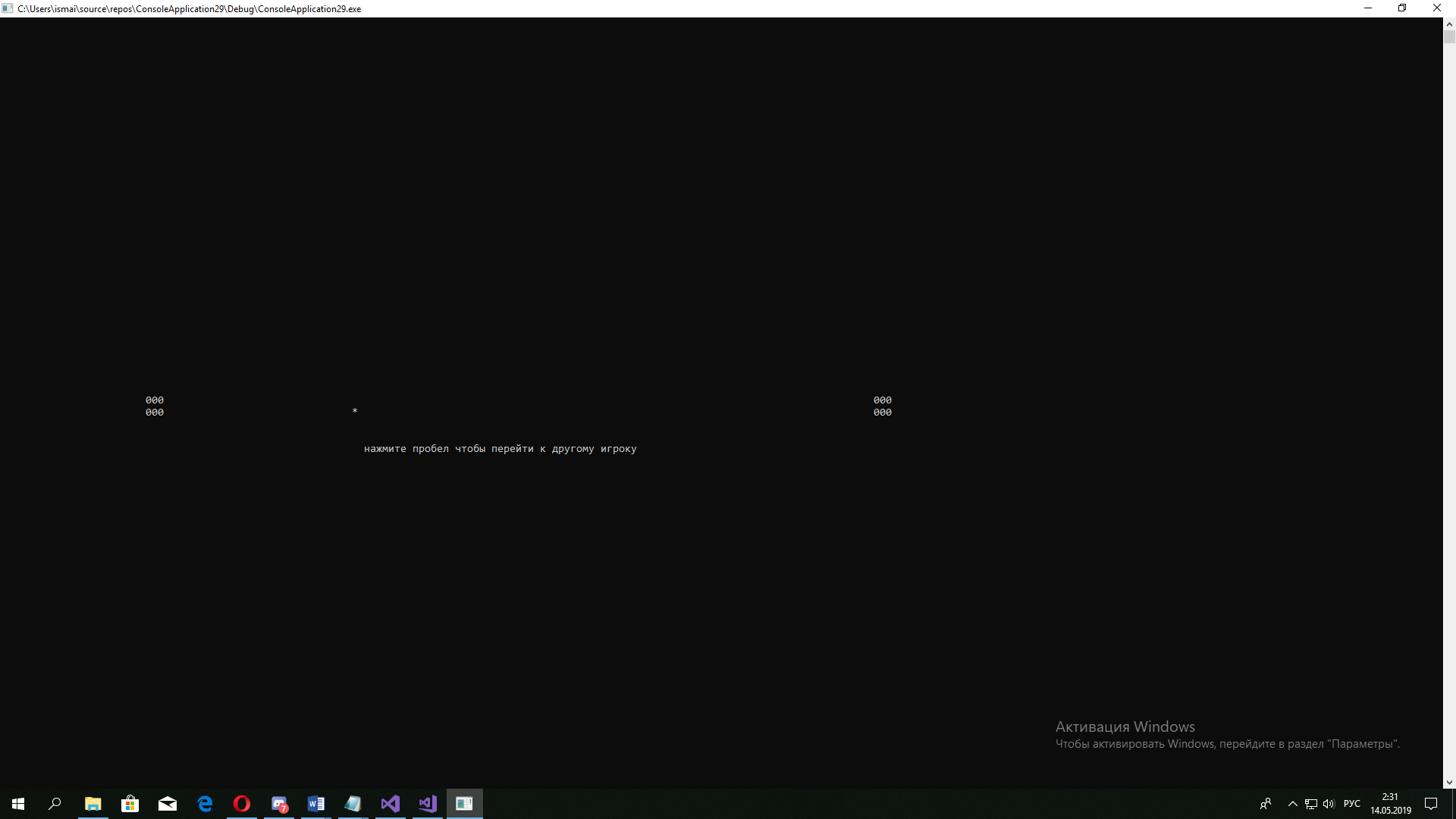


Рисунок 4.5

7)Игра продолжается до тех пор, пока один из игроков не попадет в баллисту соперника. После попадания, игра заканчивается.

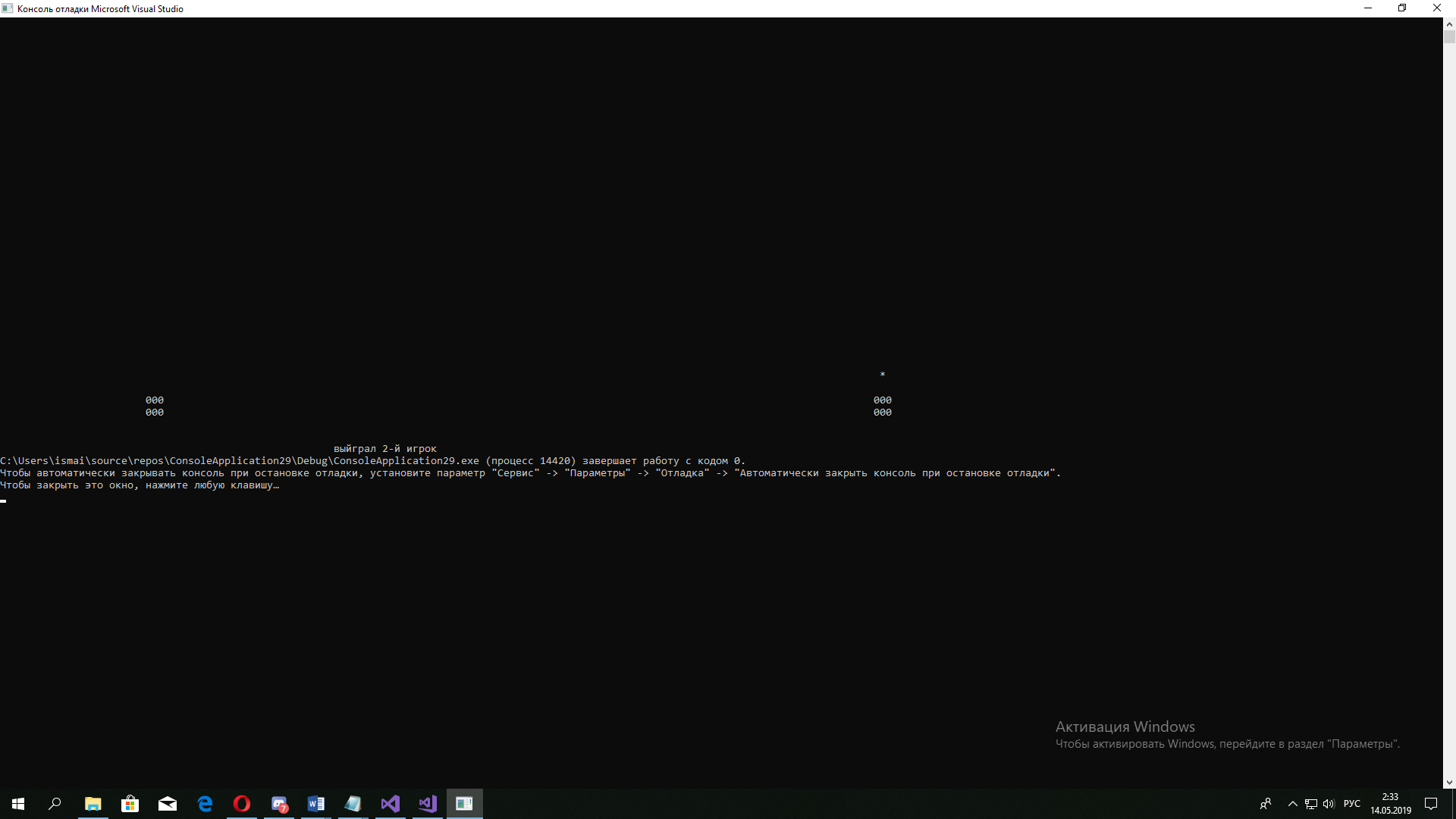


Рисунок 4.6

8)Для выхода из игры необходимо нажать на любую клавишу.

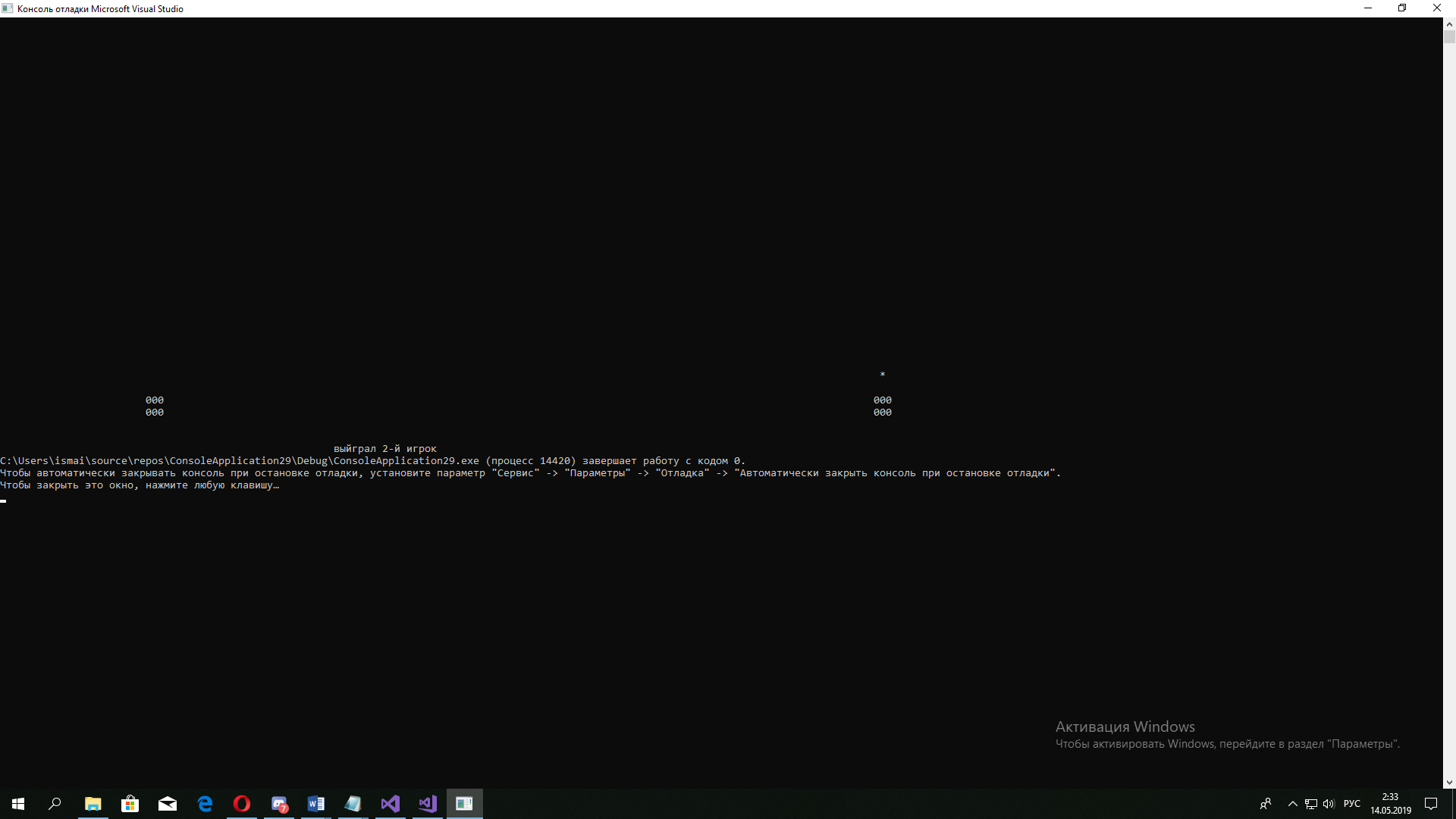


Рисунок 4.7

**Заключение**

В результате выполнения курсовой, мною была проделана большая работа по

объектно-ориентированное программированию.” Баллистическая игра” была полностью реализована на языке программирования C++. Написание программы повлияло на дополнительное изучение дополнительного материала и закреплению теоретических знаний на практике. При выполнении работы были изучены дополнительные ресурсы знаний.

#include "pch.h"

#include <iostream>

#include <Windows.h>

#include <cmath>

#include <conio.h>

#include <ctime>

#include <vector>

#define M\_PI 3.14159265

using namespace std;

HANDLE Console = GetStdHandle(STD\_OUTPUT\_HANDLE);//Управление смтрелочками

const int g = 9.81;

bool lose = false;

vector<COORD> b1(6);

vector<COORD> b2(6);

void napr(COORD cord, double alpha, int v, double vr, bool ka, bool num) {

double start = clock(), end = clock(), n = 0.1;

COORD prosh = { cord.X,30 };

COORD pervii = prosh;

if (vr == 15) {

for (double t = 0; t < vr;) {

SetConsoleCursorPosition(Console, pervii);

t = (end - start) / CLOCKS\_PER\_SEC;

end = clock();

if (num == 0)cord.X = (v \* cos(alpha)\*t) + 25;

else cord.X = (v \* cos(alpha)\*t) + 145;

cord.Y = -(-g \* t\*t / 2 + v \* sin(alpha)\*t) + 30;

if (t > n) {

if (cord.Y >= 34)break;

if (cord.Y >= 0 && cord.X >= 0 && cord.X <= 170) {

SetConsoleCursorPosition(Console, cord);

if (ka == 1) {

cout << "\*";

SetConsoleCursorPosition(Console, prosh);

cout << " ";

prosh = cord;

}

}

n += 0.1;

}

for (int i = 0; i < 6; i++) {//41-44 проверка попадания по другому игроку или по себе

if ((b1[i].X == cord.X&&b1[i].Y == cord.Y) || (b2[i].X == cord.X&&b2[i].Y == cord.Y))lose = true;

}

if (lose)break;

}

}

else {

for (double t = 0; t < vr; t += 0.1) {

if (num == 0)cord.X = (v \* cos(alpha)\*t) + 25;

else cord.X = (v \* cos(alpha)\*t) + 145;

cord.Y = -(-g \* t\*t / 2 + v \* sin(alpha)\*t) + 30;

if (cord.Y >= 34)break;

if (cord.Y >= 0 && cord.X >= 0 && cord.X <= 170) {

SetConsoleCursorPosition(Console, cord);

if (ka == 1)cout << "\*";

else cout << " ";

}

}

}

}

class bali {

public:

char knop;

int lel;

double alpha;

int v;

bool ka;

COORD cord;

virtual vector<COORD> risbalist(vector<COORD>b1) = 0;

virtual void xod() = 0;

virtual void param() = 0;

};

class balist\_1 :public bali {

public:

void param() {

napr(cord, alpha, v, 0.5, 0, 0);

knop = 1;

ka = 1;

}

vector<COORD> risbalist(vector <COORD> b1) {

for (int i = 0; i < 3; i++) {

b1[i] = this->cord;

SetConsoleCursorPosition(Console, cord);

cout << "0";

cord.Y = cord.Y + 1;

b1[i + 3] = this->cord;

SetConsoleCursorPosition(Console, cord);

cout << "0";

cord.X = cord.X - 1;

cord.Y = cord.Y - 1;

}

return b1;

}

void xod() {

v = 20;

lel = 10;

knop = 1;

alpha = M\_PI / lel;

ka = 0;

cord = { 25,30 };

while (knop != 32) {

knop = \_getch();

switch (knop) {

case 80:

if (lel < 20) {

param();

lel++;

}

break;

case 72:

if (lel > 2) {

param();

lel--;

}

break;

case 75:

if (v > 7) {

param();

v--;

}

break;

case 77:

if (v < 50) {

param();

v++;

}

break;

case 32:

napr(cord, alpha, v, 0.5, 0, 0);

napr(cord, alpha, v, 15, 1, 0);

SetConsoleCursorPosition(Console, { 50,35 });

if (!lose) {

cout << "нажмите пробел чтобы перейти к другому игроку" << endl;

knop = \_getch();

while (knop != 32) {

knop = \_getch();

}

SetConsoleCursorPosition(Console, { 50,35 });

cout << " " << endl;

napr(cord, alpha, v, 14.99, 0, 0);

}

break;

}

alpha = M\_PI / lel;

if (ka == 1) {

napr(cord, alpha, v, 0.5, 1, 0);

ka = 0;

}

}

}

};

class balist\_2 :public bali {

public:

void param() {

napr(cord, alpha, v, 0.5, 0, 1);

knop = 1;

ka = 1;

}

vector<COORD> risbalist(vector <COORD> b1) {

for (int i = 0; i < 3; i++) {

b1[i] = this->cord;

SetConsoleCursorPosition(Console, cord);

cout << "0";

cord.Y = cord.Y + 1;

b1[i + 3] = this->cord;

SetConsoleCursorPosition(Console, cord);

cout << "0";

cord.X = cord.X - 1;

cord.Y = cord.Y - 1;

}

return b1;

}

void xod() {

v = 20;

knop = 0;

lel = 10;//на что делаить пи на угол который должен быть для понятия компьютером

alpha = M\_PI - (M\_PI / lel);

ka = 0;

cord = { 145,30 };

while (knop != 32) {

knop = \_getch();

switch (knop) {

case 80:

if (lel < 20) {

param();

lel++;

}

break;

case 72:

if (lel > 2) {

param();

lel--;

}

break;

case 75:

if (v < 50) {

param();

v++;

}

break;

case 77:

if (v > 7) {

param();

v--;

}

break;

case 32:

napr(cord, alpha, v, 0.5, 0, 1);

napr(cord, alpha, v, 15, 1, 1);

if (!lose) {

SetConsoleCursorPosition(Console, { 60,35 });

cout << "нажмите пробел чтобы перейти к другому игроку" << endl;

knop = \_getch();

while (knop != 32) {

knop = \_getch();

}

SetConsoleCursorPosition(Console, { 60,35 });

cout << " " << endl;

napr(cord, alpha, v, 14.99, 0, 1);

}

break;

}

alpha = M\_PI - (M\_PI / lel);

if (ka == 1) {

napr(cord, alpha, v, 0.5, 1, 1);

ka = 0;

}

}

}

};

int main()

{

HWND hwnd = GetForegroundWindow();

ShowWindow(hwnd, SW\_MAXIMIZE);

setlocale(0, "rus");

COORD stroch = { 35,10 };

SetConsoleCursorPosition(Console, stroch);

cout << "игра баллиста";

SetConsoleCursorPosition(Console, { stroch.X,stroch.Y + 1 });

cout << "правила:пробел-пуск камня,<вверх-вниз>-изменения угла,<влево-вправо>-изменение скорости";

SetConsoleCursorPosition(Console, { stroch.X,stroch.Y + 2 });

cout << "чтобы начать нажмите любую кнопку";

\_getch();

system("cls");

balist\_1 pervii;

balist\_2 vtoroi;

bool ka = 1;

pervii.cord = { 26,31 };

b1 = pervii.risbalist(b1);

vtoroi.cord = { 146,31 };

b2 = vtoroi.risbalist(b2);

while (true) {

pervii.xod();

if (lose == true) {

SetConsoleCursorPosition(Console, { 55,35 });

if (pervii.lel > 2)cout << "выйграл 1-й игрок";

else cout << "выйграл 2-й игрок";

break;

}

vtoroi.xod();

if (lose == true) {

SetConsoleCursorPosition(Console, { 55,35 });

if (pervii.lel > 2)cout << "выйграл 2-й игрок";

else cout << "выйграл 1-й игрок";

break;

}

}

}