

Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
по курсу «Программирование»
на тему «Разработка программы сложной структуры
методом нисходящего программирования»

Выполнил: Баженов Д.С.

Студент группы 21В01

Принял:

Зав кафедры ВТ Митрохин М.А.

Пенза, 2022

Содержание

Введение	3
1. Постановка задачи	3
2. Выбор решения.....	4
3. Описание разработки программы.....	5
4. Отладка и тестирование	5
5. Описание программы	5
5.1 Описание первой формы.....	5
5.2Описаниевторой формы	8
6. Руководство пользователя	29
Заключение	35
Список используемых источников.....	35
Приложение А. Листинг программы.....	36
Приложение В. Файлы	50

Введение

Сегодня большое влияние на все сферы общественной жизни оказывают информатизация и информационные технологии. Все, кто имеет дело с компьютером, так или иначе сталкивались с компьютерными играми, и подавляющее большинство может сходу назвать несколько игр, которые им особенно понравились. Те, кто уже совсем наигрался, почти наигрался или еще не наигрался, но в процессе общения с компьютером уже начал совмещать игры с чем-нибудь более полезным, возможно, хотели бы придумать какие-нибудь свои, не похожие ни на какие другие игры (1).

Многое захватывает в таком творчестве. И не сам процесс игры, а разработка игровой вселенной, ее проектирование и реализация. Когда можно слить воедино сценарий, графику, музыку, искусно задуманный и умело запрограммированный алгоритм — создать единый фантастический мир, живущий по законам, которые ты же для него и придумал.

Данная работа, посвященная написанию программного приложения «GalaWar», поспособствует изучению одной из таких программ.

1. Постановка задачи

Необходимо разработать программу «Игра GalaWar», которая должна отвечать следующим требованиям:

Многомодульность программы. Программа должна быть поделена на логические модули, что позволяет быстрее проводить отладку и тестирование, находить возможные ошибки и расширять функционал.

Использование сложных типов данных – списков, структур, файлов. Использование списка и структур необходимо для упрощения работы с данными в коде. Использование файлов необходимо для сохранения базы данных и дальнейшего её открытия.

Режим работы видеосистемы – текстовый/графический. Сначала нужно определиться с типом интерфейса и элементами управления, затем необходимо изучить способы их реализации.

Устройство ввода информации – клавиатура, мышь. Необходимо научиться анализировать события, возникающие при работе с клавиатурой или мышью. Необходимо однозначно идентифицировать и выполнять те или иные действия в зависимости от действий пользователя, это облегчит использование программы.

Интерфейс должен быть построен на основе меню и панели инструментов. Необходимо изучить приемы разработки меню для создания интуитивно понятного пользовательского интерфейса.

Наличие заставки. Пользователю, запустившему программу, необходима заставка, дающая сведения о базе данных и ее возможности перехода к работе.

2. Выбор решения

При запуске программы открывается меню с 4 кнопками:

- 1) Начало игры. При нажатии появляется окошко с самой игрой.
- 2) Показать рекорды. При нажатии появляется окошко с рекордами игроков.
- 3) Вывод информации об авторе. При нажатии появляется окошко с информацией об авторе.
- 4) Выход из игры.

При завершении волны врагов открывается меню с 1 кнопкой и 6 полями:

- 1) Поле «количество здоровья». Величина прямо влияющая на количество здоровья главного корабля. Изменяется при нажатии стрелок, находящихся рядом с полем.
- 2) Поле «количество щита». Величина прямо влияющая на количество щита главного корабля. Изменяется при нажатии стрелок, находящихся рядом с полем.
- 3) Поле «урон». Величина прямо влияющая на количество урона главного корабля. Изменяется при нажатии стрелок, находящихся рядом с полем.
- 4) Поле «радиус атаки». Величина прямо влияющая на радиус атаки главного корабля. Изменяется при нажатии стрелок, находящихся рядом с полем.
- 5) Поле «количество регенерируемого щита». Величина прямо влияющая на количество регенерируемого щита главного корабля. Изменяется при нажатии стрелок, находящихся рядом с полем.
- 6) Поле «скорость движения». Величина прямо влияющая на скорость движения главного корабля. Изменяется при нажатии стрелок, находящихся рядом с полем.
- 7) Кнопка «Подтвердить». Подтверждает выбор пользователя и проверяет соответствует ли он условиям игры.

При нажатии клавиши «escape» появляется меню с 2 кнопками:

- 1) Кнопка «Продолжить». Позволяет вернуться в игру.
- 2) Кнопка «Вернуться в главное меню». Позволяет вернуться в меню, которое появляется при запуске программы.

При уничтожении главного корабля появляется меню с 2 кнопками:

- 1) Кнопка «Ввод». Позволяет ввести имя игрока для последующего сохранения.
- 2) Кнопка «Вернуться в главное меню». Позволяет вернуться в меню, которое появляется при запуске программы.

3. Описание разработки программы

Разработанная программа состоит из нескольких модулей:

- 1) MyForm — включает функции вывода рекордов, информации об авторе, начала игры и выхода из игры.
- 2) MyForm1 — включает функции, необходимые для самой игры (функции движения главного корабля, движения врагов, выстрелов главного корабля, выстрелов врагов, создания волны врагов, обработка «навыков» главного корабля), «Паузы игры», сохранения рекорда при окончании игры, выхода в меню, которое появляется при запуске программы.

4. Отладка и тестирование

В качестве среды разработки была выбрана программа Visual Studio 2022.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы

5. Описание программы

5.1 Описание первой формы

Обработчик события `button1_click` начинается при щелчке на кнопку «Начало игры» и предназначена для вывода на экран второй формы.

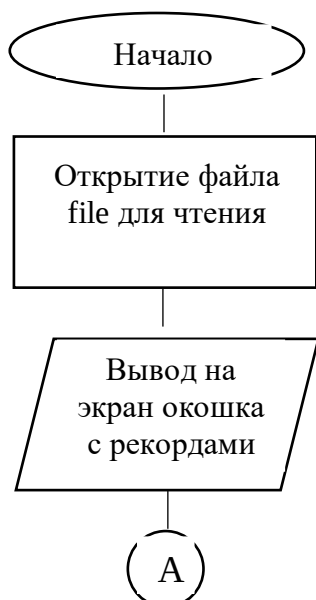
Обработчик события `button2_click` начинается при щелчке на кнопку «Показать рекорды» и предназначено для чтения из файла информации об игроке и последующим выводе ее на экран.

Обработчик события `ToolStripMenuItem_Click` начинается при щелчке на пункт меню «Вывод информации об авторе» и предназначено для вывода информации об авторе игры.

Обработчик события `button3_click` начинается при щелчке на кнопку «Выход из игры» и предназначена для выхода из игры.



Рисунок 1 – Блок-схема обработчика события `button1_click`



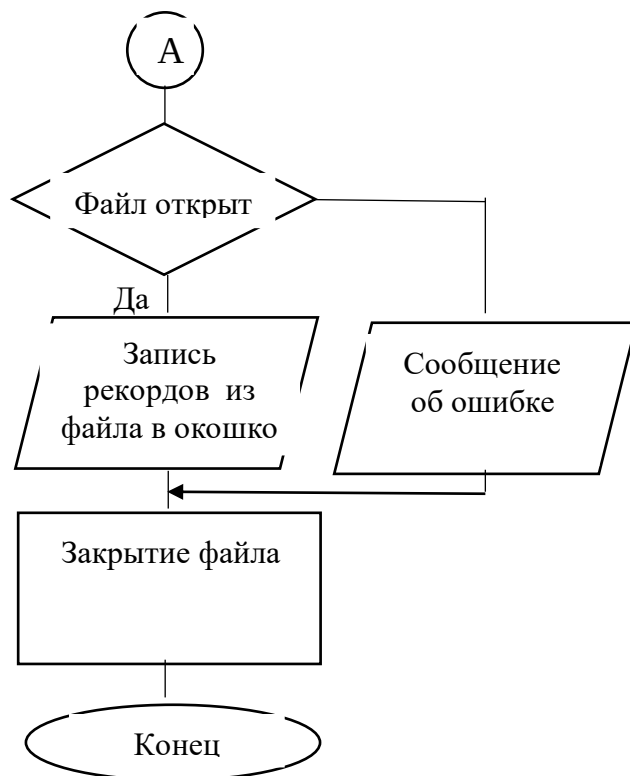
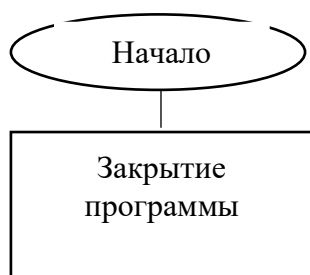


Рисунок 2 – Блок-схема обработчика событияbutton2_click



Рисунок 3 – Блок-схема обработчика событияbutton4_click



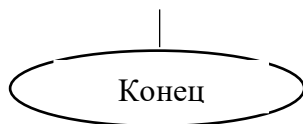


Рисунок 4 – Блок-схема обработчика события button3_click

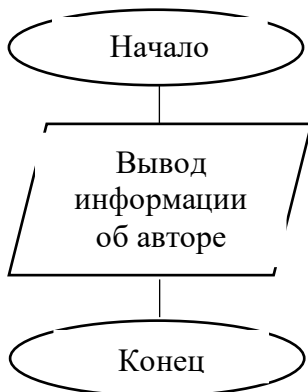


Рисунок 5 – Блок-схема обработчика события ToolStripMenuItem_Click

5.2 Описание второй формы

Обработчик события MyForm1_Load предназначено для присвоения стандартных характеристик главному кораблю.

Обработчик события button1_Click начинается при щелчке на кнопку «Подтвердить» и предназначено для подтверждения выбора очков характеристик пользователем, учитывая условия игры.

Обработчик события button2_Click начинается при щелчке на кнопку «Продолжить» и предназначено для продолжении игры.

Обработчик события button3_Click начинается при щелчке на кнопку «Вернуться в главное меню» и предназначено для выхода из второй формы.

Обработчик события button4_Click начинается при щелчке на кнопку «Ввод» и предназначено для данных в файл.

Обработчик события button5_Click начинается при щелчке на кнопку «Вернуться в главное меню» и предназначено для закрытия второй формы при уничтожении главного корабля.

Обработчик события timer1_Tick предназначено для передвижения главного корабля в след за курсором.

Обработчик события timer2_Tick предназначено для запуска волн и задания характеристик каждого врага в волне.

Обработчик события timer3_Tick предназначено для остановки игры на время и вывода на экран окна с характеристиками персонажа.

Обработчик события timer7_Tick предназначено для выбора цели атаки главного корабля.

Обработчик события timer9_Tick предназначено для регенерации щита врагов и главного корабля.

Обработчик события timer8_Tick предназначено для вывода информации о здоровье и щите врагов и главного корабля.

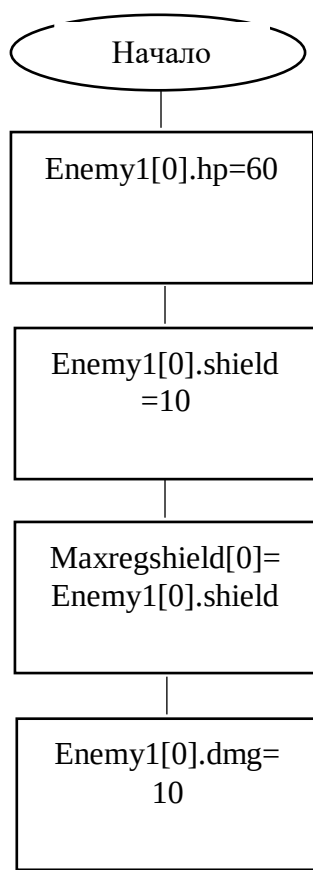
Обработчик события timer5_Tick предназначено для создания анимации выстрела врага, и просчета последствий попадания.

Обработчик события timer6_Tick предназначено для создания анимации выстрела главного корабля, и просчета последствий попадания.

Обработчик события timer6_Tick предназначено для приостановки игры на произвольное пользователю время.

Обработчик события timer12_Tick предназначено для использования навыков главного корабля.

Обработчик события **обработчика события** timer4_Tick предназначено для передвижения врагов в след за главным кораблём.



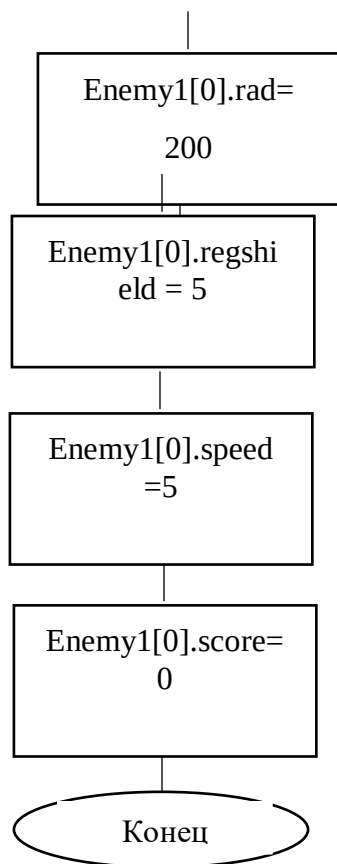


Рисунок 6 – Блок-схема обработчика события MyForm1_Load

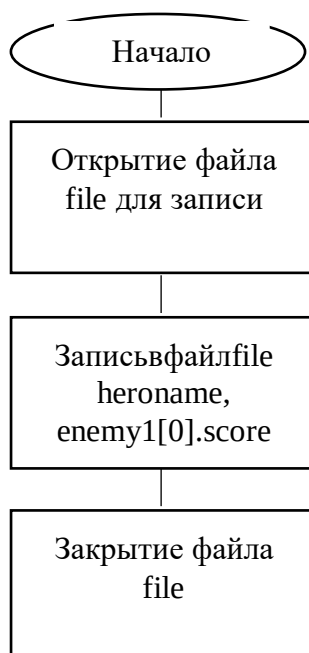




Рисунок 7 – Блок-схема обработчика событияbutton4_Click

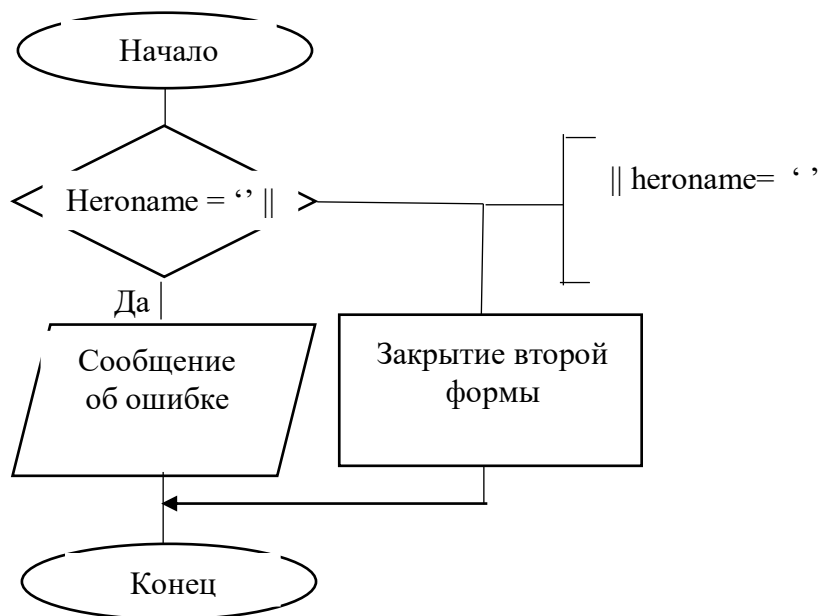


Рисунок 8– Блок-схема обработчика событияbutton5_Click

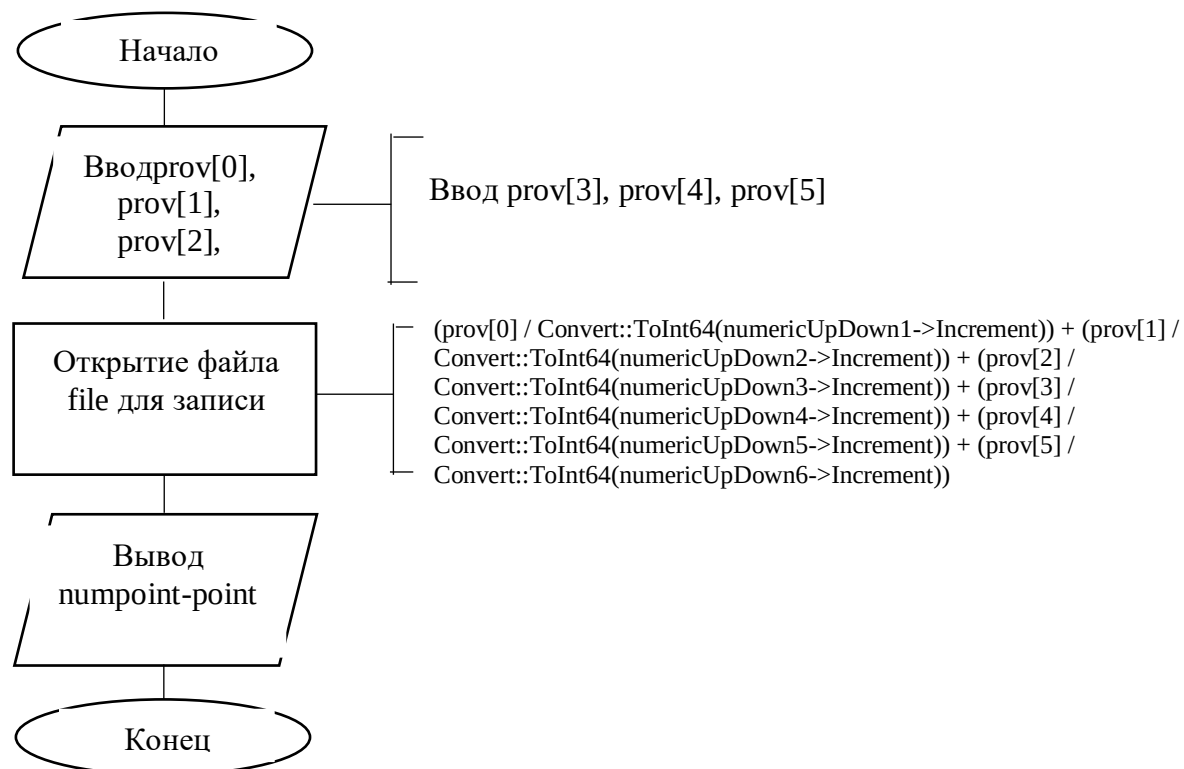


Рисунок 9– Блок-схема обработчика событияtimer10_Tick

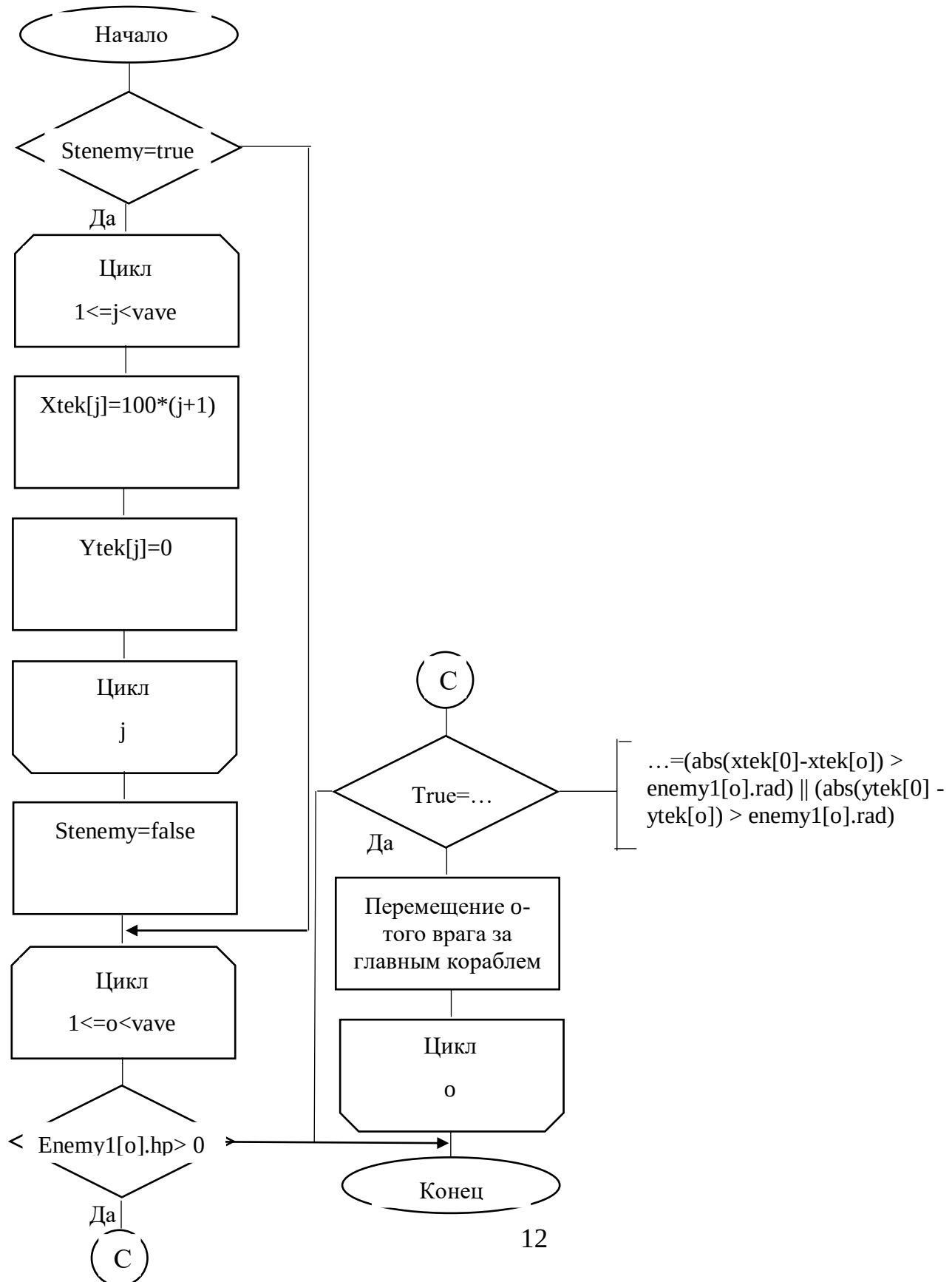


Рисунок 10– Блок-схема обработчика событияtimer4_Tick

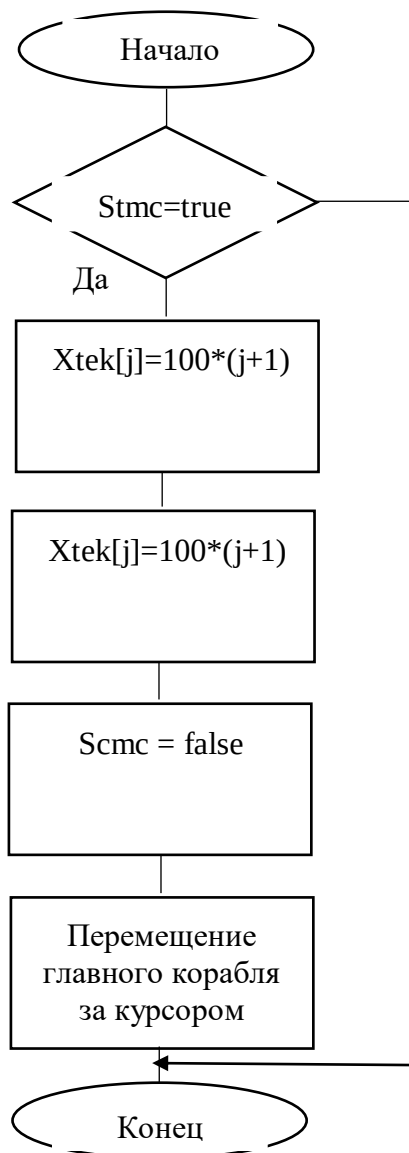
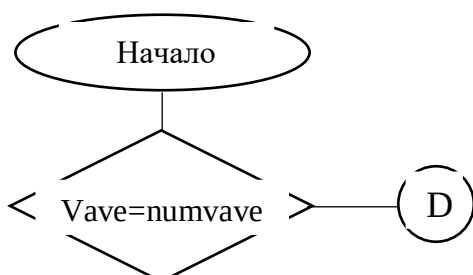


Рисунок 11 – Блок-схема обработчика событияtimer1_Tick



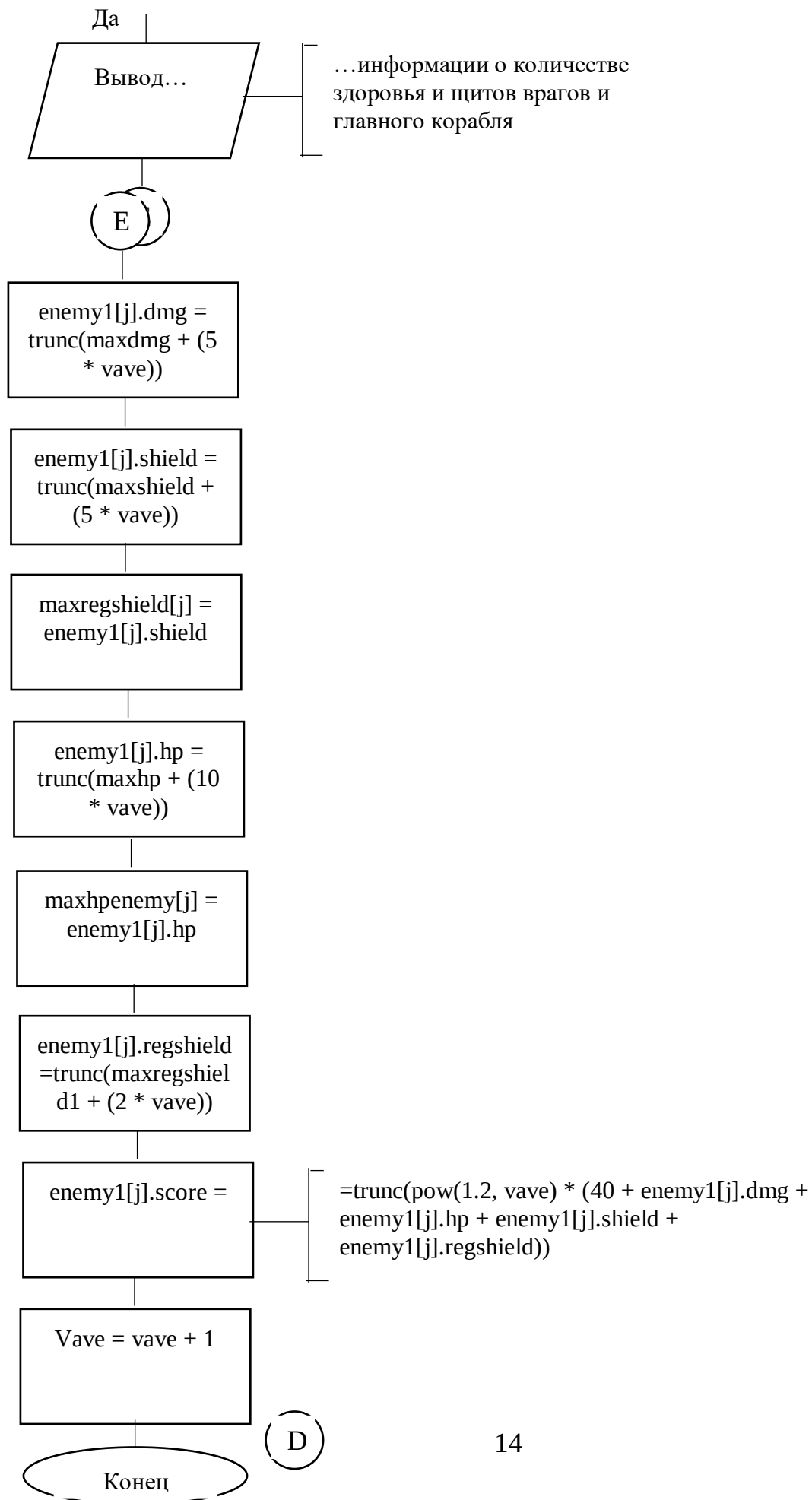




Рисунок 12– Блок-схема обработчика событияtimer2_Tick

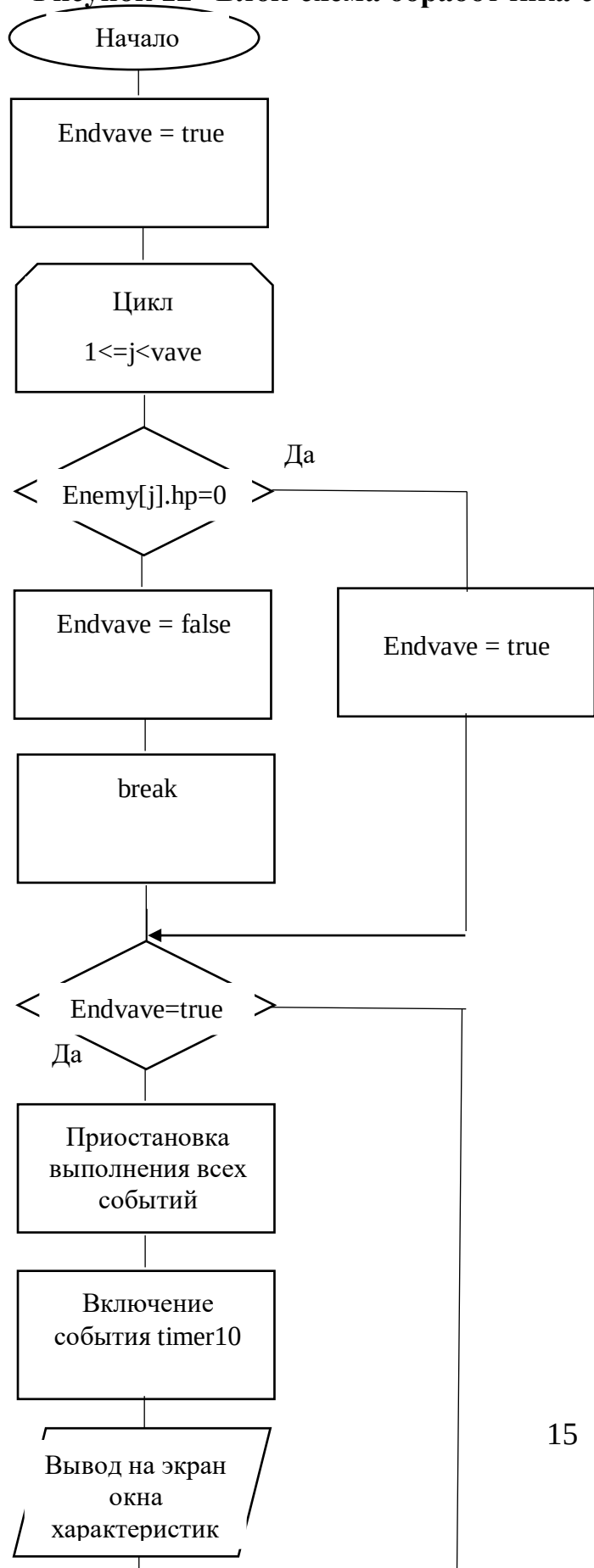
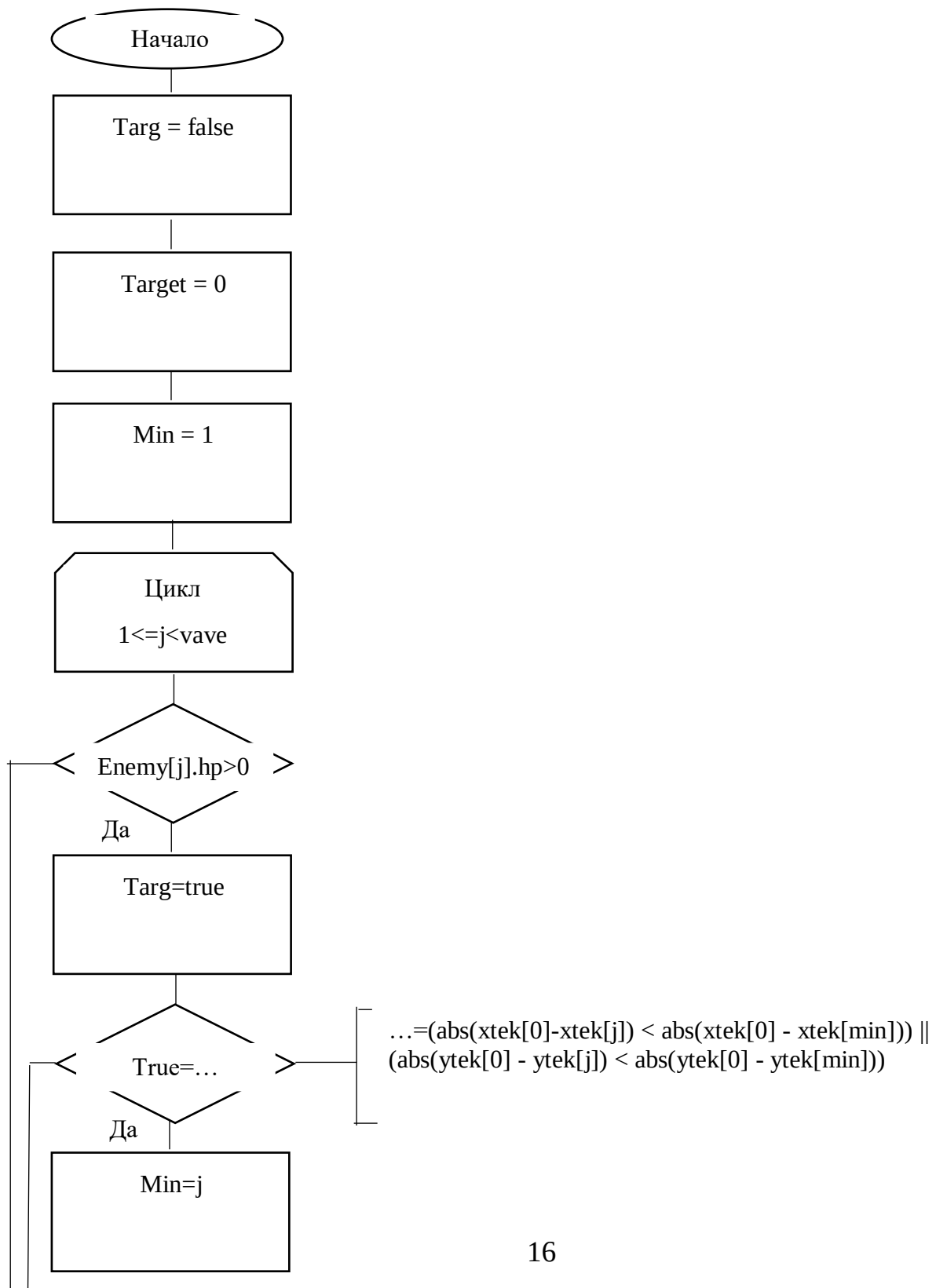


Рисунок 13– Блок-схема обработчика события timer3_Tick



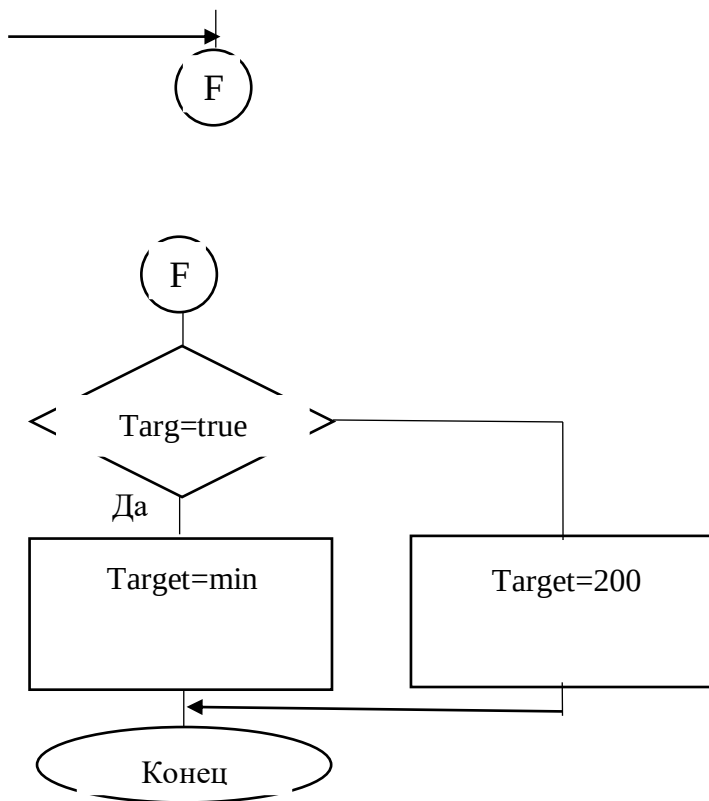
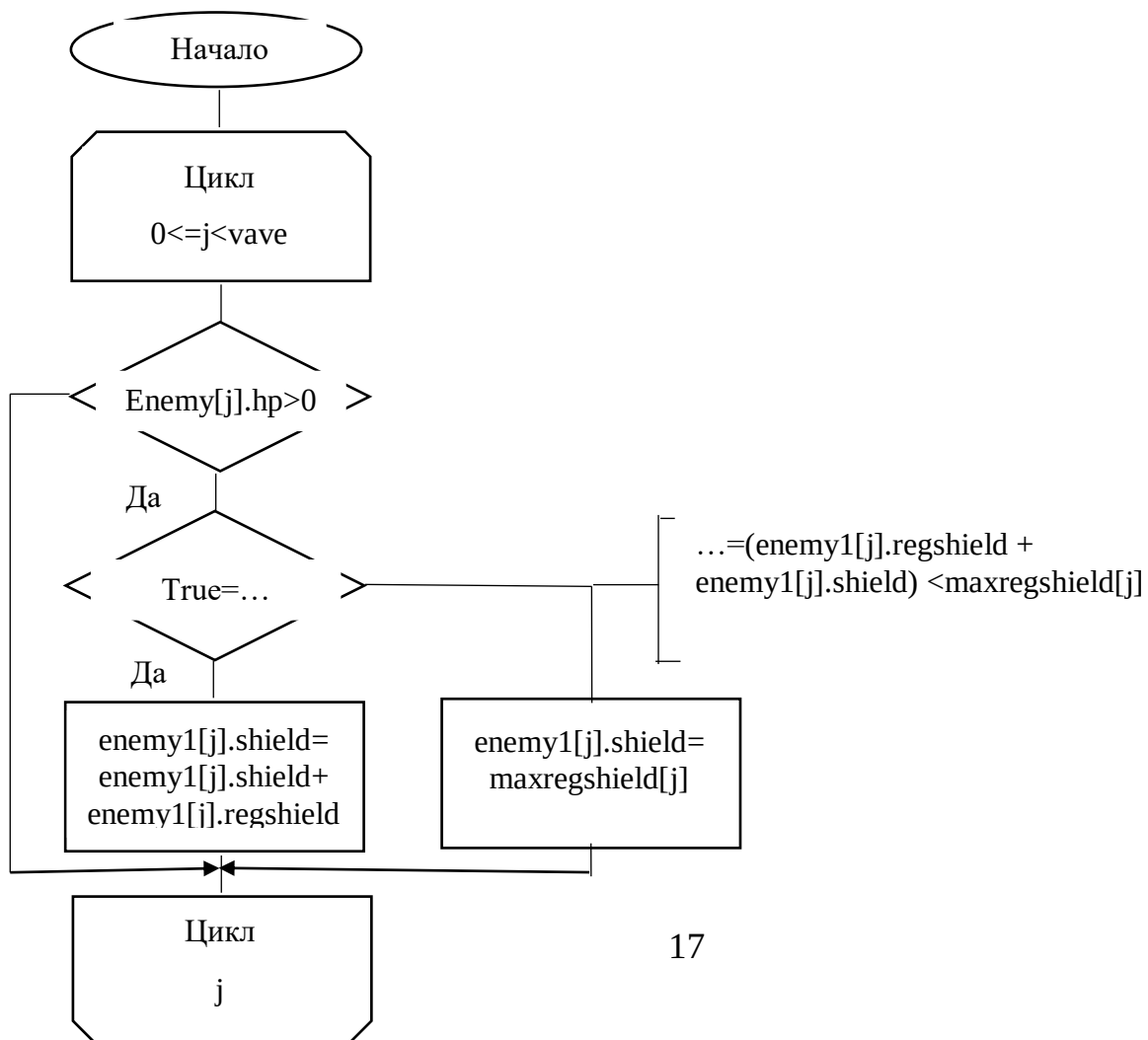


Рисунок 14– Блок-схема обработчика события timer7_Tick



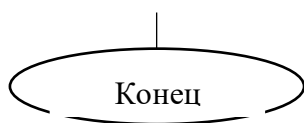
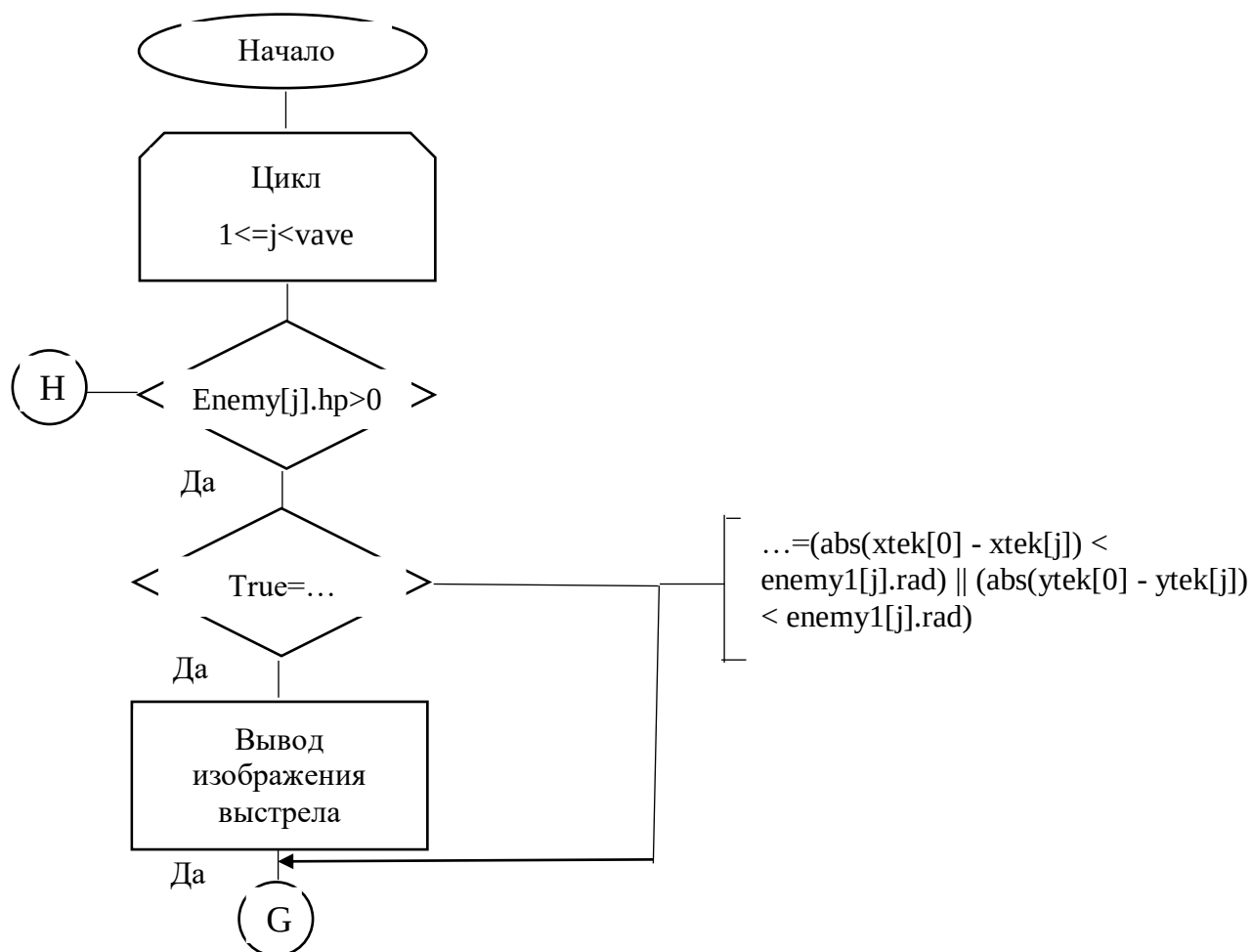
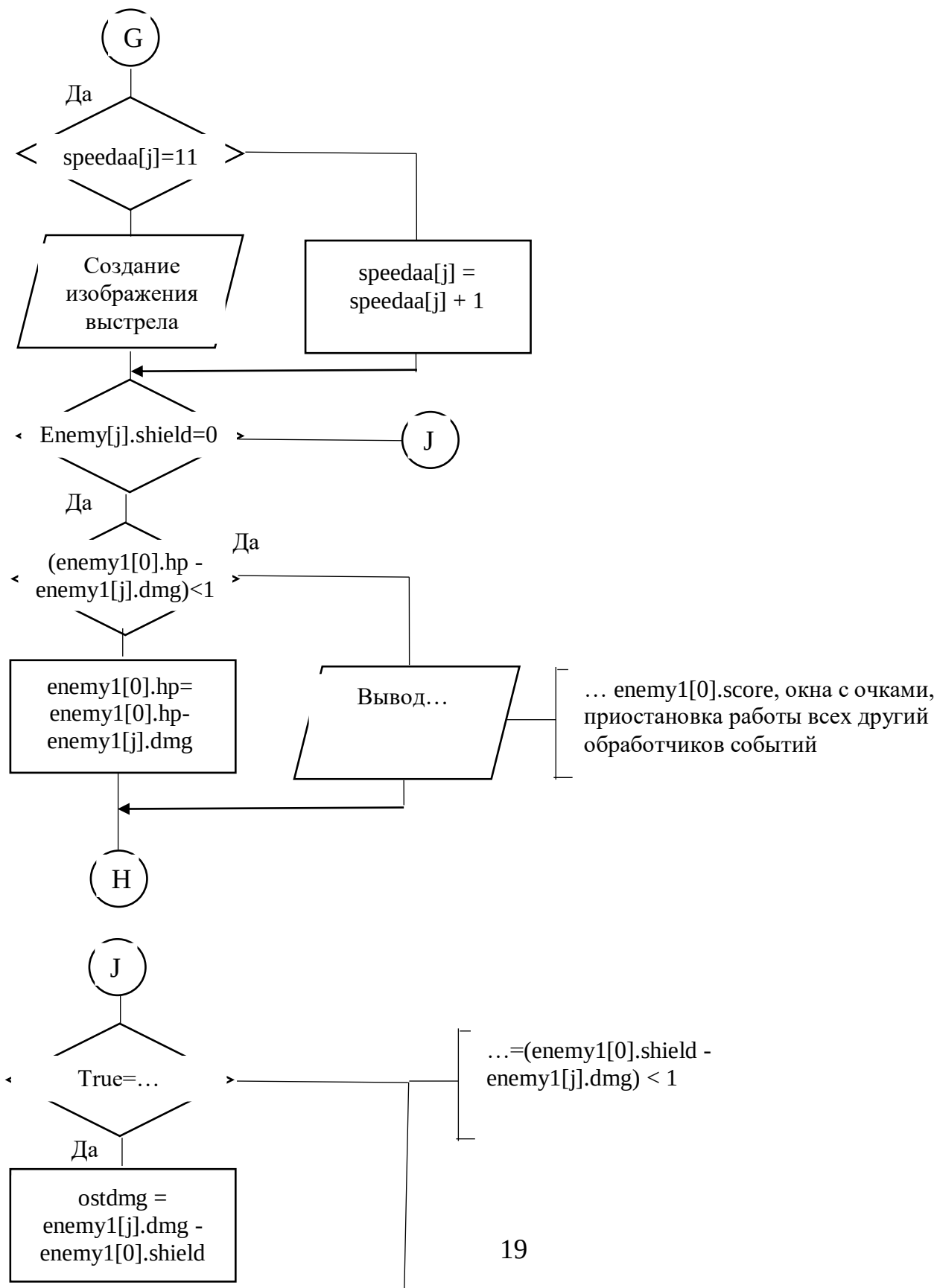


Рисунок 15– Блок-схема обработчика событияtimer9_Tick



Рисунок 16 – Блок-схема обработчика событияtimer8_Tick





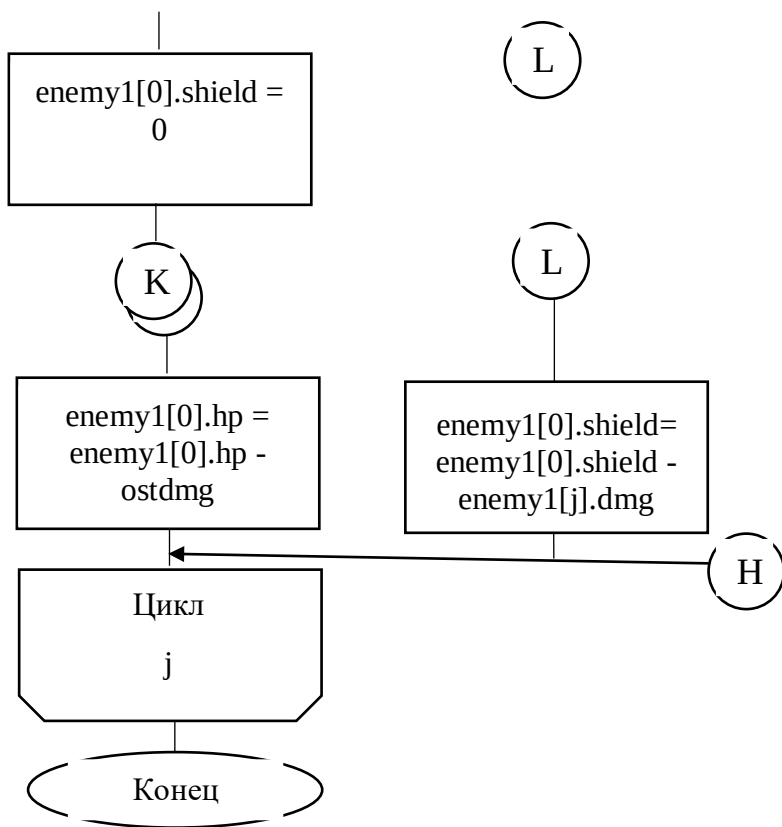
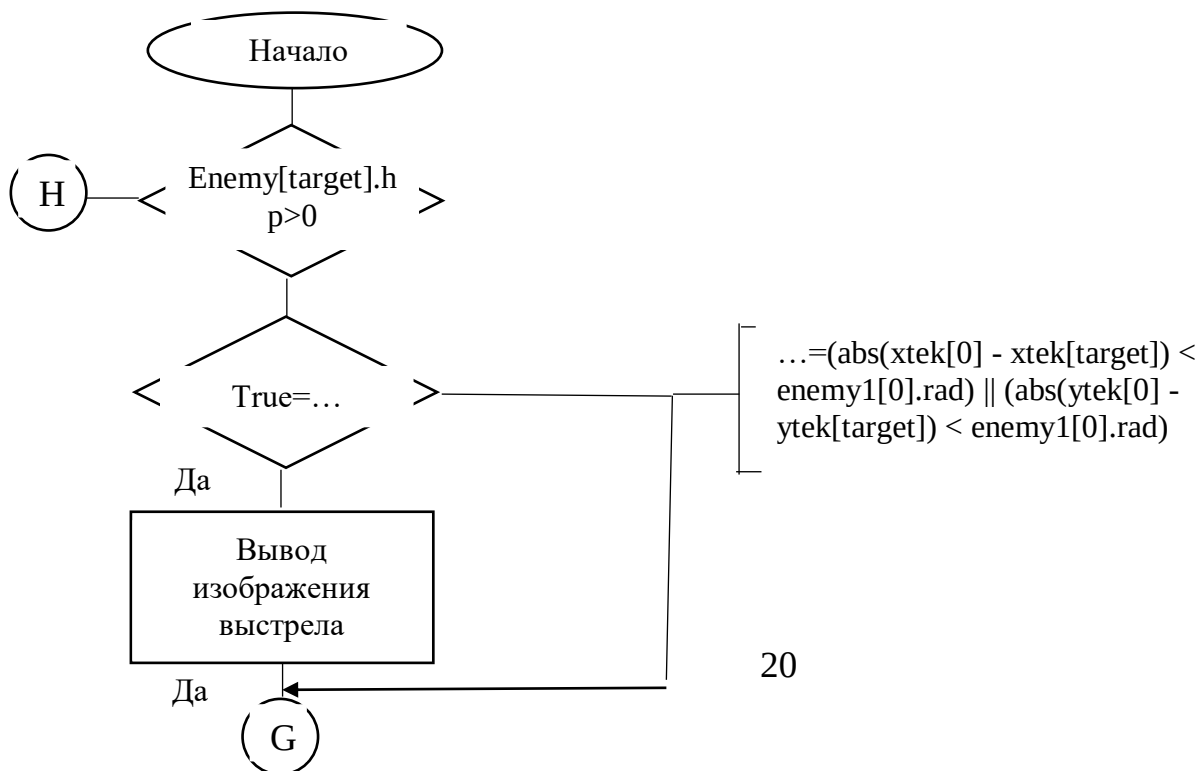
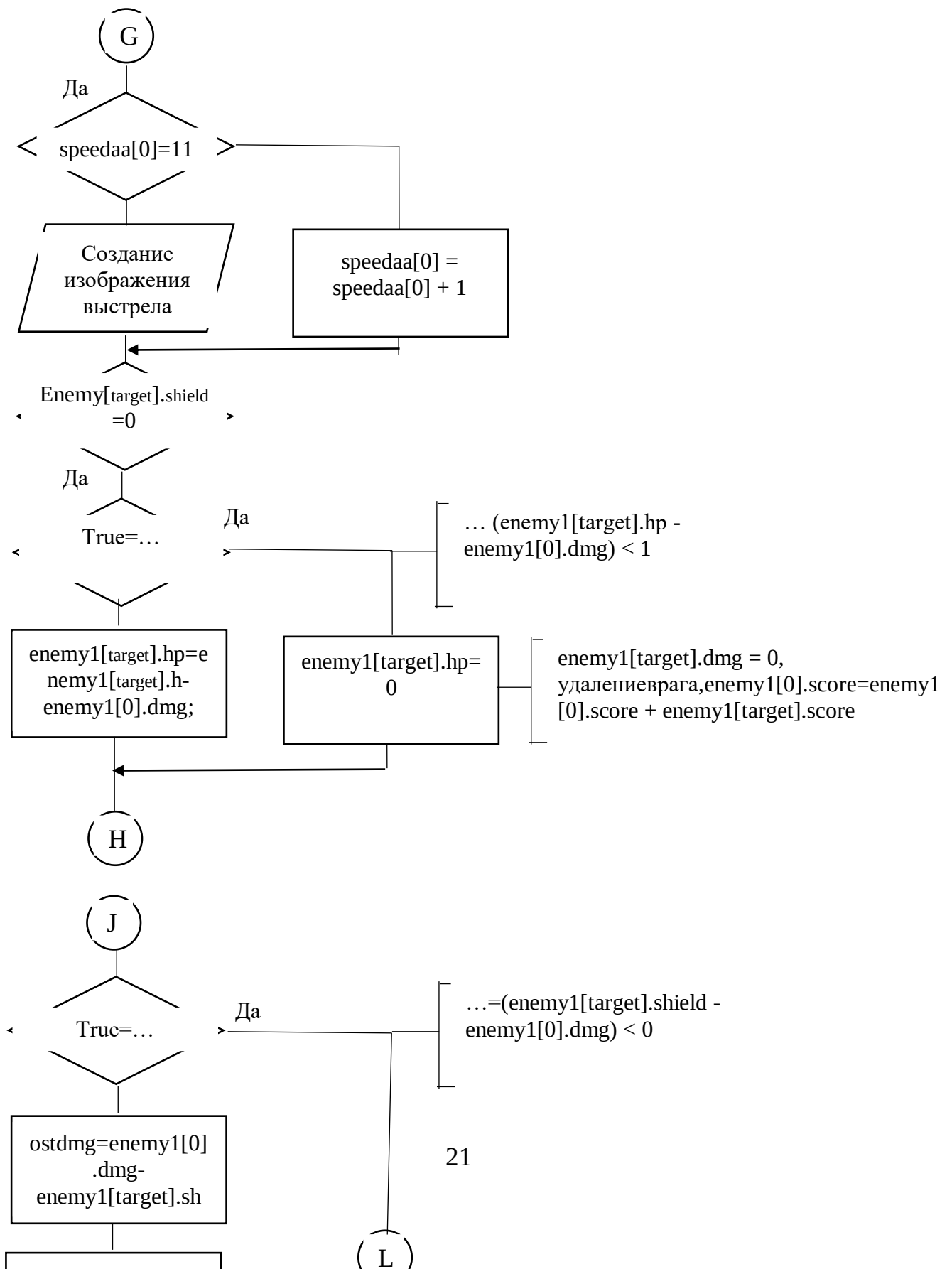


Рисунок 17– Блок-схема обработчика события timer5_Tick





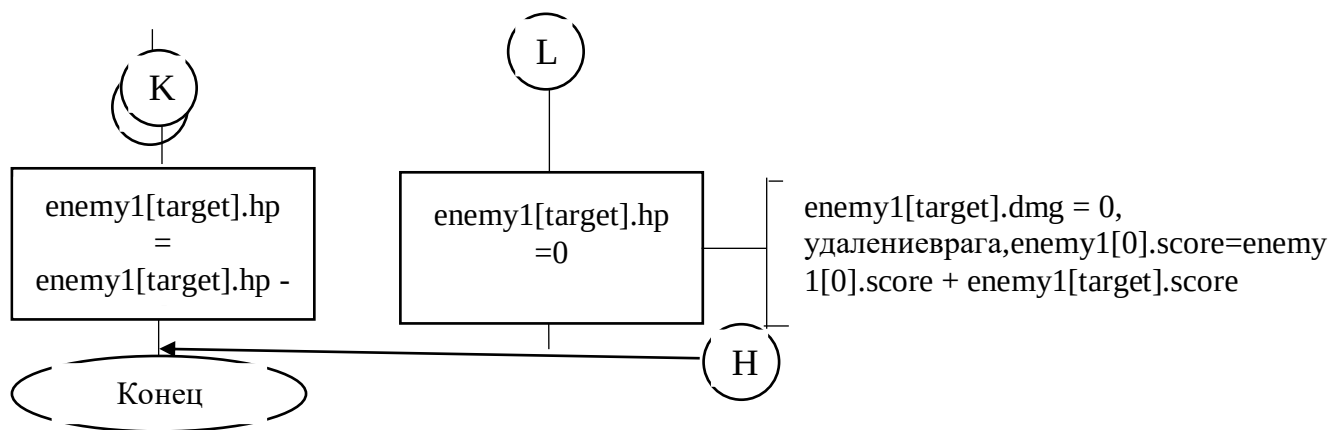


Рисунок 18– Блок-схема обработчика события timer6_Tick

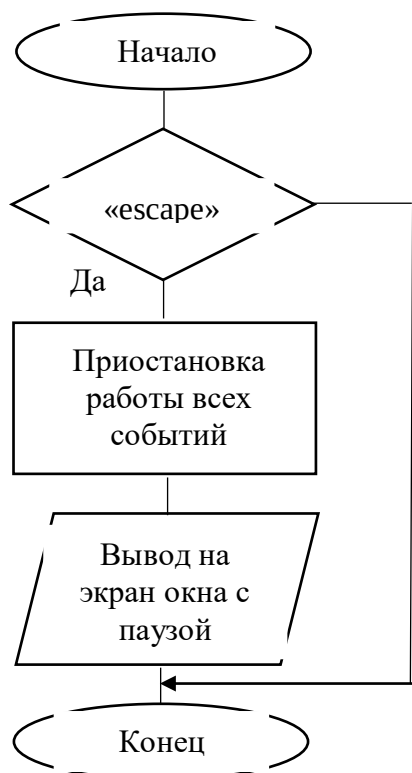
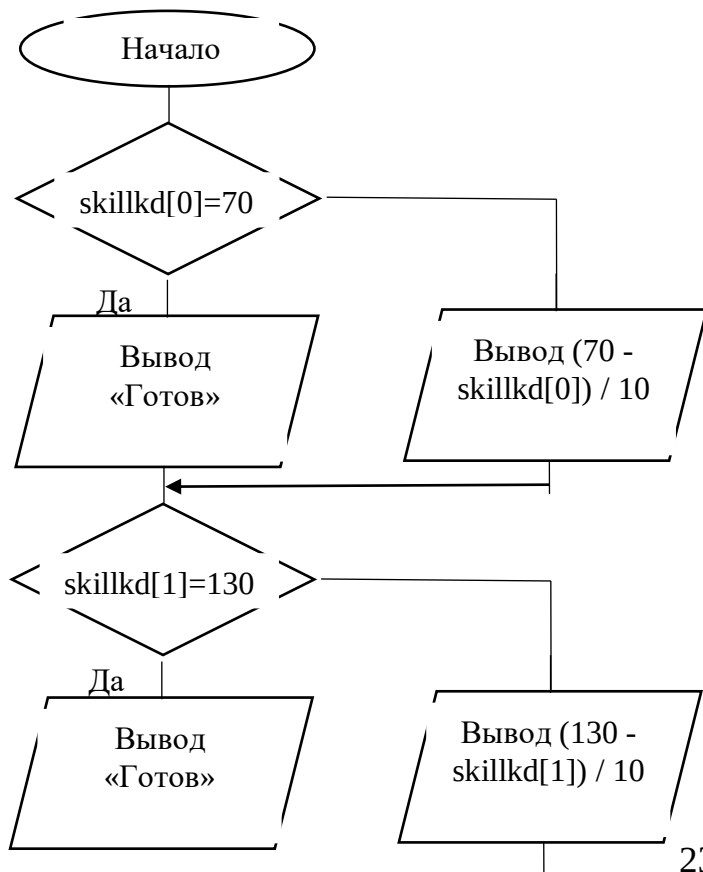
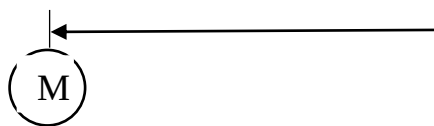
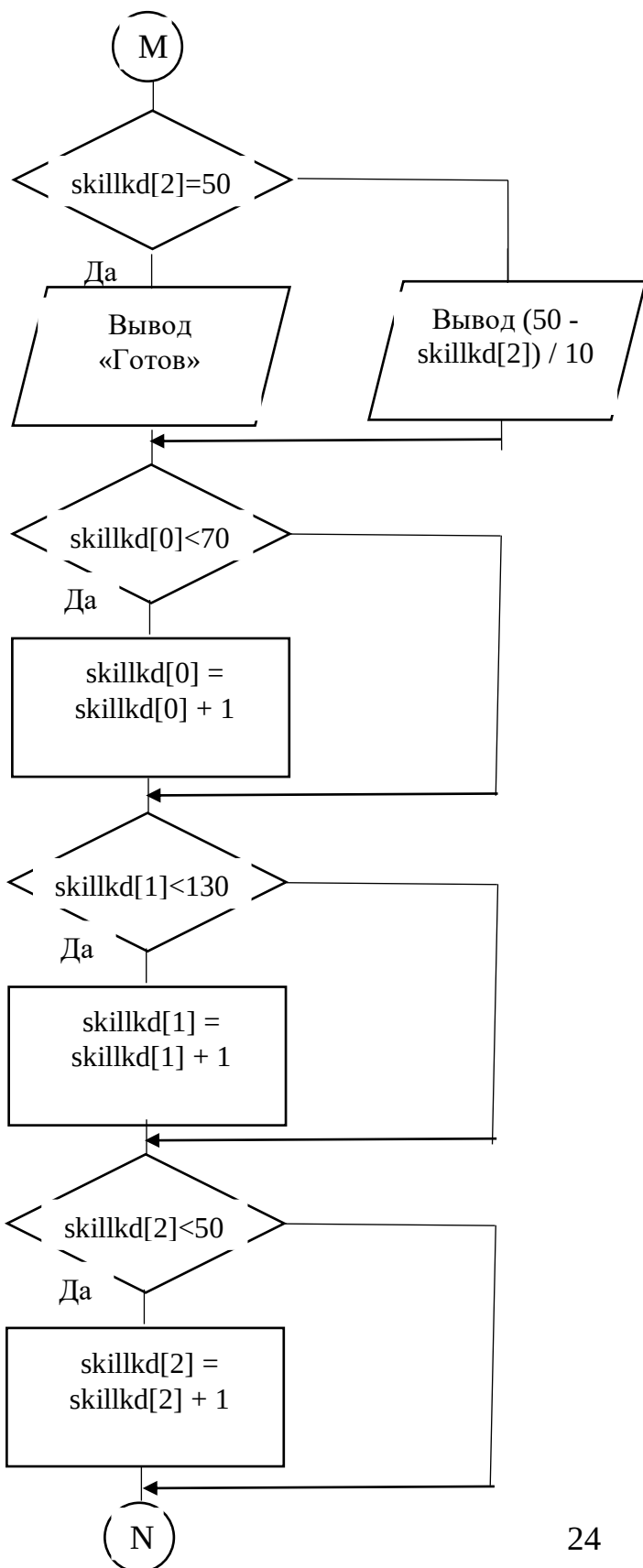


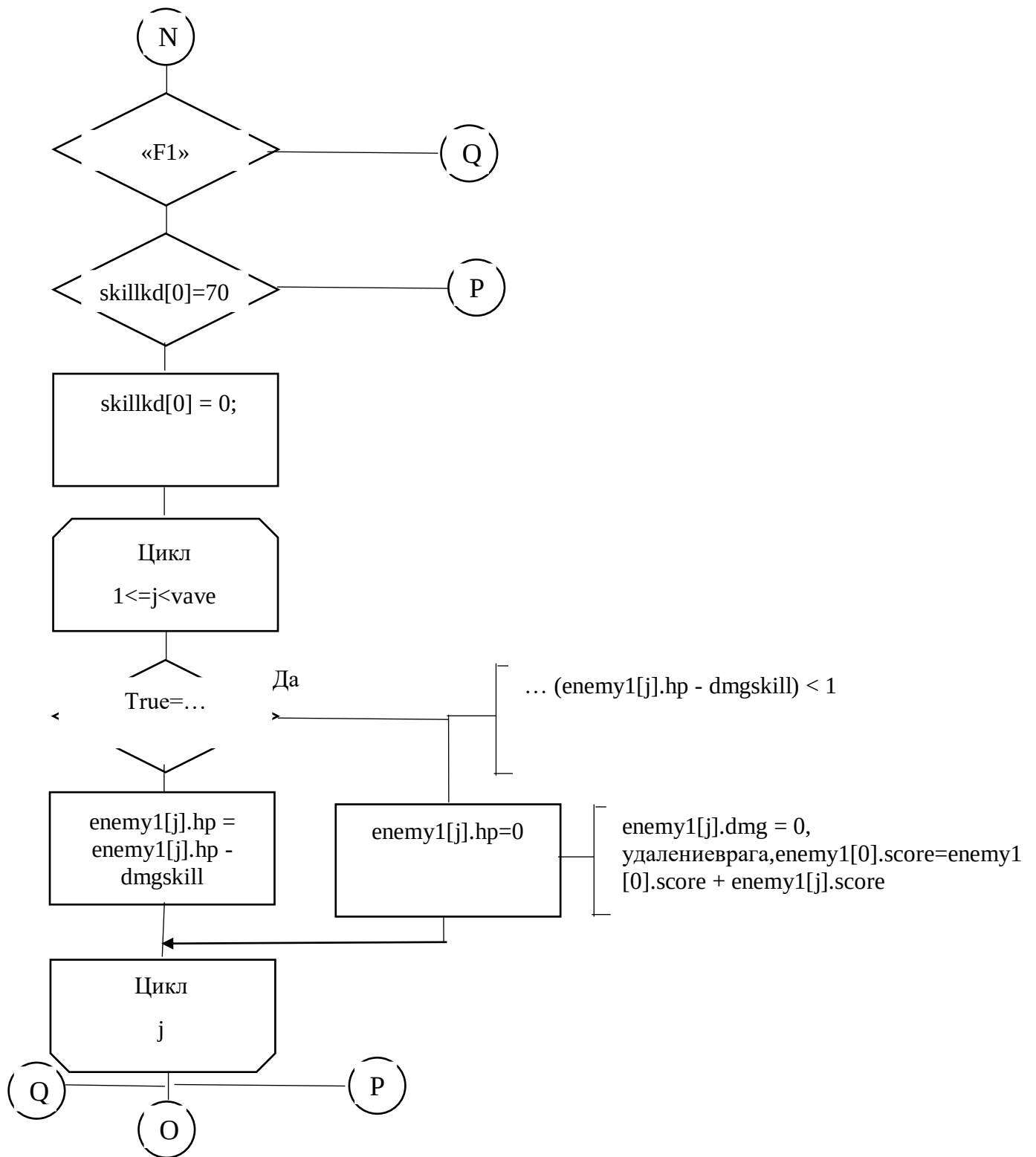
Рисунок 19– Блок-схема обработчика события timer11_Tick

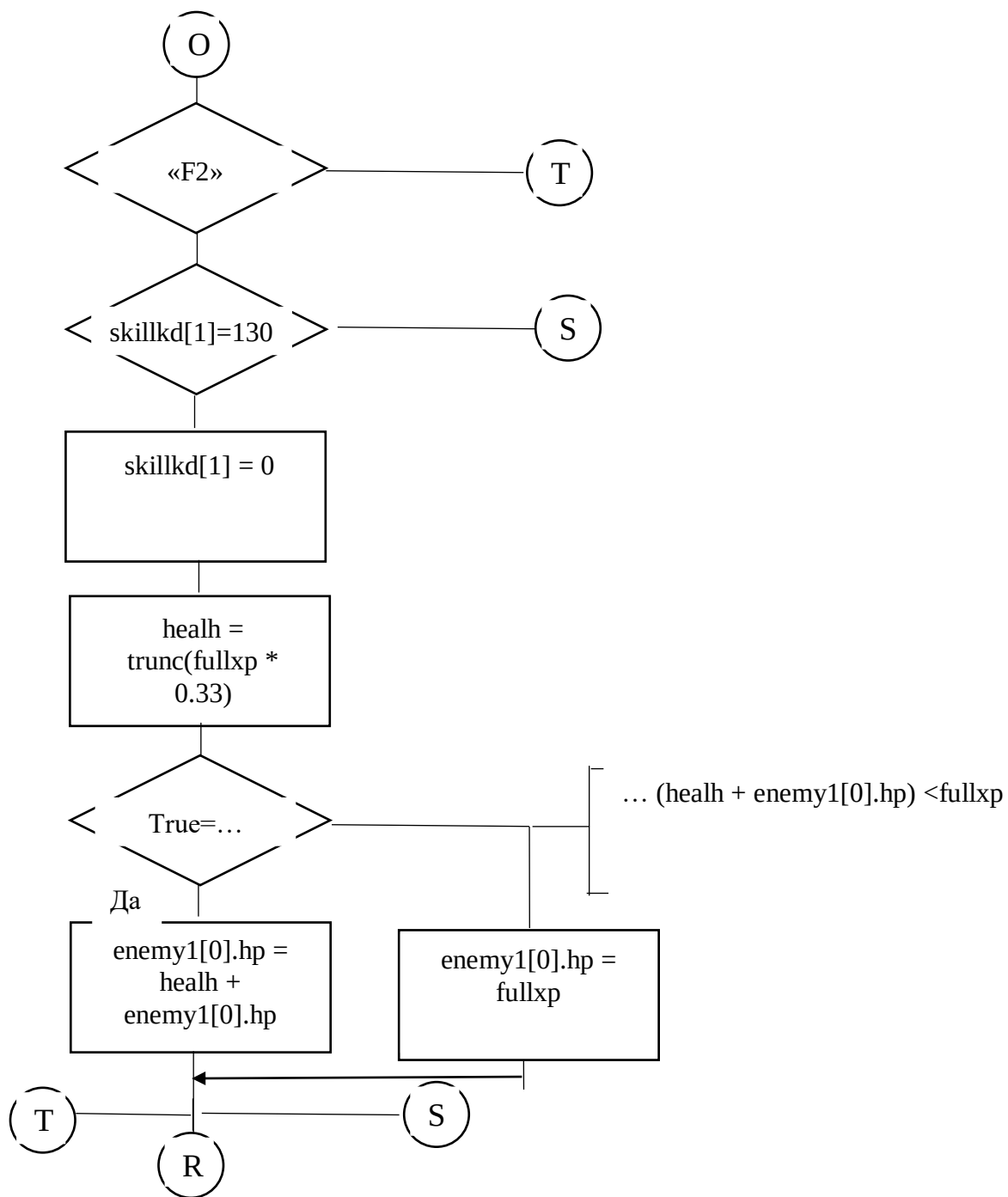


Рисунок 20– Блок-схема обработчика событияbutton2_Click









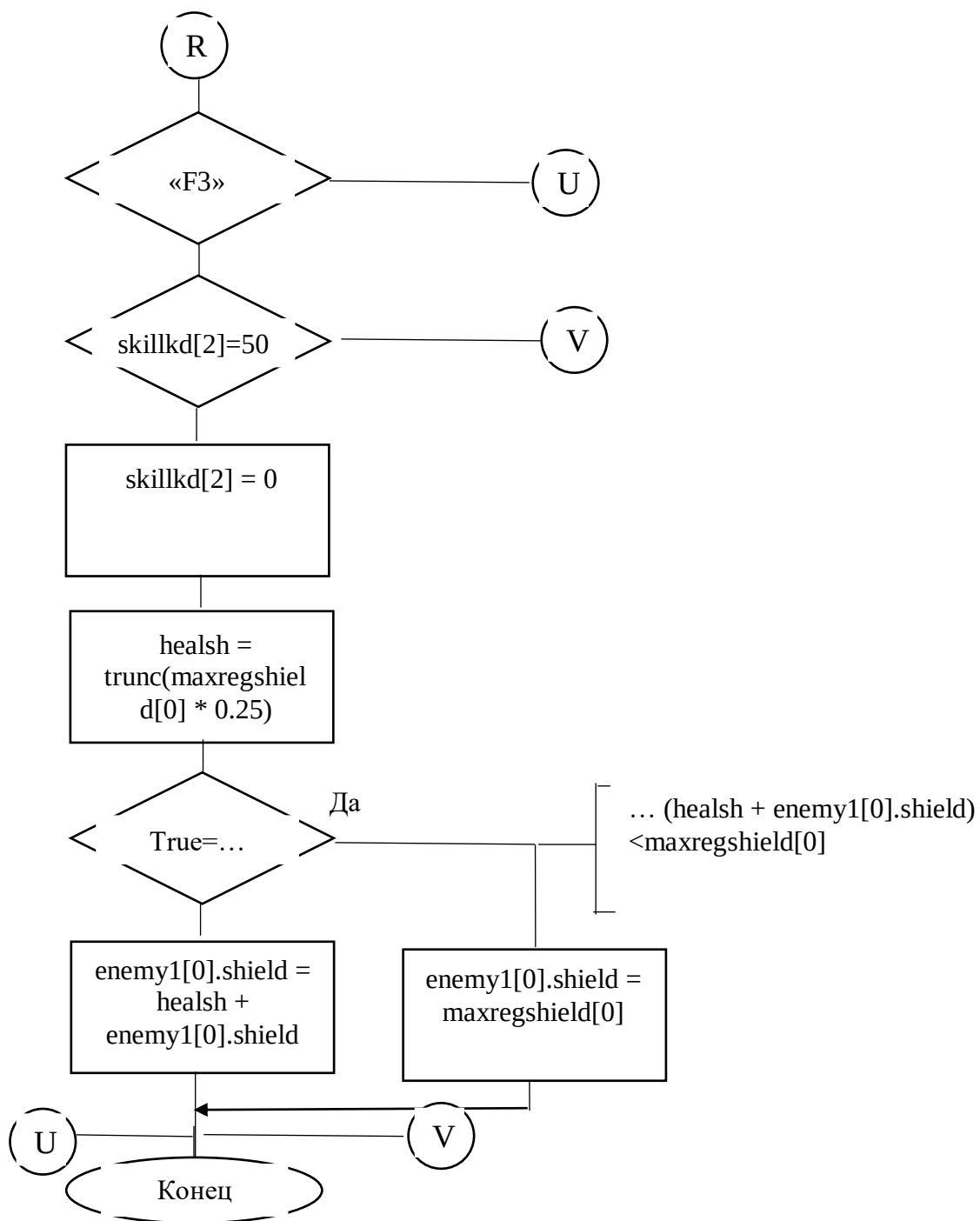


Рисунок 21– Блок-схема обработчика события timer12_Tick



Рисунок 22– Блок-схема обработчика событияbutton3_Click

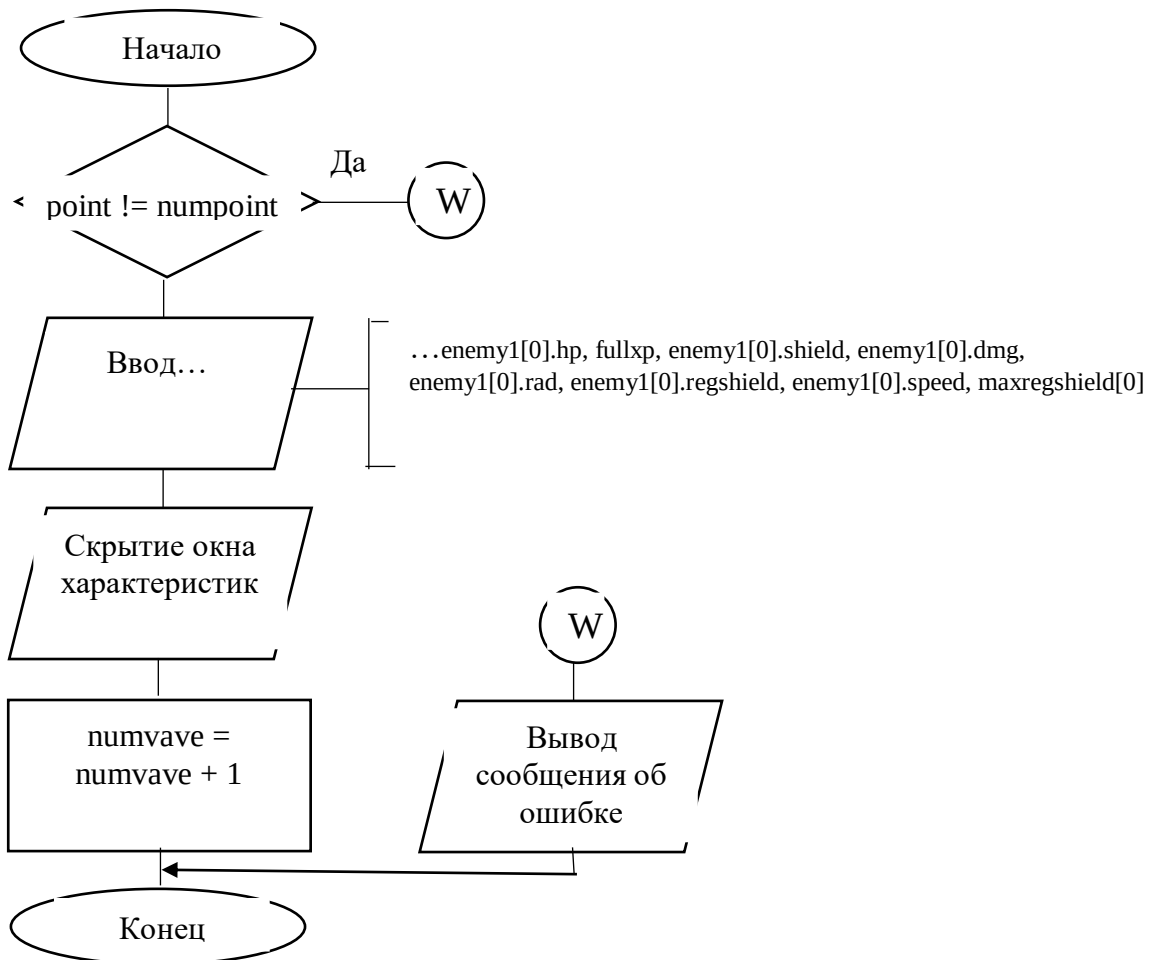


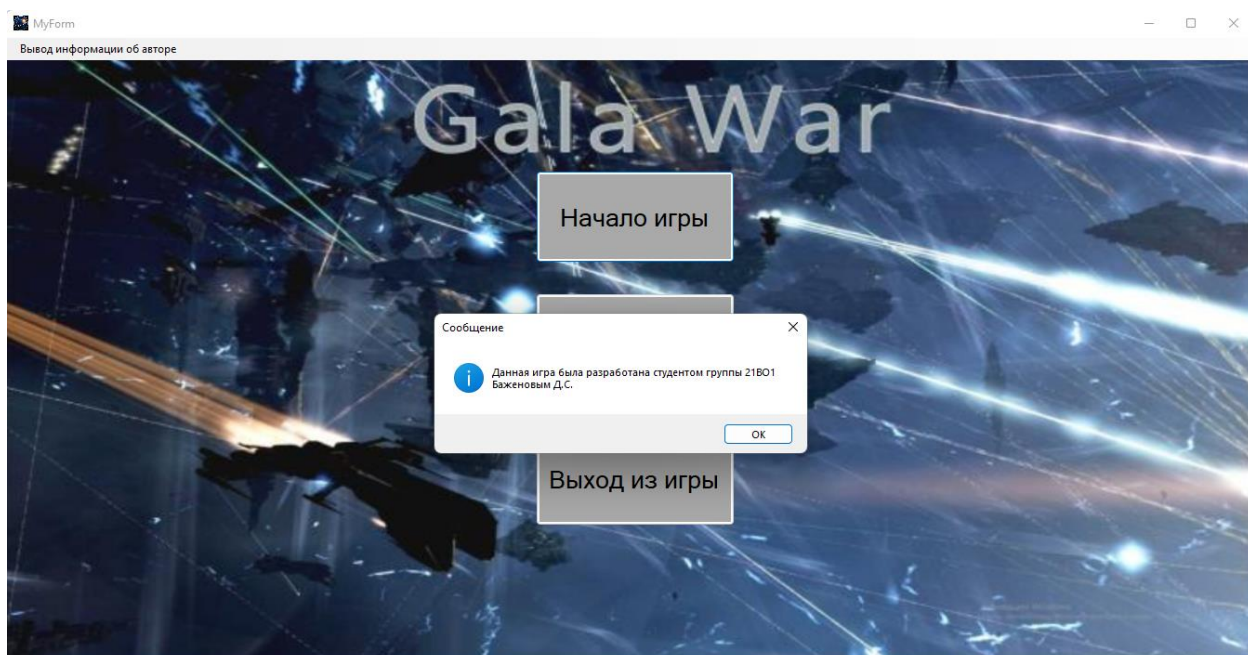
Рисунок 23– Блок-схема обработчика событияbutton1_Click

6. Руководство пользователя

При запуске программы на экран выводится главная заставка.



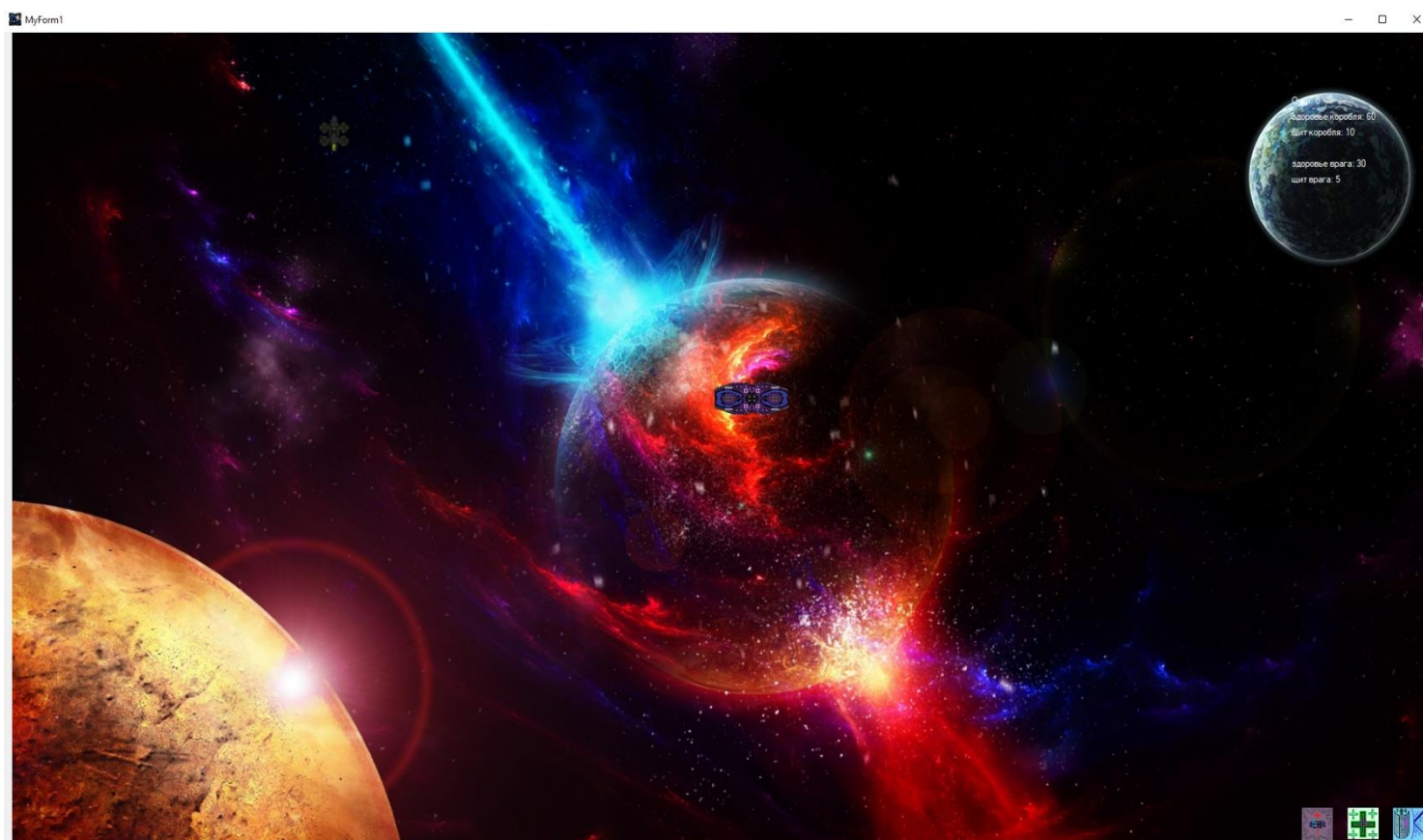
Чтобы увидеть информацию об авторе необходимо нажать на пункт меню «вывод информации об авторе»



Для запуска игры необходимо кликнуть кнопку «начало игры»



После клика, игра сразу начнется, так что будьте готовы

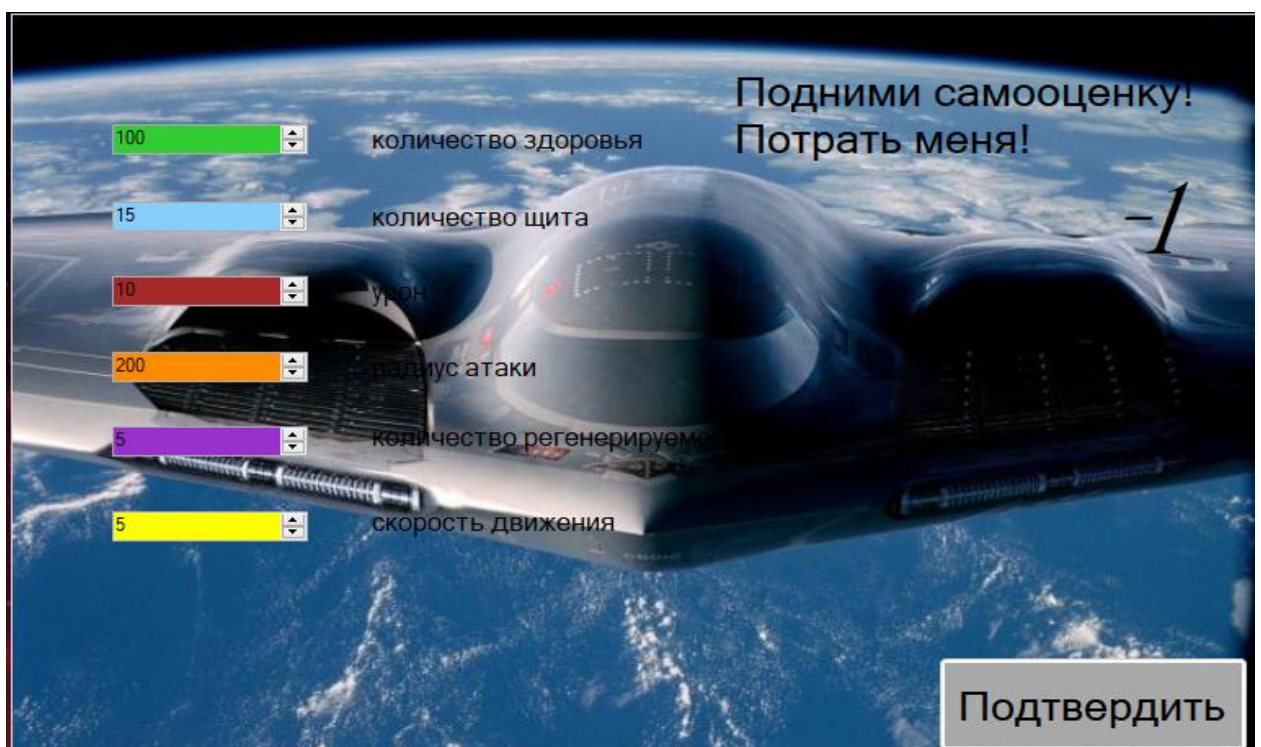


В правом нижнем углу экрана можно увидеть 3 иконки. Это способности корабля, когда они будут готовы, их можно будет использовать.



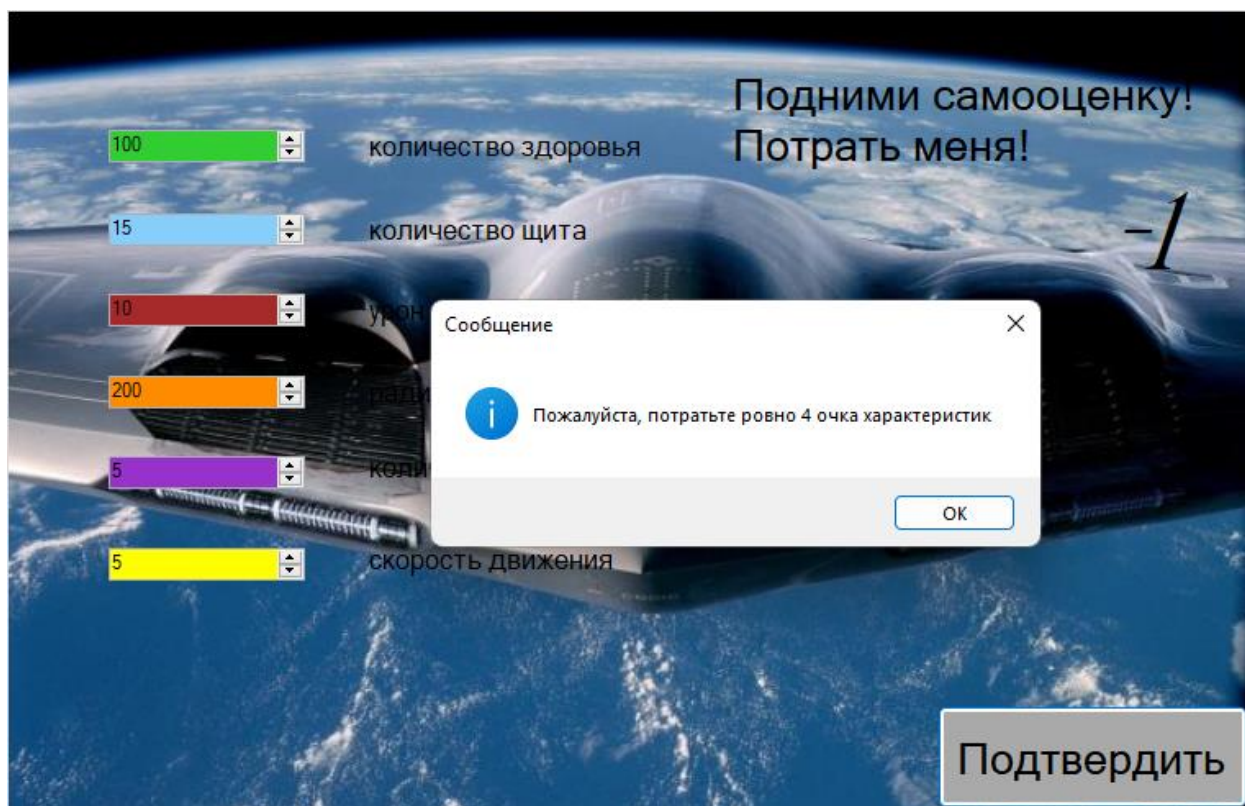
После использования на иконках заново начнет идти время. Это время показывает откат навыков.

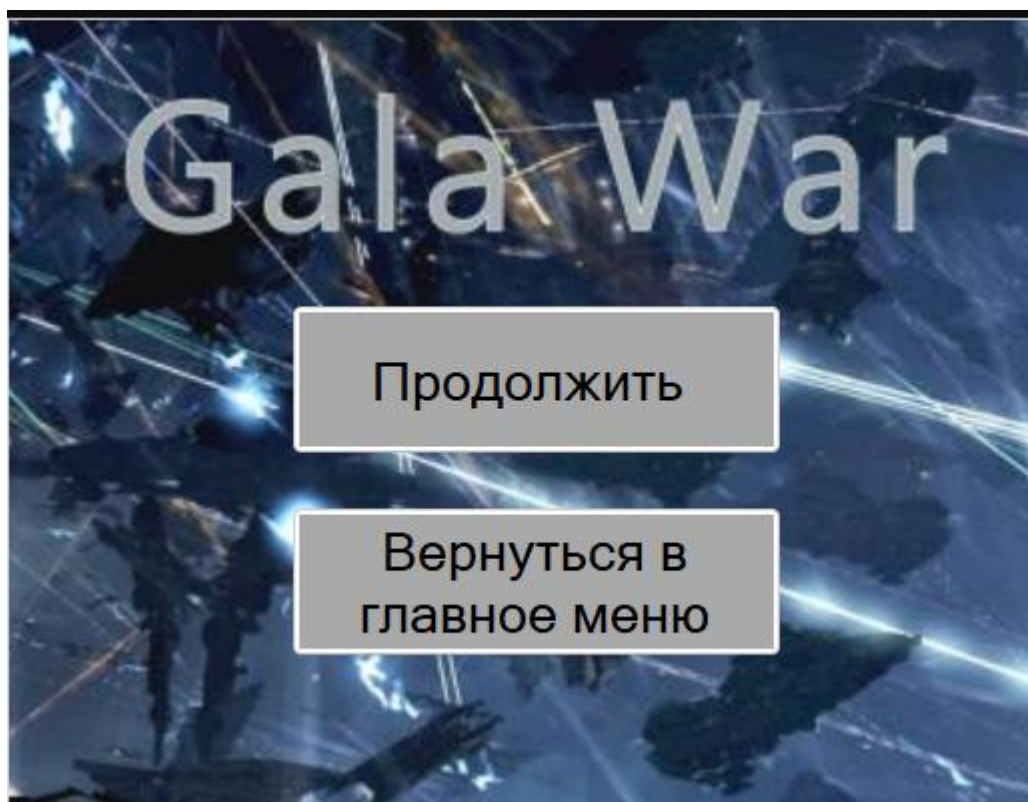
При прохождении каждой волны будет появляться специальное окно с характеристиками главного корабля. Эти характеристики можно улучшать, но не больше четырех за волну.



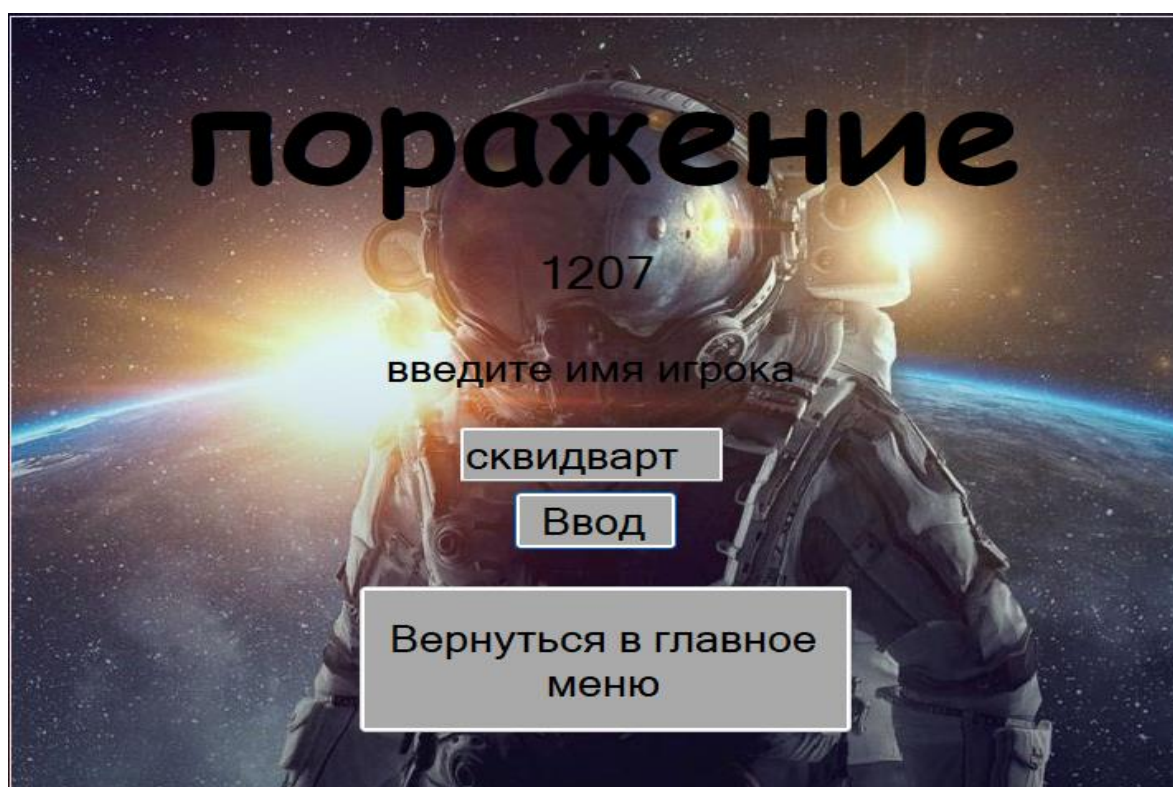
Если вы хотите улучшить больше или меньше четырех характеристик, то вылезет ошибка

В любой момент времени игру можно поставить на паузу при нажатии клавиши «escape»

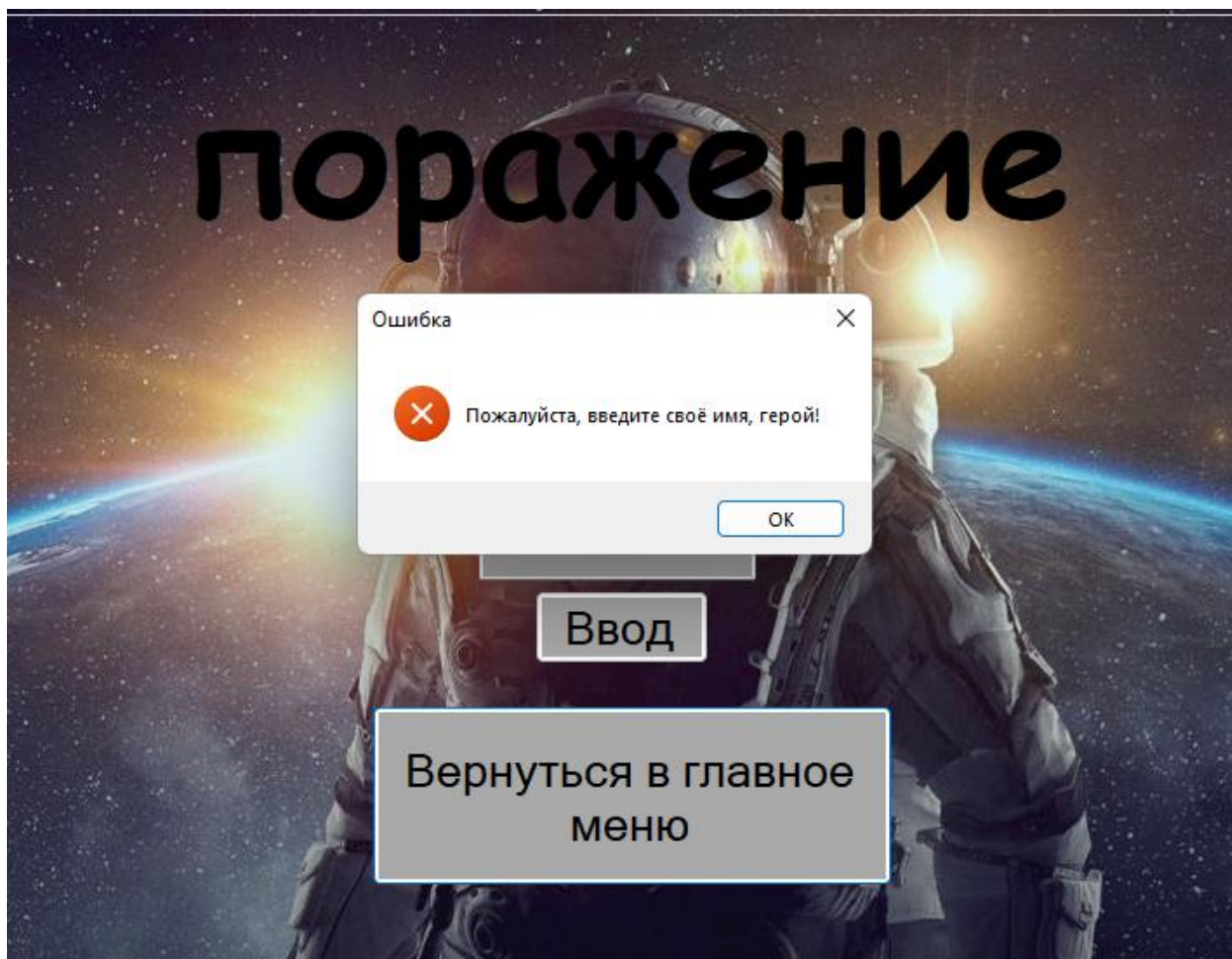




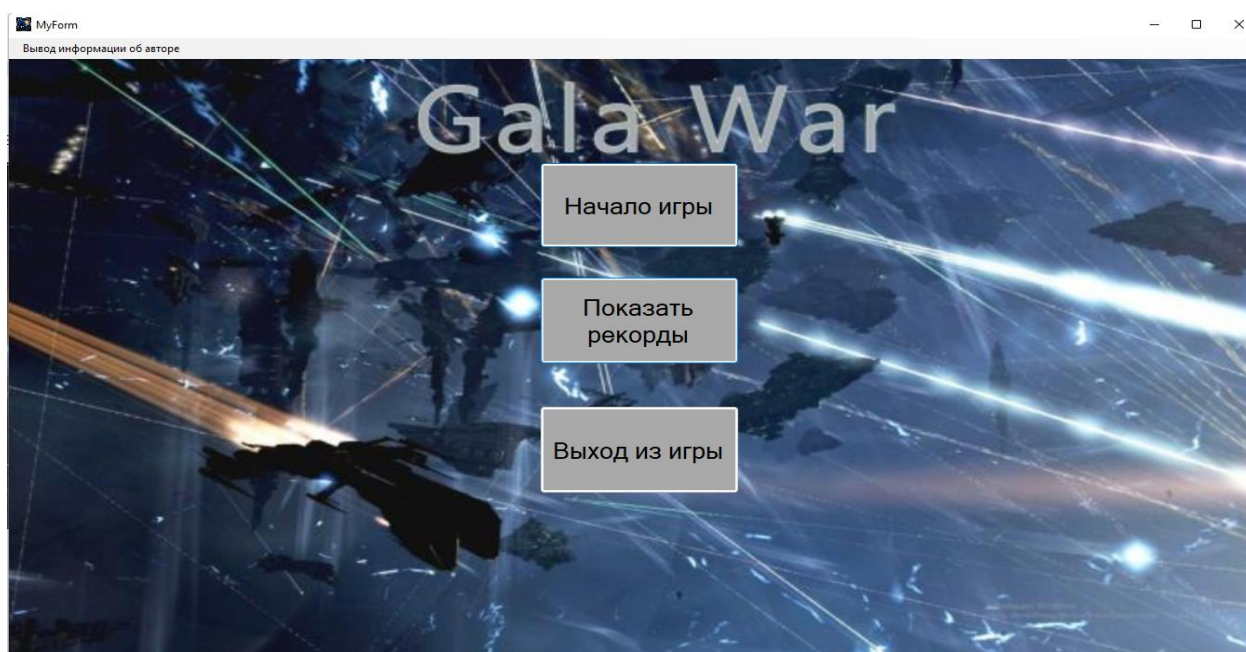
При уничтожении главного корабля появится окно.



Если вы не ввели имя, то появится ошибка



Если нажать на кнопку «показать рекорды» в главном меню, то появится окно со всеми игроками и их очками





При нажатии кнопки «Выход из игры» игра закроется

Заключение

В ходе выполнения данной курсовой работы были получены навыки разработки многомодульных программ. Была освоена работа с файлами, изучены функции работы с объектами и оперативной памятью, освоены приемы создания пользовательского интерфейса, изучены способы обработки клавиатурных нажатий. Также были получены основные навыки работы в среде Microsoft Visual Studio 2022 на языке C++ .

В рамках курсовой работы была написана игра «GaloWar». Она имеет все основные функции для стабильной работы игры.

Список используемых источников

1. Программирование на C и C++, Онлайн справочник программиста на C и C++.
2. METANIT.COM, Сайт о программировании
3. Документация по Microsoft C++, WindowsForm

Приложение А. Листинг программы MyForm1.cpp

```
#include "MyForm1.h"
#include <conio.h>
#include <cmath>
#include <math.h>
#include <Windows.h>
#include <Winuser.h>
#define WM_KEYDOWN

const int n = 100; // КОЛИЧЕСТВО ВРАГОВ

int maxhpenemy[n]; // для формулы урона со скила 1
int skillkd[3]; // кд скила
int point = 0; // текущее количество доступных очков
int numpoint = 4; // количество очков распределения
int fullxp; // фулл здоровье для гр (скил)
int respx = 100;
bool endvave; // проверка конца волны
int maxhp = 20; // для формулы здоровья
int maxdmg = 15; // для формулы урона
int maxshield = 0; // для формулы щита
int maxregshield1 = 2; // для формулы регена щита
int maxregshield[n]; // для регена щита
int kdregsh; //
int f = 20; // отступ по иксу для врагов
int target; // индекс цели
bool targ = true;
int speedaa[n]; // скорость атаки
int xtek[n+1], ytek[n+1];
bool stmc = true, stenemy = true;

struct stats
{
    int hp = 20;
    int regshield = 0;
    int shield = 0;
    int dmg = 15;
    int rad = 150;
    int speed = 5;
    int score = 100;
};
stats enemy1[100];
stats boss;

POINT pp[n]; // текущая позиция корабля
POINT p[n]; // позиция курсора/позиция главного корабля для врагов

using namespace System;
using namespace System::Windows::Forms;
using namespace Project1;
using namespace System::Drawing;
using namespace System::Drawing::Imaging;
using namespace System::IO;
```

```

System::Void Project1::MyForm1::MyForm1_Load(System::Object^ sender, System::EventArgs^ e)
{
    skillkd[0] = 0; //скилы
    skillkd[1] = 0;
    skillkd[2] = 0;

    groupBox1->Enabled = true; //убрать фон
    groupBox1->Controls->Add(label2);
    label2->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Controls->Add(label3);
    label3->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Controls->Add(label4);
    label4->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Controls->Add(label5);
    label5->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Controls->Add(label6);
    label6->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Controls->Add(label7);
    label7->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Controls->Add(label8);
    label8->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Controls->Add(label9);
    label9->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox1->Enabled = false;

    groupBox3->Enabled = true; //убрать фон
    groupBox3->Controls->Add(label12);
    label12->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox3->Controls->Add(label11);
    label11->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox3->Controls->Add(label10);
    label10->BackColor = System::Drawing::Color(Color::Transparent);
    groupBox3->Enabled = false;

    enemy1[0].hp = 60; //стандартные статы гк
    enemy1[0].shield = 10;
    maxregshield[0] = enemy1[0].shield;
    enemy1[0].dmg = 10;
    enemy1[0].rad = 200;
    enemy1[0].regshield = 5;
    enemy1[0].speed = 5;
    enemy1[0].score = 0;

    numericUpDown1->Minimum = enemy1[0].hp; //стандартные статы видно после волны
    numericUpDown2->Minimum = enemy1[0].shield;
    numericUpDown3->Minimum = enemy1[0].dmg;
    numericUpDown4->Minimum = enemy1[0].rad;
    numericUpDown5->Minimum = enemy1[0].regshield;
    numericUpDown6->Minimum = enemy1[0].speed;

    numericUpDown1->Increment = 10;
    numericUpDown2->Increment = 5;
    numericUpDown3->Increment = 5;
    numericUpDown4->Increment = 20;
    numericUpDown5->Increment = 5;
    numericUpDown6->Increment = 1;

    numericUpDown1->Maximum = enemy1[0].hp + (numpoint * numericUpDown1->Increment);
    numericUpDown2->Maximum = enemy1[0].shield + (numpoint * numericUpDown2->Increment);
    numericUpDown3->Maximum = enemy1[0].dmg + (numpoint * numericUpDown3->Increment);
    numericUpDown4->Maximum = enemy1[0].rad + (numpoint * numericUpDown4->Increment);
    numericUpDown5->Maximum = enemy1[0].regshield + (numpoint * numericUpDown5->Increment);
    numericUpDown6->Maximum = enemy1[0].speed + (numpoint * numericUpDown6->Increment);
}

```

```

        numericUpDown6->Maximum = enemy1[0].speed + (numpoint * numericUpDown6->Increment);

        numericUpDown1->Value = enemy1[0].hp;
        numericUpDown2->Value = enemy1[0].shield;
        numericUpDown3->Value = enemy1[0].dmg;
        numericUpDown4->Value = enemy1[0].rad;
        numericUpDown5->Value = enemy1[0].regshield;
        numericUpDown6->Value = enemy1[0].speed;
    }

    System::Void Project1::MyForm1::button4_Click(System::Object^ sender, System::EventArgs^ e)//ввод данных в файл
    {
        heroname = textBox1->Text;
        String^ filename = "C:\\проги для програмирования\\praktika\\records.txt";
        StreamWriter^ file = gcnew StreamWriter(filename, true);
        file->WriteLine(heroname + " получил " + Convert::ToString(enemy1[0].score) + " очка(ов)" + "\\n");
        //file->
        file->Close();
        delete filename;
    }

    System::Void Project1::MyForm1::button5_Click(System::Object^ sender, System::EventArgs^ e)//выход при поражении
    {
        if ((heroname == " ") || (heroname == ""))
        {
            MessageBox::Show(this, "Пожалуйста, введите своё имя, герой!", "Ошибка",
            MessageBoxButtons::OK, MessageBoxIcon::Error);
        }
        else
        {
            for (size_t j = 0; j < n - 1; j++)
            {
                if (j == 0)
                {
                    delete laz[j];
                    delete shield[j];
                    delete health[j];
                }
                delete laz[j];
                delete shield[j];
                delete health[j];
                delete pb[j];
            }
            delete heroname;
            delete this;
        }
    }

    System::Void Project1::MyForm1::timer10_Tick(System::Object^ sender, System::EventArgs^ e)//проверка на потраченное количество очков
    {
        int prov[6];
        prov[0] = Convert::ToInt64(numericUpDown1->Value) - Convert::ToInt64(numericUpDown1->Minimum);
        prov[1] = Convert::ToInt64(numericUpDown2->Value) - Convert::ToInt64(numericUpDown2->Minimum);
        prov[2] = Convert::ToInt64(numericUpDown3->Value) - Convert::ToInt64(numericUpDown3->Minimum);
        prov[3] = Convert::ToInt64(numericUpDown4->Value) - Convert::ToInt64(numericUpDown4->Minimum);
    }

```

```

        prov[4] = Convert::ToInt64(numericUpDown5->Value) - Convert::ToInt64(numericUpDown5-
>Minimum);
        prov[5] = Convert::ToInt64(numericUpDown6->Value) - Convert::ToInt64(numericUpDown6-
>Minimum);

        point = (prov[0] / Convert::ToInt64(numericUpDown1->Increment)) + (prov[1] /
Convert::ToInt64(numericUpDown2->Increment)) + (prov[2] /
Convert::ToInt64(numericUpDown3->Increment)) + (prov[3] /
Convert::ToInt64(numericUpDown4->Increment)) + (prov[4] /
Convert::ToInt64(numericUpDown5->Increment)) + (prov[5] /
Convert::ToInt64(numericUpDown6->Increment));
        label9->Text = Convert::ToString(numpoint - point);
    }

System::Void Project1::MyForm1::timer1_Tick(System::Object^ sender, System::EventArgs^
e)//движение гл кор
{

    if (stmc == true)
    {
        xtek[0] = 960;
        ytek[0] = 560;
        stmc = false;
    }
    pictureBox1->Controls->Add(pictureBox2); //чтобы фон убрать
    pictureBox2->BackColor = System::Drawing::Color::Transparent;//чтобы фон убрать

    GetCursorPos(&p[0]);

    int x = p[0].x;
    int y = p[0].y;
    p[0].x = x + 8;
    p[0].y = y + 14;
    pp[0].x = pictureBox2->Location.X + 50;
    pp[0].y = pictureBox2->Location.Y + 60;

    int iSpeed = enemy1[0].speed; // скорость ////////////////////////////////// до 13

    // дистанция между курсором и pictureBox1 (главный корабль) (в пикселях)
    double fDistance = sqrt((p[0].x - pp[0].x) * (p[0].x - pp[0].x) + (p[0].y - pp[0].y) *
(p[0].y - pp[0].y));
    // если pictureBox1 (главный корабль) рядом с курсором, то выход (чтобы не дёргался)
    if (fDistance < iSpeed) return;

    // определяем угол в радианах между катетами, образованными позицией курсора и
позицией pictureBox1 (главный корабль)
    double fRadians = 0;
    if (p[0].x != pp[0].x)
    {
        fRadians = atan(1.0 * abs(p[0].y - pp[0].y) / abs(p[0].x - pp[0].x));
    }

    // определяем, на сколько нужно изменить позицию pictureBox1 (главный корабль) по
осям
    double fDiffX = fabs(iSpeed * cos(fRadians));
    double fDiffY = fabs(iSpeed * sin(fRadians));

    // преобразование fDiffX, fDiffY в целые числа
    int iDiffX = fDiffX;
    int iDiffY = fDiffY;

    // изменение позиции pictureBox1 (главный корабль)
    if (pp[0].x < p[0].x) xtek[0] += iDiffX;
    else if (pp[0].x > p[0].x) xtek[0] -= iDiffX;

```

```

    if (pp[0].y < p[0].y) ytek[0] += iDiffY;
    else if (pp[0].y > p[0].y) ytek[0] -= iDiffY;
    pictureBox2->Location = System::Drawing::Point(xtek[0], ytek[0]);
}

System::Void Project1::MyForm1::timer2_Tick(System::Object^ sender, System::EventArgs^ e)//запуск волны
{
    if (vave == numvave)
    {
        label1->BackColor = System::Drawing::Color(Color::Transparent);
        label1->BringToFront();
        pictureBox1->Controls->Add(label1);

        health[0] = gcnew Label();//создание label с хп гк
        health[0]->Name = "health" + name + name;
        health[0]->BackColor = System::Drawing::Color(Color::Black);
        health[0]->ForeColor = System::Drawing::Color(Color::White);
        health[0]->Location = System::Drawing::Point(1700, 100);
        health[0]->Text = "";
        health[0]->Size = System::Drawing::Size(150, 13);
        health[0]->AutoSize;
        health[0]->BringToFront();
        pictureBox1->Controls->Add(health[0]);

        f = 20;

        shield[0] = gcnew Label();//создание label с щитом гк
        shield[0]->Name = "shield" + name + name + name;
        shield[0]->BackColor = System::Drawing::Color(Color::Black);
        shield[0]->ForeColor = System::Drawing::Color(Color::White);
        shield[0]->Location = System::Drawing::Point(1700, 100 + f);
        shield[0]->Text = "";
        shield[0]->Size = System::Drawing::Size(150, 13);
        shield[0]->AutoSize;
        shield[0]->BringToFront();
        pictureBox1->Controls->Add(shield[0]);

        stenemy = true;
        respx = 100;
        for (size_t j = 1; j < 1 + vave; j++)
        {
            f += 20;
            pb[j] = gcnew PictureBox();
            pb[j]->Image = Image::FromFile("C:\\проги для
программирования\\practika\\enemy.PNG");
            pb[j]->Location = System::Drawing::Point(50 * j, 0);
            pb[j]->Size = System::Drawing::Size(199, 100);
            pb[j]->Name = "PictureBox" + name + j;
            pb[j]->SizeMode = System::Windows::Forms::PictureBoxSizeMode::AutoSize;
            pb[j]->Enabled = true;
            pb[j]->Visible = true;
            pb[j]->BringToFront();
            this->Controls->Add(pb[j]);

            enemy1[j].dmg = trunc(maxdmg + (5 * vave));
            enemy1[j].shield = trunc(maxshield + (5 * vave));
            maxregshield[j] = enemy1[j].shield;//количество щита которое можно отрегениить
            enemy1[j].hp = trunc(maxhp + (10 * vave));
            maxhpenemy[j] = enemy1[j].hp;
            enemy1[j].regshield = trunc(maxregshield1 + (2 * vave));

            enemy1[j].score = trunc(pow(1.2, vave) * (40 + enemy1[j].dmg + enemy1[j].hp +
            enemy1[j].shield + enemy1[j].regshield));

```

```

        f += 20;
        health[j] = gcnew Label(); //создание label с хп врага
        health[j]->Name = "health" + name + name + j;
        health[j]->BackColor = System::Drawing::Color(Color::Black);
        health[j]->ForeColor = System::Drawing::Color(Color::White);
        health[j]->Location = System::Drawing::Point(1700, 100 + f);
        health[j]->Text = "";
        health[j]->Size = System::Drawing::Size(150, 13);
        health[j]->AutoSize;
        health[j]->BringToFront();
        pictureBox1->Controls->Add(health[j]);

        f += 20;
        shield[j] = gcnew Label(); //создание label с щитом врага
        shield[j]->Name = "shield" + name + name + name + j;
        shield[j]->BackColor = System::Drawing::Color(Color::Black);
        shield[j]->ForeColor = System::Drawing::Color(Color::White);
        shield[j]->Location = System::Drawing::Point(1700, 100 + f);
        shield[j]->Text = "";
        shield[j]->Size = System::Drawing::Size(150, 13);
        shield[j]->AutoSize;
        shield[j]->BringToFront();
        pictureBox1->Controls->Add(shield[j]);
    }

    ++vave;
    timer4->Enabled = true; //запуск всех timer не считая таймера с обработкой выхода
    timer5->Enabled = true;
    timer6->Enabled = true;
    timer7->Enabled = true;
    timer8->Enabled = true;
    timer9->Enabled = true;
    timer3->Enabled = true;
    timer12->Enabled = true;
}

}

System::Void Project1::MyForm1::timer3_Tick(System::Object^ sender, System::EventArgs^ e) //проверка для остановки на волне
{
    endvave = true;
    for (size_t j = 1; j < vave ; j++)
    {
        if (enemy1[j].hp == 0)
        {
            endvave = true;
        }
        else
        {
            endvave = false;
            break;
        }
    }
    if (endvave == true)
    {
        for (size_t j = 1; j < vave; j++)
        {
            pb[j]->Invalidate();
            timer1->Enabled = false; //прекращение работы всех таймеров не считая таймера с обработкой выхода
            timer2->Enabled = false;
            timer4->Enabled = false;
            timer5->Enabled = false;
            timer6->Enabled = false;

```

```

        timer7->Enabled = false;
        timer8->Enabled = false;
        timer9->Enabled = false;
        timer12->Enabled = false;

    }
    timer10->Enabled = true;//включение таймера для повышения статусов после каждой волны
    groupBox1->Enabled = true;
        groupBox1->Location = System::Drawing::Point(515, 278);
        groupBox1->Visible = true;
        timer3->Enabled = false;
    }
}

System::Void Project1::MyForm1::timer7_Tick(System::Object^ sender, System::EventArgs^ e)//цель
{
    targ = false;
    target = 0;
    int min = 1;
    for (size_t j = 1; j < vave; j++)
    {
        if (enemy1[j].hp > 0)
        {
            targ = true;
            if ((abs(xtek[0]-xtek[j]) < abs(xtek[0] - xtek[min])) || (abs(ytek[0] - ytek[j]) < abs(ytek[0] - ytek[min])))
            {
                min = j;
            }
        }
    }
    if (targ == true)
    {
        target = min;
    }
    else
    {
        target = 200;
    }
}

System::Void Project1::MyForm1::timer9_Tick(System::Object^ sender, System::EventArgs^ e)//регенерация щита
{
    for (size_t j = 0; j < vave; j++)
    {
        if (enemy1[j].hp > 0)
        {
            if ((enemy1[j].regshield + enemy1[j].shield) < maxregshield[j])
            {
                enemy1[j].shield = enemy1[j].shield + enemy1[j].regshield;
            }
            else
            {
                enemy1[j].shield = maxregshield[j];
            }
        }
    }
}

System::Void Project1::MyForm1::timer8_Tick(System::Object^ sender, System::EventArgs^ e)//вывод инфы о врагах

```

```

{
    label1->Text = L"Очки: " + Convert::ToString(enemy1[0].score);

    pictureBox1->Controls->Add(health[0]); //чтобы фон убрать
    health[0]->BackColor = System::Drawing::Color::Transparent; //чтобы фон убрать
    health[0]->Text = L"здоровье корабля: " + Convert::ToString(enemy1[0].hp);

    pictureBox1->Controls->Add(shield[0]); //чтобы фон убрать
    shield[0]->BackColor = System::Drawing::Color::Transparent; //чтобы фон убрать
    shield[0]->Text = L"щит корабля: " + Convert::ToString(enemy1[0].shield);

    for (size_t j = 1; j < vave; j++)
    {
        pictureBox1->Controls->Add(health[j]); //чтобы фон убрать
        health[j]->BackColor = System::Drawing::Color::Transparent; //чтобы фон убрать
        health[j]->Text = L"здоровье врага: " + Convert::ToString(enemy1[j].hp);

        pictureBox1->Controls->Add(shield[j]); //чтобы фон убрать
        shield[j]->BackColor = System::Drawing::Color::Transparent; //чтобы фон убрать
        shield[j]->Text = L"щит врага: " + Convert::ToString(enemy1[j].shield);
    }
}

System::Void Project1::MyForm1::timer5_Tick(System::Object^ sender, System::EventArgs^
e)//выстрел врагов
{
    int ost dmg;
    for (size_t j = 1; j < vave; j++)
    {
        if (enemy1[j].hp > 0)
        {
            if ((abs(xtek[0] - xtek[j]) < enemy1[j].rad) || (abs(ytek[0] - ytek[j]) <
enemy1[j].rad))
            {
                laz[j] = gcnew PictureBox();
                laz[j]->BackColor = System::Drawing::Color::Transparent;
                laz[j]->Location = System::Drawing::Point(-2, -1);
                laz[j]->Size = System::Drawing::Size(1920, 1080);
                laz[j]->Name = "lazer" + name + n + j;

                if (speedaa[j] == 11)
                {
                    pictureBox1->Controls->Add(laz[j]);
                    laz[j]->BringToFront();
                    Graphics^ g = laz[j]->CreateGraphics();
                    Pen^ p = gcnew Pen(Color::Blue, 4.0f);
                    g->DrawLine(p, pictureBox2->Location.X + 50, pictureBox2->Location.Y
+ 20, pb[j]->Location.X + 20, pb[j]->Location.Y + 20);
                    speedaa[j] = 0;
                    if (enemy1[0].shield == 0)
                    {
                        if ((enemy1[0].hp - enemy1[j].dmg) < 1)
                        {
                            timer1->Enabled = false;
                            timer2->Enabled = false;
                            timer3->Enabled = false;
                            timer4->Enabled = false;
                            timer5->Enabled = false;
                            timer6->Enabled = false;
                            timer7->Enabled = false;
                            timer8->Enabled = false;
                            timer9->Enabled = false;
                            timer10->Enabled = false;
                            timer11->Enabled = false;

```

```

        timer12->Enabled = false;
        label12->Text = Convert::ToString(enemy1[0].score);
        groupBox3->Enabled = true;
        groupBox3->Location = System::Drawing::Point(588, 289);
        groupBox3->Visible = true;
    }
    else
    {
        enemy1[0].hp = enemy1[0].hp - enemy1[j].dmg;
    }
}
else
{
    if ((enemy1[0].shield - enemy1[j].dmg) < 1)
    {
        ost dmg = enemy1[j].dmg - enemy1[0].shield;
        enemy1[0].shield = 0;
        enemy1[0].hp = enemy1[0].hp - ost dmg;
    }
    else
    {
        enemy1[0].shield = enemy1[0].shield - enemy1[j].dmg;
    }
}
}
else
{
    speedaa[j] = speedaa[j] + 1;
}
}
}
pictureBox1->Controls->Clear();
pictureBox1->Controls->Add(pb[j]);
pictureBox1->Controls->Add(label1);
}
}

System::Void Project1::MyForm1::timer6_Tick(System::Object^ sender, System::EventArgs^
e)//выстрел гк
{
    if (enemy1[target].hp != 0)
    {
        int ost dmg = 0;
        if ((abs(xtek[0] - xtek[target]) < enemy1[0].rad) || (abs(ytek[0] - ytek[target])
< enemy1[0].rad))
        {
            laz[0] = gcnew PictureBox();
            laz[0]->BackColor = System::Drawing::Color::Transparent;
            laz[0]->Location = System::Drawing::Point(0, 0);
            laz[0]->Size = System::Drawing::Size(1920, 1080);
            laz[0]->Name = "PictureBox" + name;
            if (speedaa[0] == 11)
            {
                pictureBox1->Controls->Add(laz[0]);
                laz[0]->BringToFront();
                Graphics^ g = laz[0]->CreateGraphics();
                Pen^ p = gcnew Pen(Color::Red, 4.0f);
                g->DrawLine(p, pictureBox2->Location.X + 20, pictureBox2->Location.Y +
20, pb[target]->Location.X + 20, pb[target]->Location.Y + 20);
                speedaa[0] = 0;
                if (enemy1[target].shield == 0)
                {
                    if ((enemy1[target].hp - enemy1[0].dmg) < 1)
                    {

```

```

        enemy1[target].hp = 0;
        enemy1[target].dmg = 0;
        pb[target]->Visible = false;
        pb[target]->Location = System::Drawing::Point(0 + respx, 0);
        respx = respx + 100;
        pb[target]->Enabled = false;
        enemy1[0].score = enemy1[0].score + enemy1[target].score;
    }
    else
    {
        enemy1[target].hp = enemy1[target].hp - enemy1[0].dmg;
    }
}
else
{
    if ((enemy1[target].shield - enemy1[0].dmg) < 0)
    {
        ostdmg = enemy1[0].dmg - enemy1[target].shield;
        enemy1[target].shield = 0;
        if ((enemy1[target].hp - ostdmg) < 1)
        {
            enemy1[target].hp = 0;
            enemy1[target].dmg = 0;
            pb[target]->Visible = false;
            pb[target]->Location = System::Drawing::Point(0 + respx, 0);
            respx = respx + 100;
            pb[target]->Enabled = false;
            enemy1[0].score = enemy1[0].score + enemy1[target].score;
        }
        else
        {
            enemy1[target].hp = enemy1[target].hp - ostdmg;
        }
    }
    else
    {
        enemy1[target].shield = enemy1[target].shield - enemy1[0].dmg;
    }
}
}
else
{
    pictureBox1->Controls->Clear();
    pictureBox1->Controls->Add(pictureBox2);
    speedaa[0] = speedaa[0] + 1;
}
}
}
}

```

```

System::Void Project1::MyForm1::timer11_Tick(System::Object^ sender, System::EventArgs^ e)//назад
{

```

```

    if (GetAsyncKeyState(VK_ESCAPE))
    {
        timer1->Enabled = false;
        timer2->Enabled = false;
        timer3->Enabled = false;
        timer4->Enabled = false;
        timer5->Enabled = false;
        timer6->Enabled = false;
        timer7->Enabled = false;
        timer8->Enabled = false;
        timer9->Enabled = false;
    }
}

```

```

        timer10->Enabled = false;
        timer12->Enabled = false;
        groupBox2->Enabled = true;
        groupBox2->Location = System::Drawing::Point(691, 300);
        groupBox2->Visible = true;
    }
}

System::Void Project1::MyForm1::button2_Click(System::Object^ sender, System::EventArgs^ e)//продолжить игру
{
    timer1->Enabled = true;
    timer2->Enabled = true;
    timer3->Enabled = true;
    timer4->Enabled = true;
    timer5->Enabled = true;
    timer6->Enabled = true;
    timer7->Enabled = true;
    timer8->Enabled = true;
    timer9->Enabled = true;
    timer10->Enabled = true;
    timer12->Enabled = true;
    groupBox2->Visible = false;
    groupBox2->Enabled = false;
}

System::Void Project1::MyForm1::timer12_Tick(System::Object^ sender, System::EventArgs^ e)//использование скилов
{
    pictureBox4->Controls->Add(label14);
    label14->Location = System::Drawing::Point(12, 14);
    label14->BackColor = System::Drawing::Color(Color::Transparent);
    label14->BringToFront();
    if (skillkd[0] == 70)
    {
        label14->Text = "Готов";
        label14->Location = System::Drawing::Point(5, 14);
    }
    else
    {
        label14->Text = Convert::ToString((70 - skillkd[0]) / 10);
    }

    pictureBox3->Controls->Add(label13);
    label13->Location = System::Drawing::Point(12, 14);
    label13->BackColor = System::Drawing::Color(Color::Transparent);
    label13->BringToFront();
    if (skillkd[1] == 130)
    {
        label13->Text = "Готов";
        label13->Location = System::Drawing::Point(5, 14);
    }
    else
    {
        label13->Text = Convert::ToString((130 - skillkd[1]) / 10);
    }

    pictureBox5->Controls->Add(label15);
    label15->Location = System::Drawing::Point(12, 14);
    label15->BackColor = System::Drawing::Color(Color::Transparent);
    label15->BringToFront();
    if (skillkd[2] == 50)
    {
        label15->Text = "Готов";
    }
}

```

```

        label15->Location = System::Drawing::Point(5, 14);
    }
    else
    {
        label15->Text = Convert::ToString((50 - skillkd[2]) / 10);
    }

    if (skillkd[0] < 70)//7 секунд
    {
        ++skillkd[0];//уронпообласти
    }
    if (skillkd[1] < 130)//13 секунд
    {
        ++skillkd[1];//хилл
    }
    if (skillkd[2] < 50)//5 секунд
    {
        ++skillkd[2];//щит
    }
    if (GetAsyncKeyState(VK_F1))//наносит 15% хпврагов + половинауронагк. чистойурон
    {
        if ((skillkd[0] == 70) && (groupBox1->Visible == false))
        {
            skillkd[0] = 0;
            for (size_t j = 1; j < vave; j++)
            {
                int dmgskill = trunc(maxhpenemy[j] * 0.15) + trunc(enemy1[0].dmg * 0.5);
                if ((enemy1[j].hp - dmgskill) < 1)
                {
                    enemy1[j].hp = 0;
                    enemy1[j].dmg = 0;
                    pb[j]->Visible = false;
                    pb[j]->Location = System::Drawing::Point(0 + respx, 0);
                    respx = respx + 100;
                    pb[j]->Enabled = false;
                    enemy1[0].score = enemy1[0].score + enemy1[j].score;
                }
                else
                {
                    enemy1[j].hp = enemy1[j].hp - dmgskill;
                }
            }
        }
    }
    if (GetAsyncKeyState(VK_F2))//регениит 33% здоровья
    {
        if ((skillkd[1] == 130) && (groupBox1->Visible == false))
        {
            skillkd[1] = 0;
            int healh = trunc(fullxp * 0.33);
            if ((healh + enemy1[0].hp) < fullxp)
            {
                enemy1[0].hp = healh + enemy1[0].hp;
            }
            else
            {
                enemy1[0].hp = fullxp;
            }
        }
    }
    if (GetAsyncKeyState(VK_F3))//регениит 25% щита
    {
        if ((skillkd[2] == 50) && (groupBox1->Visible == false))

```

```

    {
        skillkd[2] = 0;
        int healsh = trunc(maxregshield[0] * 0.25);
        if ((healsh + enemy1[0].shield) < maxregshield[0])
        {
            enemy1[0].shield = healsh + enemy1[0].shield;
        }
        else
        {
            enemy1[0].shield = maxregshield[0];
        }
    }
}

System::Void Project1::MyForm1::button3_Click(System::Object^ sender, System::EventArgs^ e)//выход из игры в главное меню
{
    delete this;
}

System::Void Project1::MyForm1::button1_Click(System::Object^ sender, System::EventArgs^ e)//подтверждение выбора очков характеристик
{
    if (point != numpoint)
    {
        MessageBox::Show(this, "Пожалуйста, потратьте ровно 4 очка характеристик",
"Сообщение", MessageBoxButtons::OK, MessageBoxIcon::Information);
    }
    else
    {
        numericUpDown1->Minimum = numericUpDown1->Value;
        numericUpDown2->Minimum = numericUpDown2->Value;
        numericUpDown3->Minimum = numericUpDown3->Value;
        numericUpDown4->Minimum = numericUpDown4->Value;
        numericUpDown5->Minimum = numericUpDown5->Value;
        numericUpDown6->Minimum = numericUpDown6->Value;

        enemy1[0].hp = Convert::ToInt64(numericUpDown1->Value);
        fullxp = enemy1[0].hp;
        enemy1[0].shield = Convert::ToInt64(numericUpDown2->Value);
        maxregshield[0] = enemy1[0].shield;
        enemy1[0].dmg = Convert::ToInt64(numericUpDown3->Value);
        enemy1[0].rad = Convert::ToInt64(numericUpDown4->Value);
        enemy1[0].regshield = Convert::ToInt64(numericUpDown5->Value);
        enemy1[0].speed = Convert::ToInt64(numericUpDown6->Value);

        numericUpDown1->Maximum = enemy1[0].hp + (numpoint * numericUpDown1->Increment);
        numericUpDown2->Maximum = enemy1[0].shield + (numpoint * numericUpDown2->Increment);
        numericUpDown3->Maximum = enemy1[0].dmg + (numpoint * numericUpDown3->Increment);
        numericUpDown4->Maximum = enemy1[0].rad + (numpoint * numericUpDown4->Increment);
        numericUpDown5->Maximum = enemy1[0].regshield + (numpoint * numericUpDown5->Increment);
        numericUpDown6->Maximum = enemy1[0].speed + (numpoint * numericUpDown6->Increment);

        groupBox1->Enabled = false;
        groupBox1->Visible = false;
        timer1->Enabled = true;
        timer2->Enabled = true;
        numvave = numvave + 1;
    }
}

```

```

System::Void Project1::MyForm1::timer4_Tick(System::Object^ sender, System::EventArgs^
e)//перемещение врагов к главному кораблю
{
    double fDistance[100];
    double fRadians[100];
    double fDiffX[100];
    double fDiffY[100];
    int iDiffX[100];
    int iDiffY[100];

    if (stenemy == true)
    {
        for (size_t j = 1; j < vave; j++)
        {
            xtek[j] = 100 * (j + 1);
            ytek[j] = 0;
        }
        stenemy = false;
    }

    for (size_t o = 1; o < vave; o++)
    {
        if (enemy1[o].hp > 0)
        {
            if ((abs(xtek[0]-xtek[o]) > enemy1[o].rad) || (abs(ytek[0] - ytek[o]) >
enemy1[o].rad))
            {

                pictureBox1->Controls->Add(pb[o]); //чтобы фон убрать
                pb[o]->BackColor = System::Drawing::Color::Transparent;//чтобы фон убрать
                GetCursorPos(&p[o]);

                pp[o].x = pb[o]->Location.X;
                pp[o].y = pb[o]->Location.Y;

                int iSpeed = enemy1[o].speed; // скорость
                //////////////////////////////////////////*****

                // дистанция между курсором и pictureBox1 (главный корабль) (в пикселях)
                fDistance[o] = sqrt((p[o].x - pp[o].x) * (p[o].x - pp[o].x) + (p[o].y - pp[o].y) *
                (p[o].y - pp[o].y));
                // если pictureBox1 (главный корабль) рядом с курсором, то выход
                (чтобы не дёргался)
                if (fDistance[o] < iSpeed) return;

                // определяем угол в радианах между катетами, образованными позицией курсора и позицией
                pictureBox1 (главный корабль)
                fRadians[o] = 0;
                if (p[o].x != pp[o].x)
                {
                    fRadians[o] = atan(1.0 * abs(p[o].y - pp[o].y) / abs(p[o].x -
                pp[o].x));
                }

                // определяем, на сколько нужно изменить позицию pictureBox1 (главный
                корабль) по осям
                fDiffX[o] = fabs(iSpeed * cos(fRadians[o]));
                fDiffY[o] = fabs(iSpeed * sin(fRadians[o]));

                // преобразование fDiffX, fDiffY в целые числа
                iDiffX[o] = fDiffX[o];
                iDiffY[o] = fDiffY[o];
            }
        }
    }
}

```

```
// изменение позиции pictureBox1 (главный корабль)
if (pp[o].x < p[o].x) xtek[o] += iDiffX[o];
    else if (pp[o].x > p[o].x) xtek[o] -= iDiffX[o];
    if (pp[o].y < p[o].y) ytek[o] += iDiffY[o];
    else if (pp[o].y > p[o].y) ytek[o] -= iDiffY[o];
    pb[o]->Location = System::Drawing::Point(xtek[o], ytek[o]);
    }
    }
}
```

MyForm.cpp

```
#include "MyForm.h"
#include "MyForm1.h"
#include <Windows.h>

using namespace System;
using namespace System::Windows::Forms;
using namespace Project1;
using namespace System::IO;

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Application::Run(gcnew MyForm);
    return 0;
}

System::Void Project1::MyForm::button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    MyForm1^ f2 = gcnew MyForm1();
    f2->Show();
}

System::Void Project1::MyForm::button2_Click(System::Object^ sender, System::EventArgs^ e) // чтение из файла
{
    String^ filename = "C:\\программ для программирования\\praktika\\records.txt";

    try
    {
        StreamReader^ file = File::OpenText(filename);
        groupBox1->Visible = true;
        groupBox1->Enabled = true;
        textBox1->Text = file->ReadToEnd();
        file->Close();
    }
    catch (Exception^ e)
    {
        MessageBox::Show(this, "Файл не был открыт", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
    delete filename;
}

System::Void Project1::MyForm::button4_Click(System::Object^ sender, System::EventArgs^ e)
{
    groupBox1->Visible = false;
    groupBox1->Enabled = false;
}

System::Void Project1::MyForm::выводИнформацииОбАвтореИОИграToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
{
    MessageBox::Show(this, "Данная игра была разработана студентом группы 21В01 Баженовым Д.С.", "Сообщение",
    MessageBoxButtons::OK, MessageBoxIcon::Information);
}
```

```

}

System::Void Project1::MyForm::button3_Click(System::Object^ sender, System::EventArgs^ e)
{
    this->Close();
}

```

MyForm1.h

```

#pragma once

namespace Project1 {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Сводка для MyForm1
    /// </summary>
    public ref class MyForm1 : public System::Windows::Forms::Form
    {
    public:
        MyForm1(void)
        {
            InitializeComponent();
            //
            //TODO: добавьте код конструктора
            //
        }

    protected:
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        ~MyForm1()
        {
            if (components)
            {
                delete components;
            }
        }

    private: System::Windows::Forms::PictureBox^ pictureBox1;
    private: System::Windows::Forms::Timer^ timer1;
    private: System::Windows::Forms::PictureBox^ pictureBox2;
    private: System::Windows::Forms::Timer^ timer2;
    private: System::Windows::Forms::Timer^ timer3;
    private: System::Windows::Forms::Timer^ timer4;
    private: System::Windows::Forms::Timer^ timer5;
    private: System::Windows::Forms::Button^ button1;
    private: System::Windows::Forms::Timer^ timer6;
    private: System::Windows::Forms::Timer^ timer7;
    private: System::Windows::Forms::Timer^ timer8;
    private: System::Windows::Forms::Timer^ timer9;
    private: System::Windows::Forms::Label^ label1;
    private: System::Windows::Forms::GroupBox^ groupBox1;
    private: System::Windows::Forms::Button^ button2;
    private: System::Windows::Forms::NumericUpDown^ numericUpDown4;
    private: System::Windows::Forms::NumericUpDown^ numericUpDown3;
    private: System::Windows::Forms::NumericUpDown^ numericUpDown2;

```

```

private: System::Windows::Forms::NumericUpDown^ numericUpDown1;
private: System::Windows::Forms::GroupBox^ groupBox2;
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::NumericUpDown^ numericUpDown6;
private: System::Windows::Forms::NumericUpDown^ numericUpDown5;
private: System::Windows::Forms::Label^ label7;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::Label^ label9;
private: System::Windows::Forms::Timer^ timer10;
private: System::Windows::Forms::Timer^ timer11;
private: System::Windows::Forms::Timer^ timer12;
private: System::Windows::Forms::GroupBox^ groupBox3;
private: System::Windows::Forms::Button^ button5;
private: System::Windows::Forms::Button^ button4;
private: System::Windows::Forms::Label^ label11;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label10;
private: System::Windows::Forms::Label^ label12;
private: System::Windows::Forms::PictureBox^ pictureBox3;
private: System::Windows::Forms::PictureBox^ pictureBox4;
private: System::Windows::Forms::PictureBox^ pictureBox5;
private: System::Windows::Forms::Label^ label13;
private: System::Windows::Forms::Label^ label14;
private: System::Windows::Forms::Label^ label15;

private: System::ComponentModel::IContainer^ components;
protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора – не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        this->components = (gcnew System::ComponentModel::Container());
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(MyForm1::typeid));
        this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
        this->timer1 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());
        this->timer2 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer3 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer4 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer5 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->button1 = (gcnew System::Windows::Forms::Button());

```

```

        this->timer6 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer7 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer8 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer9 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
        this->label9 = (gcnew System::Windows::Forms::Label());
        this->label8 = (gcnew System::Windows::Forms::Label());
        this->label7 = (gcnew System::Windows::Forms::Label());
        this->label6 = (gcnew System::Windows::Forms::Label());
        this->label5 = (gcnew System::Windows::Forms::Label());
        this->label4 = (gcnew System::Windows::Forms::Label());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->numericUpDown6 = (gcnew
System::Windows::Forms::NumericUpDown());
        this->numericUpDown5 = (gcnew
System::Windows::Forms::NumericUpDown());
        this->numericUpDown4 = (gcnew
System::Windows::Forms::NumericUpDown());
        this->numericUpDown3 = (gcnew
System::Windows::Forms::NumericUpDown());
        this->numericUpDown2 = (gcnew
System::Windows::Forms::NumericUpDown());
        this->numericUpDown1 = (gcnew
System::Windows::Forms::NumericUpDown());
        this->button2 = (gcnew System::Windows::Forms::Button());
        this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());
        this->button3 = (gcnew System::Windows::Forms::Button());
        this->timer10 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer11 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->timer12 = (gcnew System::Windows::Forms::Timer(this-
>components));
        this->groupBox3 = (gcnew System::Windows::Forms::GroupBox());
        this->label12 = (gcnew System::Windows::Forms::Label());
        this->button5 = (gcnew System::Windows::Forms::Button());
        this->button4 = (gcnew System::Windows::Forms::Button());
        this->label11 = (gcnew System::Windows::Forms::Label());
        this->textBox1 = (gcnew System::Windows::Forms::TextBox());
        this->label10 = (gcnew System::Windows::Forms::Label());
        this->pictureBox3 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox4 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox5 = (gcnew System::Windows::Forms::PictureBox());
        this->label13 = (gcnew System::Windows::Forms::Label());
        this->label14 = (gcnew System::Windows::Forms::Label());
        this->label15 = (gcnew System::Windows::Forms::Label());
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox2))->BeginInit();
        this->groupBox1->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown6))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown5))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown4))->BeginInit();

```

```

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown3))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown2))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown1))->BeginInit();
        this->groupBox2->SuspendLayout();
        this->groupBox3->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox3))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox4))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox5))->BeginInit();
        this->SuspendLayout();
        //
        // pictureBox1
        //
        this->pictureBox1->Image =
(cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox1.Image")));
        this->pictureBox1->Location = System::Drawing::Point(12, 1);
        this->pictureBox1->Name = L"pictureBox1";
        this->pictureBox1->Size = System::Drawing::Size(1920, 1080);
        this->pictureBox1->TabIndex = 0;
        this->pictureBox1->TabStop = false;
        //
        // timer1
        //
        this->timer1->Enabled = true;
        this->timer1->Interval = 5;
        this->timer1->Tick += gcnew System::EventHandler(this,
&MyForm1::timer1_Tick);
        //
        // pictureBox2
        //
        this->pictureBox2->BackColor = System::Drawing::Color::Transparent;
        this->pictureBox2->Image =
(cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox2.Image")));
        this->pictureBox2->Location = System::Drawing::Point(625, 325);
        this->pictureBox2->Name = L"pictureBox2";
        this->pictureBox2->Size = System::Drawing::Size(100, 41);
        this->pictureBox2->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::AutoSize;
        this->pictureBox2->TabIndex = 1;
        this->pictureBox2->TabStop = false;
        //
        // timer2
        //
        this->timer2->Enabled = true;
        this->timer2->Tick += gcnew System::EventHandler(this,
&MyForm1::timer2_Tick);
        //
        // timer3
        //
        this->timer3->Tick += gcnew System::EventHandler(this,
&MyForm1::timer3_Tick);
        //
        // timer4
        //
        this->timer4->Interval = 30;
        this->timer4->Tick += gcnew System::EventHandler(this,
&MyForm1::timer4_Tick);
        //
        // timer5

```

```

        //
        this->timer5->Tick += gcnew System::EventHandler(this,
&MyForm1::timer5_Tick);
        //
        // button1
        //
        this->button1->BackColor = System::Drawing::Color::DarkGray;
        this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button1->Location = System::Drawing::Point(568, 430);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(190, 63);
        this->button1->TabIndex = 2;
        this->button1->Text = L"Подтвердить";
        this->button1->UseVisualStyleBackColor = false;
        this->button1->Click += gcnew System::EventHandler(this,
&MyForm1::button1_Click);
        //
        // timer6
        //
        this->timer6->Tick += gcnew System::EventHandler(this,
&MyForm1::timer6_Tick);
        //
        // timer7
        //
        this->timer7->Interval = 30;
        this->timer7->Tick += gcnew System::EventHandler(this,
&MyForm1::timer7_Tick);
        //
        // timer8
        //
        this->timer8->Tick += gcnew System::EventHandler(this,
&MyForm1::timer8_Tick);
        //
        // timer9
        //
        this->timer9->Interval = 1100;
        this->timer9->Tick += gcnew System::EventHandler(this,
&MyForm1::timer9_Tick);
        //
        // label1
        //
        this->label1->BackColor = System::Drawing::Color::Transparent;
        this->label1->ForeColor = System::Drawing::Color::White;
        this->label1->Location = System::Drawing::Point(1700, 80);
        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(100, 13);
        this->label1->TabIndex = 3;
        this->label1->Text = L"Очки: 0";
        //
        // groupBox1
        //
        this->groupBox1->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"groupBox1.BackgroundImage")));
        this->groupBox1->Controls->Add(this->label9);
        this->groupBox1->Controls->Add(this->label8);
        this->groupBox1->Controls->Add(this->label7);
        this->groupBox1->Controls->Add(this->label6);
        this->groupBox1->Controls->Add(this->label5);
        this->groupBox1->Controls->Add(this->label4);
        this->groupBox1->Controls->Add(this->label3);

```

```

this->groupBox1->Controls->Add(this->label2);
this->groupBox1->Controls->Add(this->numericUpDown6);
this->groupBox1->Controls->Add(this->numericUpDown5);
this->groupBox1->Controls->Add(this->numericUpDown4);
this->groupBox1->Controls->Add(this->numericUpDown3);
this->groupBox1->Controls->Add(this->numericUpDown2);
this->groupBox1->Controls->Add(this->numericUpDown1);
this->groupBox1->Controls->Add(this->button1);
this->groupBox1->Enabled = false;
this->groupBox1->Location = System::Drawing::Point(182, 221);
this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(764, 499);
this->groupBox1->TabIndex = 4;
this->groupBox1->TabStop = false;
this->groupBox1->Visible = false;
//
// label9
//
this->label9->AutoSize = true;
this->label9->Font = (gcnew System::Drawing::Font(L"Perpetua Titling
MT", 48, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label9->Location = System::Drawing::Point(660, 103);
this->label9->Name = L"label9";
this->label9->Size = System::Drawing::Size(73, 75);
this->label9->TabIndex = 16;
this->label9->Text = L"4";
//
// label8
//
this->label8->AutoSize = true;
this->label8->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label8->Location = System::Drawing::Point(437, 41);
this->label8->Name = L"label8";
this->label8->Size = System::Drawing::Size(296, 62);
this->label8->TabIndex = 15;
this->label8->Text = L"Подними самооценку!\r\nПотрать меня! ";
//
// label7
//
this->label7->AutoSize = true;
this->label7->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label7->Location = System::Drawing::Point(217, 331);
this->label7->Name = L"label7";
this->label7->Size = System::Drawing::Size(158, 20);
this->label7->TabIndex = 14;
this->label7->Text = L"скорость движения";
//
// label6
//
this->label6->AutoSize = true;
this->label6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label6->Location = System::Drawing::Point(217, 275);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(277, 20);
this->label6->TabIndex = 13;
this->label6->Text = L"количество регенерируемого щита";

```

```

//
// label5
//
this->label5->AutoSize = true;
this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label5->Location = System::Drawing::Point(217, 229);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(110, 20);
this->label5->TabIndex = 12;
this->label5->Text = L"радиус атаки";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label4->Location = System::Drawing::Point(217, 179);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(43, 20);
this->label4->TabIndex = 11;
this->label4->Text = L"урон";
//
// label3
//
this->label3->AutoSize = true;
this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label3->Location = System::Drawing::Point(217, 130);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(142, 20);
this->label3->TabIndex = 10;
this->label3->Text = L"количество щита";
//
// label2
//
this->label2->AutoSize = true;
this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label2->Location = System::Drawing::Point(217, 79);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(175, 20);
this->label2->TabIndex = 9;
this->label2->Text = L"количество здоровья";
//
// numericUpDown6
//
this->numericUpDown6->BackColor = System::Drawing::Color::Yellow;
this->numericUpDown6->Location = System::Drawing::Point(62, 334);
this->numericUpDown6->Name = L"numericUpDown6";
this->numericUpDown6->Size = System::Drawing::Size(120, 20);
this->numericUpDown6->TabIndex = 8;
//
// numericUpDown5
//
this->numericUpDown5->BackColor = System::Drawing::Color::DarkOrchid;
this->numericUpDown5->Location = System::Drawing::Point(62, 278);
this->numericUpDown5->Name = L"numericUpDown5";
this->numericUpDown5->Size = System::Drawing::Size(120, 20);
this->numericUpDown5->TabIndex = 7;

```

```

//
// numericUpDown4
//
this->numericUpDown4->BackColor = System::Drawing::Color::DarkOrange;
this->numericUpDown4->Location = System::Drawing::Point(62, 229);
this->numericUpDown4->Name = L"numericUpDown4";
this->numericUpDown4->Size = System::Drawing::Size(120, 20);
this->numericUpDown4->TabIndex = 6;
//
// numericUpDown3
//
this->numericUpDown3->BackColor = System::Drawing::Color::Brown;
this->numericUpDown3->Location = System::Drawing::Point(62, 179);
this->numericUpDown3->Name = L"numericUpDown3";
this->numericUpDown3->Size = System::Drawing::Size(120, 20);
this->numericUpDown3->TabIndex = 5;
//
// numericUpDown2
//
this->numericUpDown2->BackColor =
System::Drawing::Color::LightSkyBlue;
this->numericUpDown2->Location = System::Drawing::Point(62, 130);
this->numericUpDown2->Name = L"numericUpDown2";
this->numericUpDown2->Size = System::Drawing::Size(120, 20);
this->numericUpDown2->TabIndex = 4;
//
// numericUpDown1
//
this->numericUpDown1->BackColor = System::Drawing::Color::LimeGreen;
this->numericUpDown1->Location = System::Drawing::Point(62, 79);
this->numericUpDown1->Name = L"numericUpDown1";
this->numericUpDown1->Size = System::Drawing::Size(120, 20);
this->numericUpDown1->TabIndex = 3;
//
// button2
//
this->button2->BackColor = System::Drawing::Color::DarkGray;
this->button2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button2->Location = System::Drawing::Point(142, 149);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(245, 75);
this->button2->TabIndex = 3;
this->button2->Text = L"Продолжить ";
this->button2->UseVisualStyleBackColor = false;
this->button2->Click += gcnew System::EventHandler(this,
&MyForm1::button2_Click);
//
// groupBox2
//
this->groupBox2->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"groupBox2.BackgroundImage")));
this->groupBox2->Controls->Add(this->button3);
this->groupBox2->Controls->Add(this->button2);
this->groupBox2->Enabled = false;
this->groupBox2->Location = System::Drawing::Point(1048, 12);
this->groupBox2->Name = L"groupBox2";
this->groupBox2->Size = System::Drawing::Size(515, 404);
this->groupBox2->TabIndex = 5;
this->groupBox2->TabStop = false;
this->groupBox2->Visible = false;

```

```

        //
        // button3
        //
        this->button3->BackColor = System::Drawing::Color::DarkGray;
        this->button3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button3->Location = System::Drawing::Point(142, 250);
        this->button3->Name = L"button3";
        this->button3->Size = System::Drawing::Size(245, 75);
        this->button3->TabIndex = 4;
        this->button3->Text = L"Вернуться в главное меню";
        this->button3->UseVisualStyleBackColor = false;
        this->button3->Click += gcnew System::EventHandler(this,
&MyForm1::button3_Click);
        //
        // timer10
        //
        this->timer10->Interval = 10;
        this->timer10->Tick += gcnew System::EventHandler(this,
&MyForm1::timer10_Tick);
        //
        // timer11
        //
        this->timer11->Enabled = true;
        this->timer11->Tick += gcnew System::EventHandler(this,
&MyForm1::timer11_Tick);
        //
        // timer12
        //
        this->timer12->Tick += gcnew System::EventHandler(this,
&MyForm1::timer12_Tick);
        //
        // groupBox3
        //
        this->groupBox3->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"groupBox3.BackgroundImage")));
        this->groupBox3->Controls->Add(this->label12);
        this->groupBox3->Controls->Add(this->button5);
        this->groupBox3->Controls->Add(this->button4);
        this->groupBox3->Controls->Add(this->label11);
        this->groupBox3->Controls->Add(this->textBox1);
        this->groupBox3->Controls->Add(this->label10);
        this->groupBox3->Enabled = false;
        this->groupBox3->Location = System::Drawing::Point(1022, 473);
        this->groupBox3->Name = L"groupBox3";
        this->groupBox3->Size = System::Drawing::Size(715, 556);
        this->groupBox3->TabIndex = 6;
        this->groupBox3->TabStop = false;
        this->groupBox3->Visible = false;
        //
        // label12
        //
        this->label12->AutoSize = true;
        this->label12->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 24, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label12->Location = System::Drawing::Point(314, 168);
        this->label12->Name = L"label12";
        this->label12->Size = System::Drawing::Size(118, 37);
        this->label12->TabIndex = 5;
        this->label12->Text = L"label12";

```

```

//
// button5
//
this->button5->BackColor = System::Drawing::Color::DarkGray;
this->button5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button5->Location = System::Drawing::Point(211, 404);
this->button5->Name = L"button5";
this->button5->Size = System::Drawing::Size(300, 104);
this->button5->TabIndex = 4;
this->button5->Text = L"Вернуться в главное меню";
this->button5->UseVisualStyleBackColor = false;
this->button5->Click += gcnew System::EventHandler(this,
&MyForm1::button5_Click);
//
// button4
//
this->button4->BackColor = System::Drawing::Color::DarkGray;
this->button4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button4->Location = System::Drawing::Point(305, 338);
this->button4->Name = L"button4";
this->button4->Size = System::Drawing::Size(100, 42);
this->button4->TabIndex = 3;
this->button4->Text = L"Ввод";
this->button4->UseVisualStyleBackColor = false;
this->button4->Click += gcnew System::EventHandler(this,
&MyForm1::button4_Click);
//
// label11
//
this->label11->AutoSize = true;
this->label11->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label11->Location = System::Drawing::Point(222, 236);
this->label11->Name = L"label11";
this->label11->Size = System::Drawing::Size(262, 31);
this->label11->TabIndex = 2;
this->label11->Text = L"введите имя игрока";
//
// textBox1
//
this->textBox1->BackColor = System::Drawing::Color::DarkGray;
this->textBox1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 20.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->textBox1->Location = System::Drawing::Point(273, 294);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(159, 38);
this->textBox1->TabIndex = 1;
//
// label10
//
this->label10->AutoSize = true;
this->label10->Font = (gcnew System::Drawing::Font(L"Comic Sans MS",
72, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));

```

```

        this->label10->Location = System::Drawing::Point(79, 16);
        this->label10->Name = L"label10";
        this->label10->Size = System::Drawing::Size(569, 135);
        this->label10->TabIndex = 0;
        this->label10->Text = L"поражение";
        //
        // pictureBox3
        //
        this->pictureBox3->Image =
(cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox3.Image"))));
        this->pictureBox3->Location = System::Drawing::Point(1789, 988);
        this->pictureBox3->Name = L"pictureBox3";
        this->pictureBox3->Size = System::Drawing::Size(41, 41);
        this->pictureBox3->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::AutoSize;
        this->pictureBox3->TabIndex = 7;
        this->pictureBox3->TabStop = false;
        //
        // pictureBox4
        //
        this->pictureBox4->Image =
(cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox4.Image"))));
        this->pictureBox4->Location = System::Drawing::Point(1727, 988);
        this->pictureBox4->Name = L"pictureBox4";
        this->pictureBox4->Size = System::Drawing::Size(42, 44);
        this->pictureBox4->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::AutoSize;
        this->pictureBox4->TabIndex = 8;
        this->pictureBox4->TabStop = false;
        //
        // pictureBox5
        //
        this->pictureBox5->Image =
(cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox5.Image"))));
        this->pictureBox5->Location = System::Drawing::Point(1848, 988);
        this->pictureBox5->Name = L"pictureBox5";
        this->pictureBox5->Size = System::Drawing::Size(44, 44);
        this->pictureBox5->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::AutoSize;
        this->pictureBox5->TabIndex = 9;
        this->pictureBox5->TabStop = false;
        //
        // label13
        //
        this->label13->AutoSize = true;
        this->label13->ForeColor = System::Drawing::Color::Fuchsia;
        this->label13->Location = System::Drawing::Point(1801, 1006);
        this->label13->Name = L"label13";
        this->label13->Size = System::Drawing::Size(13, 13);
        this->label13->TabIndex = 10;
        this->label13->Text = L"0";
        //
        // label14
        //
        this->label14->AutoSize = true;
        this->label14->ForeColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(0))
, static_cast<System::Int32>(static_cast<System::Byte>(192))),
        static_cast<System::Int32>(static_cast<System::Byte>(192)));
        this->label14->Location = System::Drawing::Point(1724, 1006);
        this->label14->Name = L"label14";
        this->label14->Size = System::Drawing::Size(13, 13);
        this->label14->TabIndex = 11;
        this->label14->Text = L"0";

```

```

//
// label15
//
this->label15->AutoSize = true;
this->label15->ForeColor = System::Drawing::Color::Fuchsia;
this->label15->Location = System::Drawing::Point(1851, 1006);
this->label15->Name = L"label15";
this->label15->Size = System::Drawing::Size(13, 13);
this->label15->TabIndex = 12;
this->label15->Text = L"0";
//
// MyForm1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(1904, 1041);
this->Controls->Add(this->label15);
this->Controls->Add(this->label14);
this->Controls->Add(this->groupBox3);
this->Controls->Add(this->label13);
this->Controls->Add(this->pictureBox5);
this->Controls->Add(this->pictureBox4);
this->Controls->Add(this->pictureBox3);
this->Controls->Add(this->groupBox2);
this->Controls->Add(this->groupBox1);
this->Controls->Add(this->label1);
this->Controls->Add(this->pictureBox2);
this->Controls->Add(this->pictureBox1);
this->Icon = (cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
this->Name = L"MyForm1";
this->Text = L"MyForm1";
this->Load += gcnew System::EventHandler(this,
&MyForm1::MyForm1_Load);
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox2))->EndInit();
this->groupBox1->ResumeLayout(false);
this->groupBox1->PerformLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown6))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown5))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown4))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown3))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown2))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>numericUpDown1))->EndInit();
this->groupBox2->ResumeLayout(false);
this->groupBox3->ResumeLayout(false);
this->groupBox3->PerformLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox3))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox4))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox5))->EndInit();
this->ResumeLayout(false);
this->PerformLayout();

```

```

    }
#pragma endregion
    const int n = 100; //
    int numvave = 1;
    int vave = 1; // волны
    int name = 100; // номер в названии объекта
    array<PictureBox^>^ pb = gcnew array<PictureBox^>(n);
    array<PictureBox^>^ laz = gcnew array<PictureBox^>(n);
    array<Label^>^ health = gcnew array<Label^>(n);
    array<Label^>^ shield = gcnew array<Label^>(n);
    String^ heroname = gcnew String(""); // имя игрока

private: System::Void timer1_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer2_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer3_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer4_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer5_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer6_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer7_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer8_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer9_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void MyForm1_Load(System::Object^ sender, System::EventArgs^ e);
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer10_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer11_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void timer12_Tick(System::Object^ sender, System::EventArgs^ e);
private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e);
private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e);

};
}

```

MyForm.h

```

#pragma once

namespace Project1 {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

/// <summary>
/// Сводка для MyForm
/// </summary>
public ref class MyForm : public System::Windows::Forms::Form
{
public:
    MyForm(void)
    {
        InitializeComponent();
        //
        // TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.

```

```

    /// </summary>
    ~MyForm()
    {
        if (components)
        {
            delete components;
        }
    }

private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::GroupBox^ groupBox1;
private: System::Windows::Forms::Button^ button4;
private: System::Windows::Forms::MenuStrip^ menuStrip1;
private: System::Windows::Forms::ToolStripMenuItem^
выводИнформацииОбАвтореИОИгреToolStripMenuItem;

protected:

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(MyForm::typeid));
        this->button1 = (gcnew System::Windows::Forms::Button());
        this->button2 = (gcnew System::Windows::Forms::Button());
        this->button3 = (gcnew System::Windows::Forms::Button());
        this->textBox1 = (gcnew System::Windows::Forms::TextBox());
        this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
        this->button4 = (gcnew System::Windows::Forms::Button());
        this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
        this->выводИнформацииОбАвтореИОИгреToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->groupBox1->SuspendLayout();
        this->menuStrip1->SuspendLayout();
        this->SuspendLayout();
        //
        // button1
        //
        this->button1->BackColor = System::Drawing::Color::DarkGray;
        this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 20.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button1->Location = System::Drawing::Point(570, 143);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(214, 99);
        this->button1->TabIndex = 0;
        this->button1->Text = L"НачалоИгры";
        this->button1->UseVisualStyleBackColor = false;
        this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1_Click);
        //
        // button2

```

```

//
this->button2->BackColor = System::Drawing::Color::DarkGray;
this->button2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 20.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button2->Location = System::Drawing::Point(570, 275);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(214, 101);
this->button2->TabIndex = 1;
this->button2->Text = L"Показатърекорды";
this->button2->UseVisualStyleBackColor = false;
this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2_Click);
//
// button3
//
this->button3->BackColor = System::Drawing::Color::DarkGray;
this->button3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 20.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button3->Location = System::Drawing::Point(570, 424);
this->button3->Name = L"button3";
this->button3->Size = System::Drawing::Size(214, 101);
this->button3->TabIndex = 2;
this->button3->Text = L"Выходизигры";
this->button3->UseVisualStyleBackColor = false;
this->button3->Click += gcnew System::EventHandler(this, &MyForm::button3_Click);
//
// textBox1
//
this->textBox1->BackColor = System::Drawing::Color::LightGray;
this->textBox1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 12,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->textBox1->Location = System::Drawing::Point(6, 124);
this->textBox1->Multiline = true;
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(379, 406);
this->textBox1->TabIndex = 3;
//
// groupBox1
//
this->groupBox1->BackgroundImage = (cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"groupBox1.BackgroundImage")));
this->groupBox1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
this->groupBox1->Controls->Add(this->button4);
this->groupBox1->Controls->Add(this->textBox1);
this->groupBox1->Enabled = false;
this->groupBox1->Location = System::Drawing::Point(508, 105);
this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(392, 593);
this->groupBox1->TabIndex = 4;
this->groupBox1->TabStop = false;
this->groupBox1->Visible = false;
//
// button4
//
this->button4->BackColor = System::Drawing::Color::DarkGray;
this->button4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 20.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button4->Location = System::Drawing::Point(6, 547);
this->button4->Name = L"button4";
this->button4->Size = System::Drawing::Size(379, 40);
this->button4->TabIndex = 4;

```

```

        this->button4->Text = L"OK";
        this->button4->VisualStyleBackColor = false;
        this->button4->Click += gcnew System::EventHandler(this, &MyForm::button4_Click);
        //
        // menuStrip1
        //
        this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) { this->выводИнформацииОбАвтореИОИгpeToolStripMenuItem });
        this->menuStrip1->Location = System::Drawing::Point(0, 0);
        this->menuStrip1->Name = L"menuStrip1";
        this->menuStrip1->Size = System::Drawing::Size(1344, 24);
        this->menuStrip1->TabIndex = 5;
        this->menuStrip1->Text = L"menuStrip1";
        //
        // выводИнформацииОбАвтореИОИгpeToolStripMenuItem
        //
        this->выводИнформацииОбАвтореИОИгpeToolStripMenuItem->Font = (gcnew
System::Drawing::Font(L"Segoe UI", 9, System::Drawing::FontStyle::Regular,
        System::Drawing::GraphicsUnit::Point, static_cast<System::Byte>(204)));
        this->выводИнформацииОбАвтореИОИгpeToolStripMenuItem->Name =
L"выводИнформацииОбАвтореИОИгpeToolStripMenuItem";
        this->выводИнформацииОбАвтореИОИгpeToolStripMenuItem->Size =
System::Drawing::Size(187, 20);
        this->выводИнформацииОбАвтореИОИгpeToolStripMenuItem->Text = L"Вывод
информации об авторе";
        this->выводИнформацииОбАвтореИОИгpeToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm::выводИнформацииОбАвтореИОИгpeToolStripMenuItem_Click);
        //
        // MyForm
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->BackgroundImage = (cli::safe_cast<System::Drawing::Image^>(resources->
>GetObject(L"$this.BackgroundImage")));
        this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(1344, 673);
        this->Controls->Add(this->groupBox1);
        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->menuStrip1);
        this->DoubleBuffered = true;
        this->Icon = (cli::safe_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));
        this->MainMenuStrip = this->menuStrip1;
        this->Name = L"MyForm";
        this->Text = L"MyForm";
        this->groupBox1->ResumeLayout(false);
        this->groupBox1->PerformLayout();
        this->menuStrip1->ResumeLayout(false);
        this->menuStrip1->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();

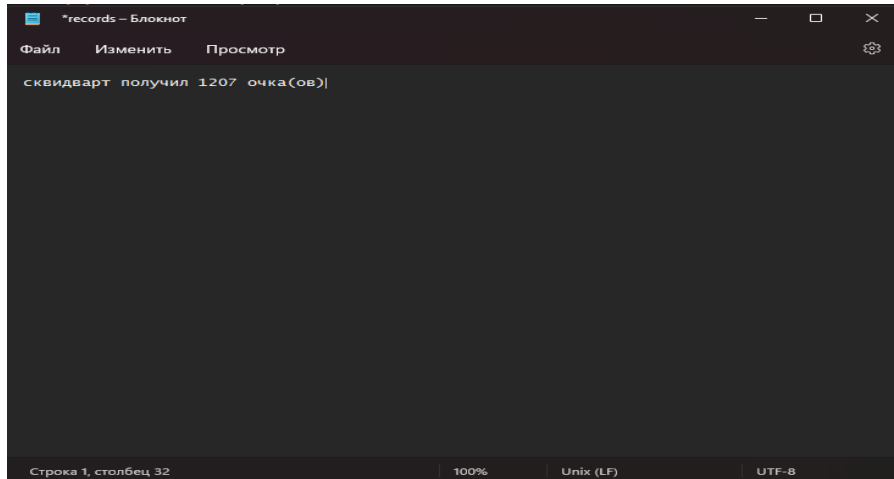
    }

#pragma endregion
    private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e);
    private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e);
    private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e);
    private: System::Void выводИнформацииОбАвтореИОИгpeToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e);
    private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e);
};
}

```

Приложение В. Файлы

Приложение В.1 Начальная заставка



Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по курсу «Программирование»
на тему «Разработка программы сложной структуры
методом нисходящего программирования»

28.12.21
ОПМ/Ке

Выполнил: Баженов Д.С.
Студент группы 21ВО1
Принял:
Патунин Д.В.

Пенза, 2022

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники
Кафедра «Вычислительная техника»

«УТВЕРЖДАЮ»

Зав. кафедрой ВТ

Митрохин М. А.

«___» _____ 2022 г

ЗАДАНИЕ
на курсовое проектирование по курсу

Программирование

Студенту Баженову Денису Сергеевичу

Группа 21ВО1

Тема проекта: Разработка программы сложной структуры
методом нисходящего программирования

Исходные данные (технические требования) на проектирование
Наименование программы: Игра «GalaWar».

Задание. Программа должна выполнять следующие функции:

- Выход из игры.
- Начало игры.
- Запись рекордов игроков в файл.
- Вывод рекордов игроков из файла.
- Вывод информации об авторе и игре.

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных — списков, структур, файлов.
3. Режим работы видеосистемы — текстовый/графический.
4. Устройство ввода информации — клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система — Windows.
8. Система программирования — C/C++ на базе среды Microsoft Visual Studio 2022.
9. Язык программирования — C++.

Объем работы по курсу
1. Расчетная часть
Разработка программы.

2. Графическая часть
Структура модулей программы, схема алгоритма программы.

3. Экспериментальная часть
Отладка программы.

Срок выполнения проекта по разделам

1. В соответствии с графиком выполнения курсового проекта.
- 2.
- 3.

Дата выдачи задания «1» октября 2022

Дата защиты проекта «28» 12 2022

Руководитель

Задание получил

Студент

Баженков Денис Сергеевич

Введение

1. Постановка задачи.

2. Выбор решения

3. Описание разработк

4. Отладка и тестиров

5. Описание програм

5.1 Описание первой

5.2 Описание второй

6. Руководство поль

Заключение

Список используем

Приложение А. Ли

Приложение В. Ф