МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

Отчет по индивидуальному заданию №2

**«Создание приложения “Морской бой”»**

Работу выполнил  
студент 42 группы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Б.М.Ибрагимов

Направление подготовки 01.03.02 Прикладная математика и информатика

Проверил  
канд. техн. наук, доц.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А.Полупанов

Краснодар 2020

**Цель работы:** разработать игру с компьютером «Морской бой» на C#.

**Задача 1.** Разработать игру «Морской бой», с интерфейсом «человек-компьютер». Корабли расставляются по игровому полю с помощью рандома. До начала игры пользователь имеет возможность перестроить свое игровое поле с помощью соответствующей кнопки.

**ЛИСТИНГ ПРОГРАММЫ**

using NHibernate.Mapping;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BattleShip {

public partial class Form1 : Form {

public int sizePole = 10;

Button[,] YourBoard, OpponentBoard;

List<Point> AcceptPoints = new List<Point>();

int o = (new Random()).Next(0,10);

public Form1() {

InitializeComponent();

}

private void Button1\_Click(object sender, EventArgs e) {

OpponentBoard = ArrangeShips(OpponentBoard,false);

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

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

OpponentBoard[i, j].Visible = true;

}

}

this.button2.Enabled = false;

this.button1.Enabled = false;

}

private Button[,] ArrangeShips(Button[,] board, bool myBoard) {

Button[,] boardCopy = board;

List<int> ships = new List<int>() { 1, 2, 3, 4, 1, 2, 3, 1, 2, 1 };

List<Point> orientations = new List<Point> { new Point(0, -1), new Point(-1, 0), new Point(0, 1), new Point(1, 0) };

Random random = new Random();

HashSet<Point> forbiddenPoints = new HashSet<Point>();

while (ships.Count != 0) {

Point orientation = orientations[random.Next(orientations.Count - 1)];

int ship = ships[random.Next(ships.Count - 1)];

int indentA = (ship - 1) \* Math.Abs(orientation.X);

int indentB = (ship - 1) \* Math.Abs(orientation.Y);

int x = random.Next(indentA, sizePole - indentA);

int y = random.Next(indentB, sizePole - indentB);

Point positionStart = new Point(x, y);

bool permission = true;

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

if (forbiddenPoints.Contains(new Point(positionStart.X + i \* orientation.X, positionStart.Y + i \* orientation.Y))) {

permission = false;

break;

}

if (permission) {

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

Point p = new Point(positionStart.X + i \* orientation.X, positionStart.Y + i \* orientation.Y);

if (myBoard) boardCopy[p.X,p.Y].BackColor = Color.Yellow;

boardCopy[p.X, p.Y].Tag = 1;

forbiddenPoints.Add(p);

forbiddenPoints.Add(new Point(p.X, p.Y - 1));

forbiddenPoints.Add(new Point(p.X, p.Y + 1));

forbiddenPoints.Add(new Point(p.X - 1, p.Y));

forbiddenPoints.Add(new Point(p.X + 1, p.Y));

forbiddenPoints.Add(new Point(p.X - 1, p.Y - 1));

forbiddenPoints.Add(new Point(p.X + 1, p.Y + 1));

forbiddenPoints.Add(new Point(p.X + 1, p.Y - 1));

forbiddenPoints.Add(new Point(p.X - 1, p.Y + 1));

}

ships.Remove(ship);

}

}

return boardCopy;

}

private void Button2\_Click(object sender, EventArgs e) {

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

for (int j = 0; j < sizePole; j++)

YourBoard[i, j].BackColor = Color.Blue;

YourBoard = ArrangeShips(YourBoard,true);

}

private void Form1\_Load(object sender, EventArgs e) {

int heightButtons = this.groupBox2.Height / sizePole - 3;

Size buttonSize = new Size(this.groupBox2.Width / sizePole - 3, heightButtons);

Font buttonsFont = new Font("Arial", heightButtons / 2);

YourBoard = new Button[sizePole, sizePole];

OpponentBoard = new Button[sizePole, sizePole];

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

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

Button button = new Button() {

Name = i + " " + j,

Font = buttonsFont,

Size = buttonSize,

Location = new Point(10 + j \* buttonSize.Width, 20 + i \* buttonSize.Height),

Enabled = false,

BackColor = Color.Blue,

Tag = 0

};

YourBoard[i, j] = button;

this.groupBox2.Controls.Add(button);

AcceptPoints.Add(new Point(i, j));

Button button2 = new Button() {

Name = i + " " + j,

Font = buttonsFont,

Size = buttonSize,

Location = new Point(10 + j \* buttonSize.Width, 20 + i \* buttonSize.Height),

Visible = false,

BackColor = Color.Blue,

Tag = 0

};

button2.Click += ButtonPoleClick;

OpponentBoard[i, j] = button2;

this.groupBox1.Controls.Add(button2);

}

}

YourBoard = ArrangeShips(YourBoard,true);

}

private bool IsWiner2(Button[,] board) {

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

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

if (Convert.ToInt32(board[i, j].Tag) == 1) return false;

}

}

return true;

}

private void ButtonPoleClick(object sender, EventArgs e) {

string[] name = sender.GetType().GetProperty("Name").GetValue(sender).ToString().Split(' ');

Point pos = new Point(int.Parse(name[0]), int.Parse(name[1]));

OpponentBoard[pos.X, pos.Y].Enabled = false;

if (Convert.ToInt32(OpponentBoard[pos.X, pos.Y].Tag) == 1) {

OpponentBoard[pos.X, pos.Y].BackColor = Color.Red;

OpponentBoard[pos.X, pos.Y].Tag = 0;

}

else OpponentBoard[pos.X, pos.Y].BackColor = Color.White;

if (IsWiner2(OpponentBoard)) MessageBox.Show("Пользователь победил");

else {

Random random = new Random();

Point posOpp = AcceptPoints[random.Next(AcceptPoints.Count)];

if (Convert.ToInt32(YourBoard[posOpp.X, posOpp.Y].Tag) == 1) {

YourBoard[posOpp.X, posOpp.Y].BackColor = Color.Red;

YourBoard[posOpp.X, posOpp.Y].Tag = 0;

}

else YourBoard[posOpp.X, posOpp.Y].BackColor = Color.White;

AcceptPoints.Remove(posOpp);

if (IsWiner2(YourBoard)) MessageBox.Show("AI победил");

}

}

}

}

**РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

До начала игры пользователь имеет возможность рандомно переставлять свои корабли на своем игровом поле. Корабли на поле обозначены желтым цветом, а синим – море (то место, где кораблик не стоит). После нажатия кнопки «Начать», рядом появляется не обозначенное игровое поле противника (компьютера). Первым ход делает пользователь, нажимая на любую ячейку поля противника. Если человек попадает по кораблю, то эта ячейка окрашивается красным цветом и он (пользователь) делает ход еще раз. Иначе – ячейка окрашивается белым цветом и ход переходит противнику. Аналогично с компьютером, который совершает ход на рандомную ячейку поля противника. Если кто-то побеждает, то выходит соответствующее уведомление.

На рисунке 1 изображено игровое поле 10\*10 до начала игры:

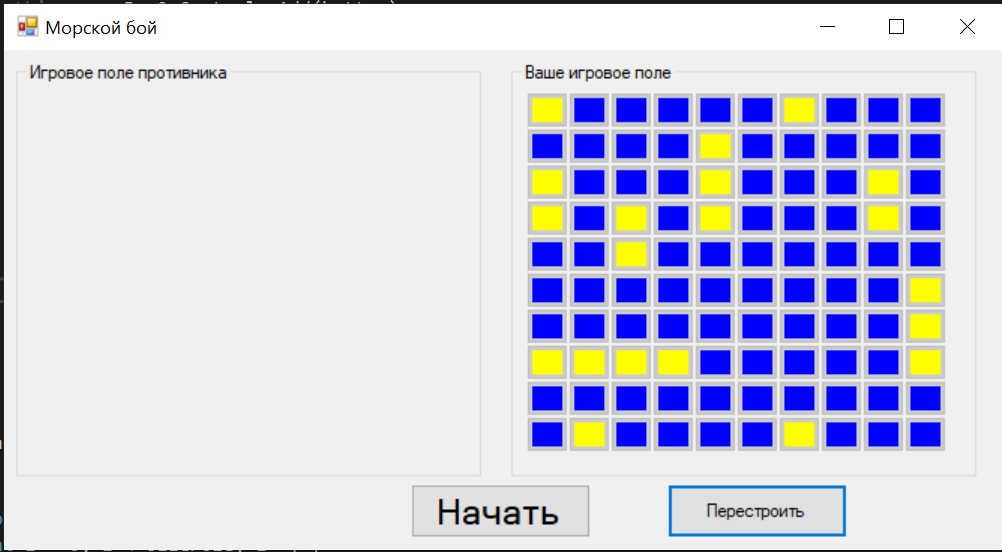


Рисунок 1 – игровое поле 10\*10 до начала игры

На рисунке 2 показано состояние игрового поля в начале игры:

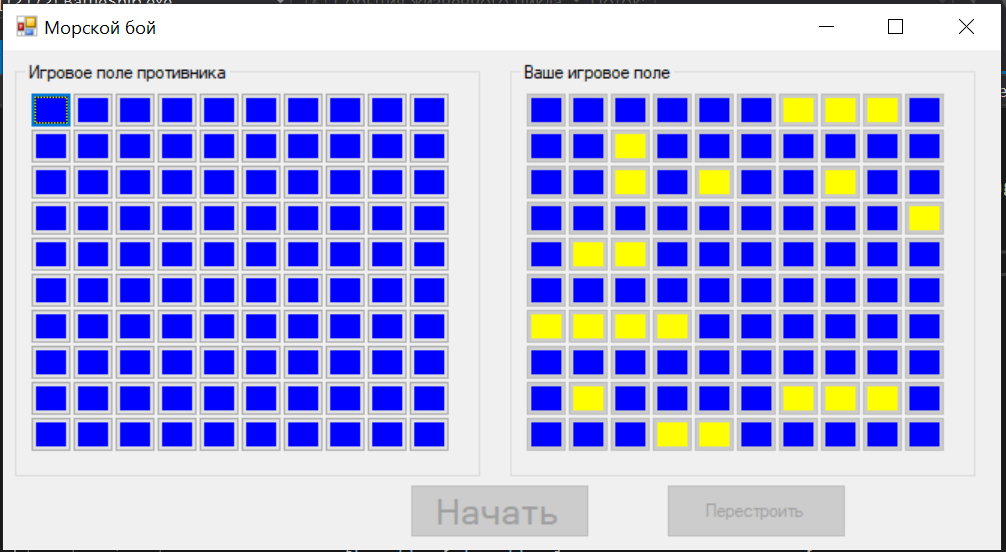


Рисунок 2 – игровое поле в начале игры

На рисунке 3 показано состояние игрового поля в процессе игры:



Рисунок 3 – игровое поле в процессе игры

**Вывод:** была написана игра «Морской бой» пользователя с компьютером, где корабли расставляются по игровому полю с помощью рандома. До начала игры пользователь имеет возможность перестроить свое игровое поле с помощью соответствующей кнопки. По окончанию игры выводится сообщение о победе одного из игроков.