Військовий інститут телекомунікацій та інформатизації

Кафедра

«Комп’ютерних інформаційних технологій»

**Курсова робота**

з дисципліни «Алгоритмізація та програмування»

на тему: «Використання мови С# для розробки програмних додатків»

Виконав: курсант 375 навчальної групи

солдат В.В. БІЛОЗОРЕЦЬ

Керівник: викладач кафедри № 21

капітан А.В.Микитюк

Київ 2018

Зміст

Анотація....................................................................................................3

Вступ.........................................................................................................5

Розділ 1.АНАЛІЗ ЗАВДАННЯ...............................................................6

1.1 Постановка задачі .........................................................................6

1.2 Вхідні дані......................................................................................7

1.3 Аналіз існуючих підходів до вирішення завдання.....................8

Розділ 2.ПРОЕКТУВАННЯ ТА ОСОБЛИВОСТІ РОЗРОБКИ….....9

2.1 Вимоги до програми………..……................................................9

2.2 Діаграми класів……………………….........................................10

2.3 Текст програмної реалізації........................................................11

2.4 Результати виконання програми…………………….................13

Розділ 3. ОСОБЛИВОСТІ ВИКОРИСТАННЯ ПРОГРАМНОГО

ДОДАТКУ………………......................................................................14

3.1 Порядок використання програмного додатку ………......…..14

ВИСНОВКИ...........................................................................................15

СПИСОК ВИКОРАСТАНИХ ДЖЕРЕЛ..............................................16

**АНОТАЦІЯ**

**курсової роботи на тему:**

**“Використання мови C# для розробки програмних додатків”**

**Курсова робота : містить 22 сторінок, 4 рисунки, 4 джерел.**

Курсова робота присвячена дослідженню баз данних, принципів ООП та роботі з WPF( *Windows Presentation Foundation*) для розробки *Desktop Applications*(настільних додатків). Результатом роботи є конструктор проїзду Х-розвілки автомобілями.

**ANNOTATION**

**Сourse work on the theme: "Using C # to develop software applications"**

**Coursework: contains 22 pages, 4 figures, 4 sources.**

Coursework is devoted to the study of databases, principles of OOP and work with WPF (*Windows Presentation Foundation*) for the development of Desktop Applications. The result of the work is the designer of the X-Intelligence car.

**ВСТУП**

**Актуальність теми**. Програмування є невід’ємним елементом для побудови системи управління. Під час участі в антитерористичній операції, боєздатність збройних сил, мобільність та ефективність на пряму залежать від чіткої роботи новітніх засобів розроблених завдяки програмуванню.

**Мета роботи:** розробка програмного модуля з GUI мовою програмування С#.

Виходячи з мети роботи, виникають наступні **завдання:**

**-** проаналізувати поставлене завдання;

- спроектувати та визначити особливості розробки;

- визначити особливості використання програмного додатку.

**Об’єкт досліджень:** побудова інтуітивно зрозумілого інтерфейсу та логіки.

**Предмет досліджень:** Морський бій.

**Розділ 1**

**АНАЛІЗ ЗАВДАННЯ**

* 1. Постановка задачі

Розробити мовою програмування C# морський бій з базою данних для статистики та можливістю зберігати та відновлювати ігри. В программі має бути ручна розстановка кораблів та автоматичка, у комп’ютер має два рівня важкості гри.

1.2 Вхідні дані

Вхідними даними для програми є початкова розтановка кораблів та вибір уровня гри комп’ютера.

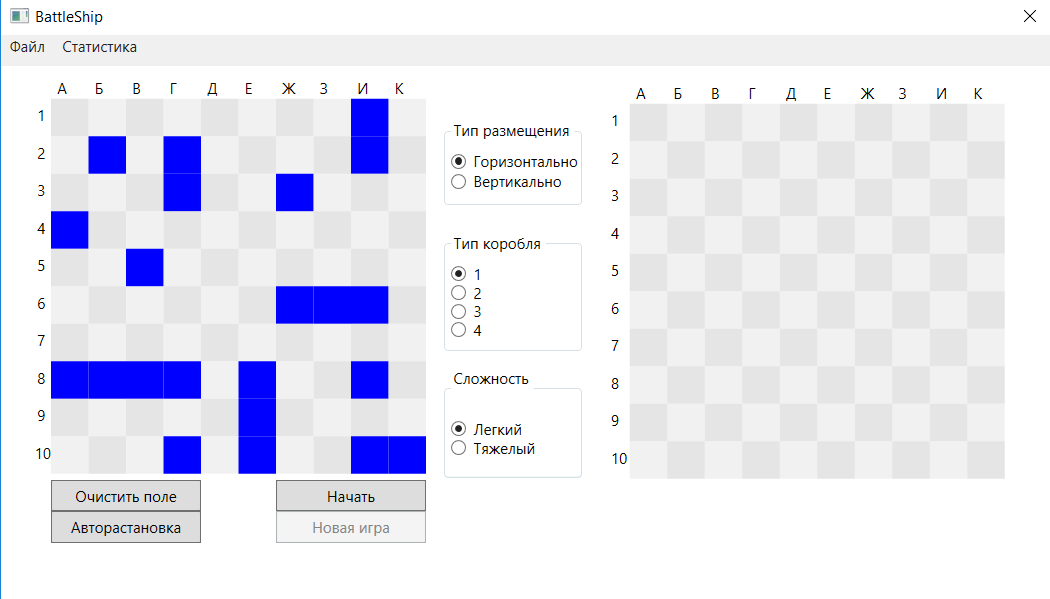


Рис 1.1 Початкова розстановка кораблів

1.3 Аналіз існуючих підходів до вирішення завдання

Основним підходом є алгоритм розстановки кораблів, так щоб кораблі не вийшли за поле або не на лізли на один одного.

**Висновок**

Отже, основними даними є кораблі та їх розташування.

**Розділ 2**

**ПРОЕКТУВАННЯ ТА ОСОБЛИВОСТІ РОЗРОБКИ**

2.1 Вимоги до програми

* конфіденційність, безпека і захист даних;
* відмовостійкість;
* час чекання відповіді при звертанні до системи (продуктивність);
* склад виконуваних функцій системи (запуск, швидкість реакції й ін.);
* положення стандартів з виконання сформульованих вимог.

Також програма має виконувати основні вимоги завдання згідно варіанту.

2.2 Діаграми класів

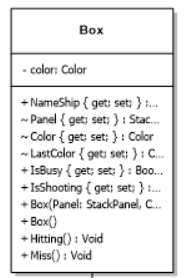


Рис 2.1 Діаграма класу Box

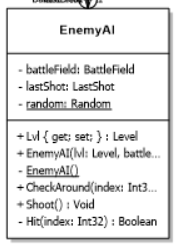


Рис 2.2 Діаграма класу *EnemyAI*

2.3 Текст програмної реалізації

**Клас EnemyAI**

*public class EnemyAI*

*{*

*class LastShot*

*{*

*public bool IsHit { get; set; } = false;*

*public int StartPositionShip { get; set; }*

*public bool LEFT { get; set; } = true;*

*public bool RIGHT { get; set; } = true;*

*public bool DOWN { get; set; } = true;*

*public bool TOP { get; set; } = true;*

*public int CountShot { get; set; }*

*public void Clear()*

*{*

*IsHit = false;*

*LEFT = true;*

*DOWN = true;*

*RIGHT = true;*

*TOP = true;*

*CountShot = 0;*

*}*

*}*

*public enum Side*

*{*

*LEFT,*

*RIGHT,*

*TOP,*

*DOWN,*

*NONE*

*}*

*public enum Level*

*{*

*Ez = 0,*

*Hard = 1*

*}*

*public Level Lvl { get; set; }*

*private BattleField battleField;*

*private LastShot lastShot = new LastShot();*

*private static Random random = new Random();*

*public EnemyAI(Level lvl, BattleField battleField)*

*{*

*this.Lvl = lvl;*

*this.battleField = battleField;*

*}*

*public bool CheckAround(int index, Side side)*

*{*

*if (index < 0)*

*return false;*

*if (battleField.boxes[index].IsShooting)*

*return false;*

*if (index - 10 >= 0)*

*{*

*if (index % 10 != 0 && index != 0)//верхний левый угол*

*{*

*if (battleField.boxes[index - 10 - 1].IsBusy && battleField.boxes[index - 10 - 1].IsShooting)*

*return false;*

*}*

*if (battleField.boxes[index - 10].IsBusy && battleField.boxes[index - 10].IsShooting && side != Side.DOWN)//верхний*

*return false;*

*if (index % 10 != 9 && index != 99)//верхний правый*

*{*

*if (battleField.boxes[index - 10 + 1].IsBusy && battleField.boxes[index - 10 + 1].IsShooting)*

*return false;*

*}*

*}//проверяем верх*

*if (index + 10 <= 99)*

*{*

*if (index % 10 != 0 && index != 0)*

*{*

*if (battleField.boxes[index + 10 - 1].IsBusy && battleField.boxes[index + 10 - 1].IsShooting)*

*return false;*

*}*

*if (battleField.boxes[index + 10].IsBusy && battleField.boxes[index + 10].IsShooting && side != Side.TOP)//нижний*

*return false;*

*if (index % 10 != 9 && index != 99)*

*{*

*if (battleField.boxes[index + 10 + 1].IsBusy && battleField.boxes[index + 10 + 1].IsShooting)*

*return false;*

*}*

*}//проверяем низ*

*if (index % 10 != 0 && side != Side.RIGHT)*

*{*

*if (battleField.boxes[index - 1].IsBusy && battleField.boxes[index - 1].IsShooting)*

*return false;*

*}*

*if (index % 10 != 9 && side != Side.LEFT)*

*{*

*if (battleField.boxes[index + 1].IsBusy && battleField.boxes[index + 1].IsShooting)*

*return false;*

*}*

*return true;*

*}*

*public void Shoot()*

*{*

*if (Lvl == Level.Ez)*

*{*

*while (true)*

*{*

*int index = random.Next(0, 100);*

*if (battleField.boxes[index].IsShooting)*

*{*

*continue;*

*}*

*else*

*{*

*if (battleField.boxes[index].IsBusy)*

*{*

*battleField.boxes[index].Hitting();*

*}*

*else*

*{*

*battleField.boxes[index].Miss();*

*}*

*break;*

*}*

*}*

*}*

*else*

*{*

*Restart:*

*if (!lastShot.IsHit)*

*{*

*while (true)*

*{*

*int index = random.Next(0, 100);*

*if (battleField.boxes[index].IsShooting)*

*{*

*continue;*

*}*

*else*

*{*

*if (battleField.boxes[index].IsBusy)*

*{*

*battleField.boxes[index].Hitting();*

*if (battleField.boxes.GetCountLivingShipsByName(battleField.boxes[index].NameShip) == 0)*

*{*

*lastShot.Clear();*

*}*

*else*

*{*

*lastShot.IsHit = true;*

*lastShot.CountShot++;*

*lastShot.StartPositionShip = index;*

*}*

*}*

*else*

*{*

*battleField.boxes[index].Miss();*

*}*

*break;*

*}*

*}*

*}*

*else*

*{*

*if (lastShot.LEFT)*

*{*

*if (CheckAround(lastShot.StartPositionShip - lastShot.CountShot, Side.LEFT))*

*{*

*if ((lastShot.StartPositionShip - lastShot.CountShot) % 10 == 9 || lastShot.StartPositionShip - lastShot.CountShot < 0)*

*{*

*lastShot.LEFT = false;*

*}*

*else*

*{*

*if (Hit(lastShot.StartPositionShip - lastShot.CountShot))*

*{*

*if (battleField.boxes.GetCountLivingShipsByName(battleField.boxes[lastShot.StartPositionShip].NameShip) != 0)*

*{*

*lastShot.TOP = false;*

*lastShot.DOWN = false;*

*}*

*return;*

*}*

*else*

*{*

*lastShot.LEFT = false;*

*return;*

*}*

*}*

*}*

*lastShot.LEFT = false;*

*}*

*if (lastShot.TOP)*

*{*

*if (CheckAround(lastShot.StartPositionShip - lastShot.CountShot \* 10, Side.TOP))*

*{*

*if (lastShot.StartPositionShip - lastShot.CountShot \* 10 % 10 == 0 || lastShot.StartPositionShip - lastShot.CountShot \* 10 < 0)*

*{*

*lastShot.TOP = false;*

*}*

*else*

*{*

*if (Hit(lastShot.StartPositionShip - lastShot.CountShot \* 10))*

*{*

*if (battleField.boxes.GetCountLivingShipsByName(battleField.boxes[lastShot.StartPositionShip].NameShip) != 0)*

*{*

*lastShot.LEFT = false;*

*lastShot.RIGHT = false;*

*}*

*return;*

*}*

*else*

*{*

*lastShot.TOP = false;*

*return;*

*}*

*}*

*}*

*lastShot.TOP = false;*

*}*

*if (lastShot.RIGHT)*

*{*

*if (CheckAround(lastShot.StartPositionShip + lastShot.CountShot, Side.RIGHT))*

*{*

*if ((lastShot.StartPositionShip + lastShot.CountShot) % 10 == 9 || lastShot.StartPositionShip + lastShot.CountShot > 99)*

*{*

*lastShot.RIGHT = false;*

*}*

*else*

*{*

*if (Hit(lastShot.StartPositionShip + lastShot.CountShot))*

*{*

*if (battleField.boxes.GetCountLivingShipsByName(battleField.boxes[lastShot.StartPositionShip].NameShip) != 0)*

*{*

*lastShot.TOP = false;*

*lastShot.DOWN = false;*

*}*

*return;*

*}*

*else*

*{*

*lastShot.RIGHT = false;*

*return;*

*}*

*}*

*}*

*lastShot.RIGHT = false;*

*}*

*if (lastShot.DOWN)*

*{*

*if (CheckAround(lastShot.StartPositionShip + lastShot.CountShot \* 10, Side.DOWN))*

*{*

*if (lastShot.StartPositionShip + lastShot.CountShot \* 10 % 10 == 9 || lastShot.StartPositionShip + lastShot.CountShot \* 10 > 99)*

*{*

*lastShot.DOWN = false;*

*}*

*else*

*{*

*if (Hit(lastShot.StartPositionShip + lastShot.CountShot \* 10))*

*{*

*if (battleField.boxes.GetCountLivingShipsByName(battleField.boxes[lastShot.StartPositionShip].NameShip) != 0)*

*{*

*lastShot.LEFT = false;*

*lastShot.RIGHT = false;*

*}*

*return;*

*}*

*else*

*{*

*lastShot.DOWN = false;*

*return;*

*}*

*}*

*}*

*lastShot.DOWN = false;*

*}*

*lastShot.Clear();*

*goto Restart;*

*}*

*}*

*}*

*private bool Hit(int index)*

*{*

*if (battleField.boxes[index].IsBusy)*

*{*

*battleField.boxes[index].Hitting();*

*if (battleField.boxes.GetCountLivingShipsByName(battleField.boxes[index].NameShip) == 0)*

*{*

*lastShot.Clear();*

*}*

*else*

*{*

*lastShot.IsHit = true;*

*lastShot.CountShot++;*

*}*

*return true;*

*}*

*else*

*{*

*battleField.boxes[index].Miss();*

*lastShot.CountShot = 1;*

*return false;*

*}*

*}*

*}* **Інтерфейс ICar**

public class Box

{

public string NameShip { get; set; }

internal StackPanel Panel { get; set; }

private Color color;

internal Color Color

{

get { return color; }

set

{

color = value;

Panel.Background = new SolidColorBrush(color);

}

}

internal Color LastColor { get; set; }

public bool IsBusy { get; set; } = false;

public bool IsShooting { get; set; } = false;

public Box(StackPanel Panel, Color Color)

{

this.Panel = Panel;

this.Color = Color;

this.LastColor = Color;

}

public Box(){}

public void Hitting()

{

StackPanel stackPanel = new StackPanel();

stackPanel.Margin = new Thickness(5, 5, 5, 5);

stackPanel.Width = 20;

stackPanel.Height = 20;

stackPanel.Background = new SolidColorBrush(Colors.Red);

Panel.Children.Add(stackPanel);

IsShooting = true;

}

public void Miss()

{

StackPanel stackPanel = new StackPanel();

stackPanel.Margin = new Thickness(10, 10, 10, 10);

stackPanel.Width = 10;

stackPanel.Height = 10;

stackPanel.Background = new SolidColorBrush(Colors.Black);

Panel.Children.Add(stackPanel);

IsShooting = true;

}

}

2.4 Результати виконання програми

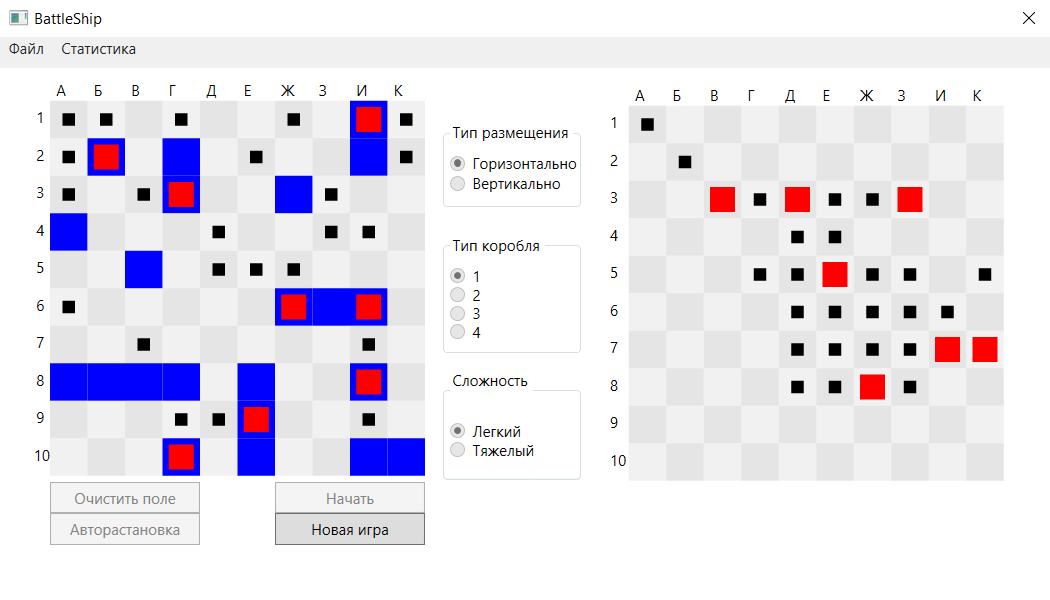


Рис 2.3 Граємо

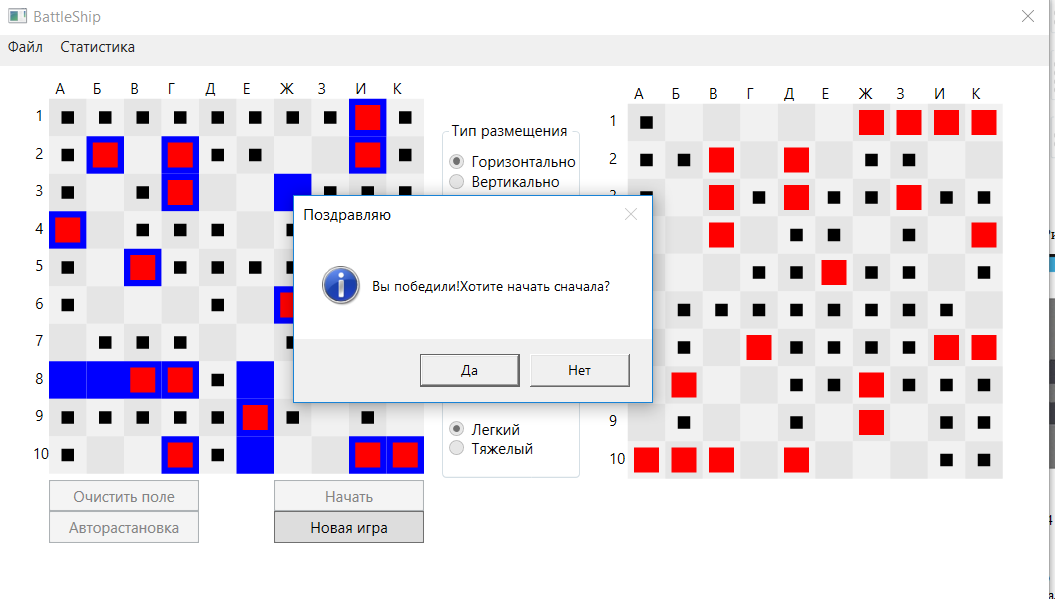


Рис 2.4 Победа

**Висновок**

Отже, після того як було побудовано діаграми класів, перейшли до розробки програмного коду. Загалом ми маємо 9 класів (разом з класами вікон). На рисунках можна побачити результати успішної роботи над програмою.

**Розділ 3**

**ОСОБЛИВОСТІ ВИКОРИСТАННЯ ПРОГРАМНОГО**

**ДОДАТКУ**

3.1 Порядок використання програмного додатку

Програмний додаток розробленно для гри морський бій, кораблі виставляются автоматично або вручну, комп’ютер має два рівня важкості гри.

**ВИСНОВКИ**

Програмування є невід’ємним елементом для побудови системи управління. Під час участі в антитерористичній операції, боєздатність збройних сил, мобільність та ефективність на пряму залежать від чіткої роботи новітніх засобів розроблених завдяки програмуванню.

Отже, основними даними є кораблі та їх розташування.

Отже, після того як було побудовано діаграми класів, перейшли до розробки програмного коду. Загалом ми маємо 9 класів (разом з класами вікон). На рисунках можна побачити результати успішної роботи над програмою.

Під час курсової роботи був розроблений програмний модуль мовою програмування С#.

Під час аналізу поставленого завдання ми визначили вхідні дані та їх типи, визначили підхід розробки програмного модуля, а також визначились с основними функціями програми.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Ендрю Троелсен «C# 5.0 and .NET4.0»
2. Стефан Ренді Девіс, Чак Сфер «C# 2005 для чайников»

**Віддалений доступ**

*Pесурсу віддаленого доступу (****Web-сайт****)*1. Курс навчання C#./NET

Режим доступу: [www.metanit.com](http://www.metanit.com) (дата звернення 08.04.2018)

*Pесурсу віддаленого доступу (****Web-сайт****)*2. C#./NET

Режим доступу: [www.profesorweb.com](http://www.profesorweb.com) (дата звернення 01.04.2018)