Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М. Машерова

Отчет

по учебной практике по программированию

«Модуль 7 Проектная работа»

Выполнил учащийся Близнецов А.А.

группы 3ПОИТ22 19.10.2024 г.

Проверил Алейников М.А.

2024 г.

Орша, 2024

Модуль 7 Проектная работа:

1. Выберите проект, который вы разработали в рамках предмета Основы алгоритмизации и программирования
2. Сформулируйте требования к проекту и создайте план разработки.
3. Разработайте проект, следуя плану и используя полученные знания и навыки.
4. Проведите тестирование и отладку проекта.
5. Подготовьте документацию и презентацию для защиты проекта перед преподавателями.

Выполнение:

Функциональные требования:

1. Начало игры:
   1. Система должна позволять игрокам начать новую игру.
   2. Игра должна автоматически сбрасывать состояние доски.
2. Игроки:
   1. Система должна поддерживать двух игроков (Игрок 1 и Игрок 2).
   2. Игроки должны иметь возможность выбрать символ (крестик или нолик).
3. Ход игры:
   1. Игроки должны поочередно делать ходы.
   2. Система должна проверять корректность хода (проверка, занята ли клетка).
4. Проверка победителя:
   1. Система должна определять победителя после каждого хода.
   2. Если один из игроков выигрывает, система должна выводить сообщение о победе.
   3. Если все клетки заполнены и нет победителя, система должна объявить ничью.
5. Сброс игры:
   1. Игроки должны иметь возможность сбросить игру в любое время.
   2. При сбросе все данные о текущей игре должны быть очищены.

Нефункциональные требования:

1. Удобство использования:
   1. Интерфейс должен быть интуитивно понятным и доступным для пользователей всех возрастов.
2. Производительность:
   1. Игра должна быстро реагировать на действия игроков (не более 1 секунды на выполнение любого действия).

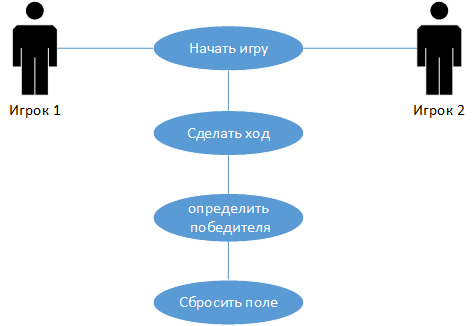


Рисунок 1 – Диаграмма UML Use-case

Листинг 1. Логика программы

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private char currentPlayer = 'X';

private char[,] board = new char[3, 3];

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

Button button = sender as Button;

if (button.Content == null)

{

button.Content = currentPlayer;

int row = Grid.GetRow(button);

int col = Grid.GetColumn(button);

board[row, col] = currentPlayer;

if (CheckWin())

{

MessageBox.Show($"{currentPlayer} победили!");

ResetGame();

}

else if (IsBoardFull())

{

MessageBox.Show("Ничья!");

ResetGame();

}

currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';

}

}

private bool CheckWin()

{

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

{

if (board[i, 0] != '\0' && board[i, 0] == board[i, 1] && board[i, 1] == board[i, 2])

return true;

}

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

{

if (board[0, i] != '\0' && board[0, i] == board[1, i] && board[1, i] == board[2, i])

return true;

}

if (board[0, 0] != '\0' && board[0, 0] == board[1, 1] && board[1, 1] == board[2, 2])

return true;

if (board[0, 2] != '\0' && board[0, 2] == board[1, 1] && board[1, 1] == board[2, 0])

return true;

return false;

}

private bool IsBoardFull()

{

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

{

for (int j = 0; j < 3; j++)

{

if (board[i, j] == '\0')

return false;

}

}

return true;

}

private void ResetGame()

{

board = new char[3, 3];

foreach (var control in MainGrid.Children)

{

if (control is Button button)

{

button.Content = null; NewGameButton.Content = "Очистить поле";

}

}

currentPlayer = 'О';

MessageBox.Show("Крестики начинают");

}

private void NewGameButton\_Click(object sender, RoutedEventArgs e)

{

ResetGame();

currentPlayer = 'X';

}

}

}

Листинг 2. Код XAML приложения

<Window x:Class="WpfApp1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid x:Name="MainGrid">

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="Auto"/>

<!-- Строка для кнопки новой игры -->

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Button Name="Button00" Click="Button\_Click" Grid.Row="0" Grid.Column="0" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button01" Click="Button\_Click" Grid.Row="0" Grid.Column="1" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button02" Click="Button\_Click" Grid.Row="0" Grid.Column="2" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button10" Click="Button\_Click" Grid.Row="1" Grid.Column="0" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button11" Click="Button\_Click" Grid.Row="1" Grid.Column="1" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button12" Click="Button\_Click" Grid.Row="1" Grid.Column="2" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button20" Click="Button\_Click" Grid.Row="2" Grid.Column="0" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button21" Click="Button\_Click" Grid.Row="2" Grid.Column="1" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="Button22" Click="Button\_Click" Grid.Row="2" Grid.Column="2" FontSize="100" FontFamily="Microsoft YaHei UI"/>

<Button Name="NewGameButton" Content="Очистить поле" Click="NewGameButton\_Click" Grid.Row="3" Grid.ColumnSpan="3" FontSize="30" FontFamily="Microsoft YaHei UI" Margin="10"/>

</Grid>

</Window>

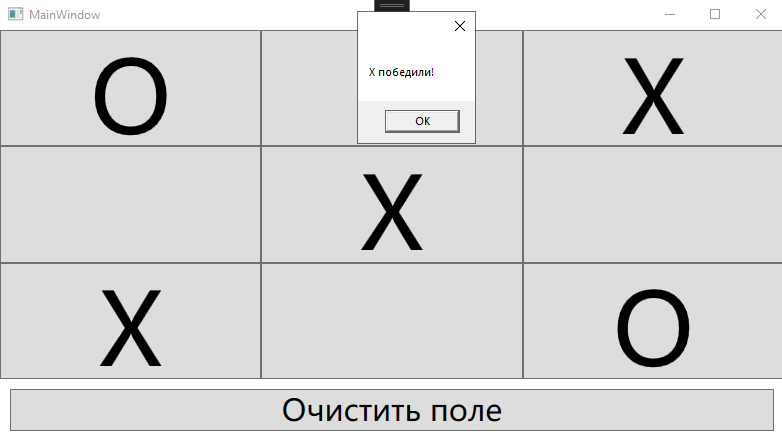


Рисунок 2 – Приложение “Крестики нолики”

### Тестовые случаи для выбора клетки

1. **Выбор пустой клетки**
   1. ****Действие****: Игрок выбирает пустую клетку.
   2. ****Ожидаемый результат****: Клетка заполнена знаком игрока (X или O), и ход передается другому игроку.
   3. ****Результат****: Клетка заполнена знаком игрока (X или O), ход передается другому игроку.
2. **Выбор занятой клетки**
   1. ****Действие****: Игрок пытается выбрать уже занятую клетку.
   2. ****Ожидаемый результат****: Знак в клетке не меняется и ход игрока продолжается
   3. ****Результат****: Знак в клетке не меняется и ход игрока продолжается

### Тестовые случаи для сброса поля

1. **Сброс пустого поля**
   1. ****Действие****: Игрок вызывает функцию сброса поля.
   2. ****Ожидаемый результат****: Поле остается пустым, игра начинается заново.
   3. ****Результат****: Поле остается пустым, игра начинается заново. Все клетки очищены.
2. **Сброс поля с заполненными клетками**
   1. ****Действие****: Игрок вызывает функцию сброса поля после нескольких ходов.
   2. ****Ожидаемый результат****: Поле очищается, все клетки становятся пустыми, игра начинается заново.
   3. ****Результат****: Поле очищается, все клетки становятся пустыми, игра начинается заново.
3. **Сброс поля после завершенной игры**
   1. ****Действие****: Игрок вызывает функцию сброса после завершенной игры.
   2. ****Ожидаемый результат****: Поле очищается, игра начинается заново.
   3. ****Результат****: Поле очищается, игра начинается заново.

### Тестовые случаи для состояния игры

1. **Выигрыш игрока X**
   1. ****Действие****: Игрок X заполняет три клетки в ряд.
   2. ****Ожидаемый результат****: Появляется сообщение о выигрыше игрока X.
   3. ****Результат****: Появляется сообщение: "Игрок X выиграл!".
2. **Выигрыш игрока O**
   1. ****Действие****: Игрок O заполняет три клетки в колонне.
   2. ****Ожидаемый результат****: Появляется сообщение о выигрыше игрока O.
   3. ****Результат****: Появляется сообщение: "Игрок O выиграл!".
3. **Ничья**
   1. ****Действие****: Все клетки заполнены, но ни один игрок не выиграл.
   2. ****Ожидаемый результат****: Появляется сообщение о ничьей.
   3. ****Результат****: Появляется сообщение: "Ничья!".

### Тестовые случаи для интерфейса

1. **Доступность кнопки сброса**
   1. ****Действие****: Игрок пытается нажать кнопку сброса.
   2. ****Ожидаемый результат****: Кнопка доступна в любой момент игры, если игра не завершена.
   3. ****Результат****: Кнопка доступна в любой момент игры, если игра не завершена. После нажатия поле сбрасывается.