Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Магнитогорский государственный технический университет им. Г.И. Носова»

Многопрофильный колледж

Отделение №2 Информационные технологии и транспорт

**Отчёт**

по МДК.02.01. Технология разработки программного обеспечения

на тему: создание игры «Крестики-нолики»

Исполнители:Варфоломеев М.В, Дымов Т.А, студенты группы ИСпП-21-2

Преподаватель: Коржина В. С.

**Техническое задание**

Создать игру “Морской бой” на языке C# WPF. С возможностью играть против игрока.

**Описание работы программы**

1. **Инициализация игры**
   * При запуске приложения создаются две сетки (по 10x10 ячеек) для каждого игрока. Также инициализируются переменные, необходимые для работы игры.
2. **Начало расстановки**
   * Игроки нажимают кнопку "Начать расстановку", чтобы перейти в режим расстановки кораблей.
   * Каждый игрок по очереди расставляет свои 5 кораблей на своей сетке, нажимая на ячейки. Ячейки с кораблями отмечаются серым цветом.
   * После расстановки всех кораблей игроком 1, начинается расстановка кораблей игроком 2.
3. **Начало игры**
   * После расстановки всех кораблей игроком 2, игроки нажимают кнопку "Начать игру". Это отключает режим расстановки и включает игровой режим.
4. **Игровой процесс**
   * Игроки по очереди выбирают ячейки на сетке противника для стрельбы.
   * Если игрок попадает в корабль противника, ячейка окрашивается в красный цвет и уменьшается счетчик оставшихся кораблей противника. Если выстрел мимо, ячейка окрашивается в синий цвет.
5. **Проверка победы**
   * После каждого удачного выстрела проверяется, остались ли у противника корабли. Если все корабли потоплены, выводится сообщение о победе и игра заканчивается.
6. **Отображение правил**
   * Игроки могут нажать кнопку "Правила игры", чтобы посмотреть основные правила игры. Окно с правилами отображает информацию о том, как расставлять корабли и как проходит игровой процесс.

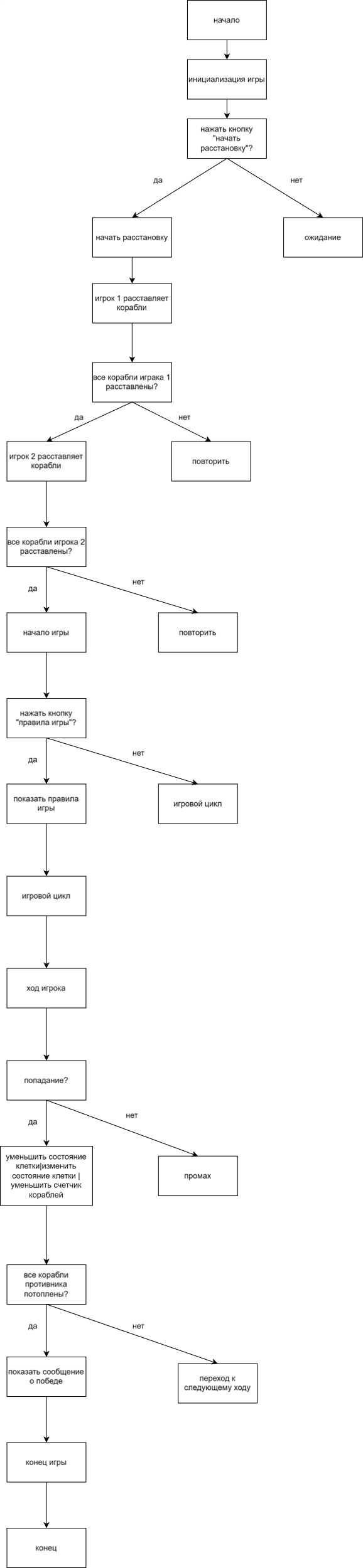
**Логика игры:**

1. **Инициализация и расстановка кораблей**
   * При нажатии кнопки "Начать расстановку" запускается процесс расстановки кораблей. Игроки по очереди размещают свои корабли на сетке. Когда один игрок заканчивает расстановку, управление передается следующему игроку.
2. **Начало игры**
   * После расстановки кораблей нажимается кнопка "Начать игру", и начинается игровой процесс. Оба игрока по очереди стреляют по сетке противника.
3. **Обработка выстрелов**
   * При нажатии на ячейку сетки противника проверяется, попал ли выстрел в корабль. В случае попадания уменьшается количество оставшихся кораблей у противника. Если все корабли противника потоплены, игра заканчивается с сообщением о победе.
4. **Победа**
   * Как только один из игроков потопит все корабли противника, выводится сообщение о победе и игра завершается.

Таким образом, игра "Морской бой" реализована с использованием WPF на C#, предоставляя пользователям возможность играть друг против друга, расставлять корабли, делать выстрелы по сетке противника и отслеживать победу на основе состояния сетки.

Ссылка на репозиторий игры: https://github.com/Balenciaga-lab/SeaBattle

**Блок-схема**



**Приложение А**

namespace SeaBattle

{

public partial class MainWindow : Window

{

private const int GridSize = 10;

private Button[,] player1Buttons = new Button[GridSize, GridSize];

private Button[,] player2Buttons = new Button[GridSize, GridSize];

private bool[,] player1Ships = new bool[GridSize, GridSize];

private bool[,] player2Ships = new bool[GridSize, GridSize];

private int player1ShipsLeft = 5;

private int player2ShipsLeft = 5;

private bool isPlayer1Turn = true;

private bool isPlacingShips = false;

private bool isGameStarted = false;

private int shipsToPlace = 5;

public MainWindow()

{

InitializeComponent();

InitializeGrid(Player1Grid, player1Buttons, true);

InitializeGrid(Player2Grid, player2Buttons, false);

}

private void InitializeGrid(Grid grid, Button[,] buttons, bool isPlayer1)

{

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

{

grid.RowDefinitions.Add(new RowDefinition());

grid.ColumnDefinitions.Add(new ColumnDefinition());

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

{

Button button = new Button();

button.Click += (sender, e) => GridButton\_Click(sender, e, isPlayer1);

grid.Children.Add(button);

Grid.SetRow(button, i);

Grid.SetColumn(button, j);

buttons[i, j] = button;

}

}

}

private void GridButton\_Click(object sender, RoutedEventArgs e, bool isPlayer1)

{

if (!isPlacingShips && !isGameStarted)

return;

Button button = sender as Button;

int row = Grid.GetRow(button);

int column = Grid.GetColumn(button);

if (isPlacingShips)

{

if (isPlayer1Turn == isPlayer1)

{

if (button.Background != System.Windows.Media.Brushes.Gray)

{

button.Background = System.Windows.Media.Brushes.Gray;

if (isPlayer1)

player1Ships[row, column] = true;

else

player2Ships[row, column] = true;

shipsToPlace--;

if (shipsToPlace == 0)

{

isPlayer1Turn = !isPlayer1Turn;

shipsToPlace = 5;

if (!isPlayer1Turn)

{

MessageBox.Show("Игрок 2, расставьте свои корабли.");

}

else

{

isPlacingShips = false;

StartGameButton.IsEnabled = true;

HideShips(player1Buttons);

HideShips(player2Buttons);

}

}

}

}

}

else if (isGameStarted)

{

if (isPlayer1Turn == isPlayer1)

return;

if (button.Background == System.Windows.Media.Brushes.LightGray)

{

bool hit = isPlayer1 ? player2Ships[row, column] : player1Ships[row, column];

button.Content = hit ? "X" : "O";

if (hit)

{

button.Background = System.Windows.Media.Brushes.Red;

if (isPlayer1)

player2ShipsLeft--;

else

player1ShipsLeft--;

if (CheckWin(isPlayer1))

{

MessageBox.Show(isPlayer1 ? "Игрок 2 победил!" : "Игрок 1 победил!");

ResetGame();

}

}

else

{

button.Background = System.Windows.Media.Brushes.Blue;

}

isPlayer1Turn = !isPlayer1Turn;

}

}

}

private bool CheckWin(bool isPlayer1)

{

return isPlayer1 ? player2ShipsLeft == 0 : player1ShipsLeft == 0;

}

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

{

ResetGame();

isPlacingShips = true;

}

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

{

isGameStarted = true;

StartGameButton.IsEnabled = false;

}

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

{

string rules = "Правила игры Морской бой:\n\n" +

"1. Каждый игрок расставляет на своем поле 5 кораблей.\n" +

"2. Корабли не могут касаться друг друга.\n" +

"3. После расстановки кораблей начинается игра.\n" +

"4. Игроки по очереди стреляют по клеткам на поле противника.\n" +

"5. Цель игры — потопить все корабли противника.";

MessageBox.Show(rules, "Правила игры");

}

private void ResetGame()

{

ResetGrid(player1Buttons, player1Ships);

ResetGrid(player2Buttons, player2Ships);

isPlayer1Turn = true;

isPlacingShips = false;

isGameStarted = false;

shipsToPlace = 5;

player1ShipsLeft = 5;

player2ShipsLeft = 5;

StartGameButton.IsEnabled = false;

}

private void ResetGrid(Button[,] buttons, bool[,] ships)

{

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

{

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

{

buttons[i, j].Content = "";

buttons[i, j].Background = System.Windows.Media.Brushes.LightGray;

ships[i, j] = false;

}

}

}

private void HideShips(Button[,] buttons)

{

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

{

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

{

if (buttons[i, j].Background == System.Windows.Media.Brushes.Gray)

{

buttons[i, j].Background = System.Windows.Media.Brushes.LightGray;

}

}

}

}

}

}