Kółko i krzyżyk – Projekt semestralny

Autorzy:

Damian Herbut

Michał Holly

David Nastevski

Celem Projektu było stworzenie aplikacji wzorowanej na popularnej grze w kółko i krzyżyk. Jest to dwuosobowa gra planszowa o wymiarach 5x5. Celem każdego z graczy jest utworzenie poziomej, pionowej lub ukośnej lini o długości 5 pól. Całość odbywa się w systemie turowym, wygrywa osoba która zrealizuje cel jako pierwsza.

Po uruchomieniu aplikacji gra się od razu zaczyna, w dolnej częsci okna wyświetlając turę danego gracza. Po zakończeniu gry jest wyświetlana informacja o tym który gracz zwyciężył. Jest także możliwość zresetowania gry do stanu startowego w dowolnym momencie poprzez naciśnięcie przycisku „Resetuj gre”.

Specyfikacja części logicznej aplikacji:

Część odpowiadajaca za liczenie ruchów graczy:

```
public Logika() { }
 private int amountOfMoves = 0;
 //liczenie ruchów
 public int AmountOfMoves
 {
     get { return amountOfMoves; }
     set { amountOfMoves = value; }
 }

 private int currentPlayer = 1;
 public int CurrentPlayer
 {
     get { return currentPlayer; }
     set { currentPlayer = value; }
 }
```

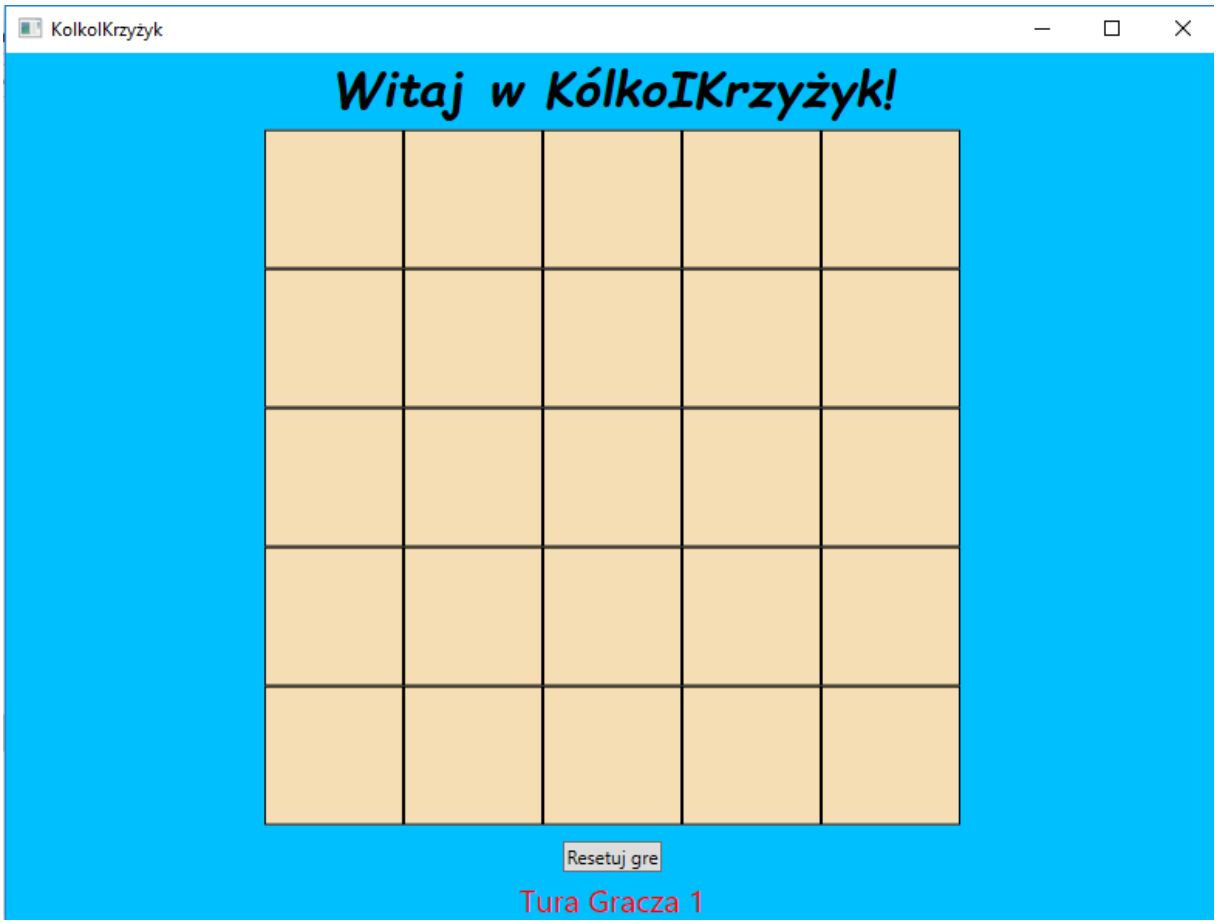
Określenie zwycięzcy w rozgrywce:

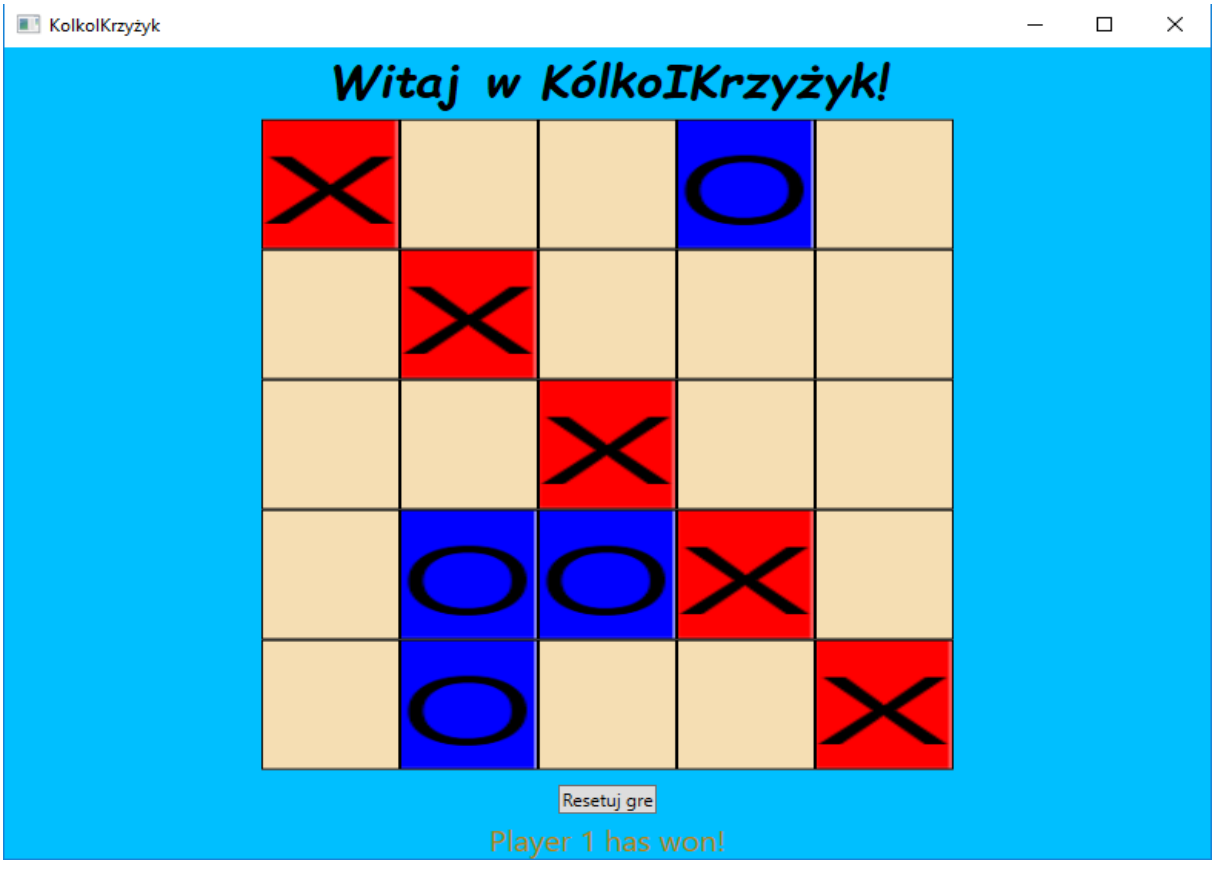
```
private bool gameOver;

 public bool GameOver
 {
     get { return gameOver; }
     set { gameOver = value; }
 }
 public bool IsPlayerOneTurn
 {
```

```
        get { return AmountOfMoves % 2 == 0; }
    }
    private int[] pressedButtons = new int[25];
    public int[] PressedButtons
    {
        get { return pressedButtons; }
        set { pressedButtons = value; }
    }
                private bool WinningCombinationFound(int left1, int left2, int
         middle, int right2, int right1)
    {
        return (PressedButtons[left1] == CurrentPlayer &&
            PressedButtons[left2] == CurrentPlayer &&
            PressedButtons[middle] == CurrentPlayer &&
            PressedButtons[right2] == CurrentPlayer &&
            PressedButtons[right1] == CurrentPlayer);
    }
    private bool GameWinnerFound()

{
        //poziomo
        if (WinningCombinationFound(0, 1, 2, 3, 4)) return true;
        if (WinningCombinationFound(5, 6, 7, 8, 9)) return true;
        if (WinningCombinationFound(10, 11, 12, 13, 14)) return true;
        if (WinningCombinationFound(15, 16, 17, 18, 19)) return true;
        if (WinningCombinationFound(20, 21, 22, 23, 24)) return true;
        //pionowo
        if (WinningCombinationFound(0, 5, 10, 15, 20)) return true;
        if (WinningCombinationFound(1, 6, 11, 16, 21)) return true;
        if (WinningCombinationFound(2, 7, 12, 17, 22)) return true;
        if (WinningCombinationFound(3, 8, 13, 18, 23)) return true;
        if (WinningCombinationFound(4, 9, 22, 23, 24)) return true;
        //skos
        if (WinningCombinationFound(0, 6, 12, 18, 24)) return true;
        if (WinningCombinationFound(4, 8, 12, 16, 20)) return true;
        return false;
    }

}
    public void CheckForGameWinner()
    {
        GameOver = GameWinnerFound();
    }
```

Funkcja odpowiadająca za resetowanie gry:

```
public void ResetGame()
    {
        AmountOfMoves = 0;
        GameOver = false;
        CurrentPlayer = 1;
        PressedButtons = new int[25];
    }
```


KolkolKrzyżyk
Witaj w KólkoIKrzyżyk!
Resetuj gre
Tura Gracza 1


KolkolKrzyżyk
Witaj w KólkoIKrzyżyk!
Resetuj gre
Player 1 has won!

Testy jednostkowe:

```
[TestMethod]
        public void CheckForGameWinner_NoWinner_ReturnsFalse()
        {

            Logika gameLogic = new Logika();
            gameLogic.CurrentPlayer = 1;
            gameLogic.PressedButtons = new int[]
{1,2,1,2,1,2,1,2,2,1,1,1,2,2,1,2,2,1,1,1,1,2,1,2,2};
            bool gameWon = false;


            gameLogic.CheckForGameWinner();


            Assert.AreEqual(gameWon, gameLogic.GameOver);
        }

        [TestMethod]
        public void CheckForGameWinner_ValidWinner_ReturnsTrue()
        {

            Logika gameLogic = new Logika();
            gameLogic.CurrentPlayer = 1;
            gameLogic.PressedButtons = new int[]
{1,1,2,0,0,0,1,2,1,0,2,1,2,2,1,2,1,2,1,2,2,1,1,2,1};
            bool gameWon = true;


            gameLogic.CheckForGameWinner();


            Assert.AreEqual(gameWon, gameLogic.GameOver);
        }
```

Aplikacja została przetestowana i wszystkie funkcjonalności działają prawdiłowo.