

Dokumentacja

1. Język programowania : C#
2. Wykorzystany framework: NETframework 4.6.1
3. Wykorzystane biblioteki:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows;  
using System.Windows.Controls;  
using System.Windows.Data;  
using System.Windows.Documents;  
using System.Windows.Input;  
using System.Windows.Media;  
using System.Windows.Media.Imaging;  
using System.Windows.Navigation;  
using System.Windows.Shapes;  
using System.Windows.Controls.Primitives;  
using System.Diagnostics;
```

4. Opis projektu: Jest to popularna gra typu kółko i krzyżyk, która ma 4 wiersze i 4 kolumny. Wygrywa ten gracz, któremu się uda zdobyć 3 X albo 3 O obok siebie w linii prostej.

5. Opis funkcji i zmiennych:

`public enum Player` – Przechowuje rodzaje graczy.

`public Player currentPlayer` – Zmienna która wskazuje na to czyja teraz jest kolej.

`int count` – zmienna która zwraca informacje o tym ile razy do kratki zostały wpisane wartości X i O.

`public static int[,] winners` – Statyczna dwuwymiarowa tablica typu int, która przechowuje warunki zwycięstwa dla kratki 3x3.

`public MainWindow()` – Główna funkcja programu, w której oprócz inicjalizowania komponentu, jest też ustawiany domyślny gracz.

`private void ButtonClick(object sender, RoutedEventArgs e)` – Funkcja wywoływana kiedy jakiś gracz kliknie w kwadrat, po czym sprawdza czy stan tego obiektu. Jeśli był już wcześniej wciśnięty to funkcja nic nie robi, jeśli nie to do danego pola jest wpisywany X albo O, zależnie od aktualnego gracza. Funkcja ta po wpisaniu tej wartości wywołuje dwie inne funkcje(`CheckVictory()` i `ChangePlayer()`) i inkrementuje zmienną `count`.

`public void ChangePlayer()` – Funkcja, która zmienia aktualnego gracza. X na O i na odwrót zależnie co będzie pierwsze. Przy zmianie wywoływane są funkcje `ToggleO()` i `ToggleX()`.

`public void CheckVictory()` – Funkcja która ma za zadanie sprawdzić czy jest spełniony warunek zwycięstwa. By to osiągnąć używana jest pomocnicza funkcja `Check()`. W samej funkcji tworzone są listy obiektów, które są częścią tabeli kratki 4x4. Następnie są przekazywane do pomocniczej funkcji.

`public void Check(List<Button> list)` – Pomocnicza funkcja, która przyjmuje przekazany do niej element. Następnie używa go by sprawdzić warunek zwycięstwa. Jeśli jest spełniony to pojawia się nowy grid, który informuje o zwycięstwie wybranego uczestnika. W oknie tym znajduje się także przycisk Replay. W wypadku, gdy żaden z graczy nie odniósł zwycięstwa pojawia się okno z komunikatem o remisie.

`private void PlayerClick(object sender, RoutedEventArgs e)` – Funkcja która jest wywoływana po tym jak zostanie kliknięty kwadrat sygnalizujący o aktualnym graczu, poprzez podkreślenie. Po kliknięciu zmienia gracza na tego, który odpowiada danemu obiektowi.

`ToggleX()` i `ToggleO()` – Pomocnicze funkcje, które sygnalizują w odpowiednim kwadracie o aktualnym graczu.

`private void ReplayClick(object sender, RoutedEventArgs e)` – Funkcja, która aktywuje się po tym jak gracz wciśnie przycisk Replay. Resetowana jest plansza X i O, a także zmienna `count`. Po czym ustalany jest nowy gracz.

`public static IEnumerable<T> FindButtons<T>(DependencyObject root) where T : DependencyObject` – Funkcja, która zwraca listę. Do listy zbierane są wszystkie obiekty, które są typu `<T>`.