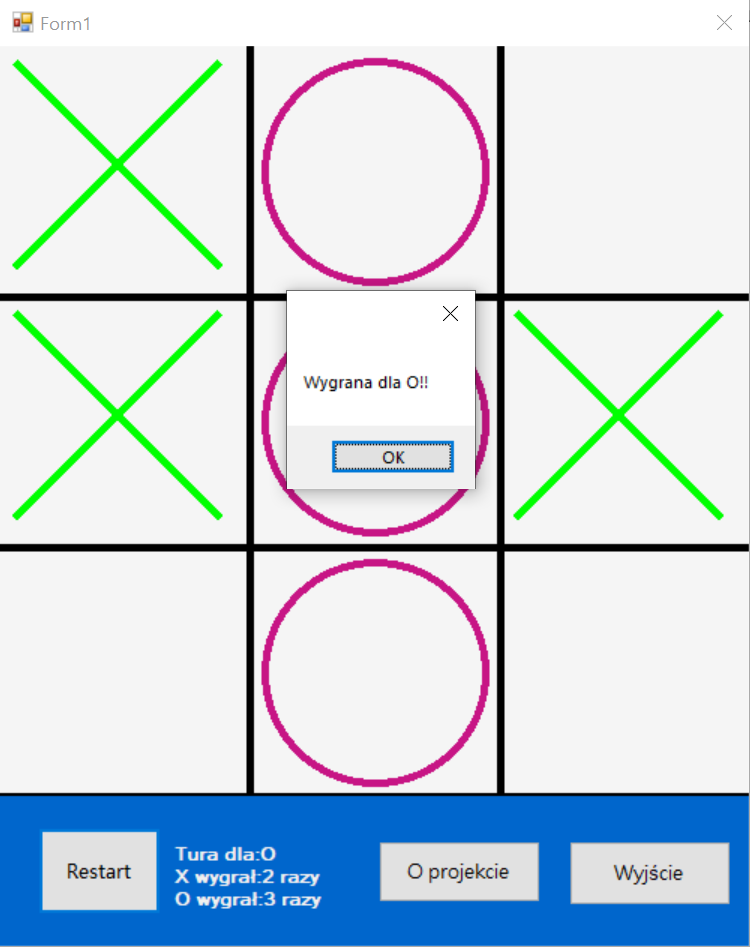
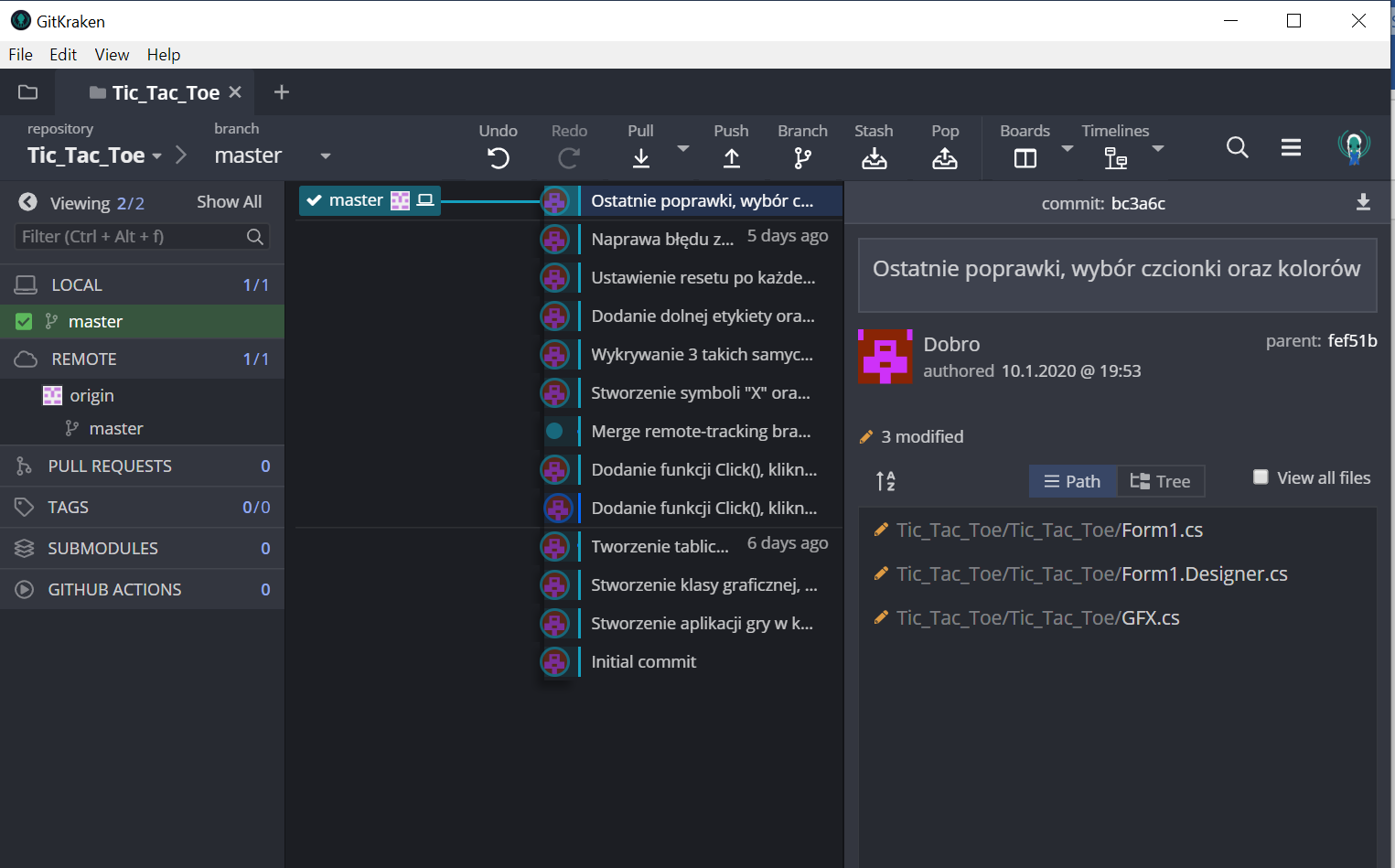
**Dokumentacja projektu**

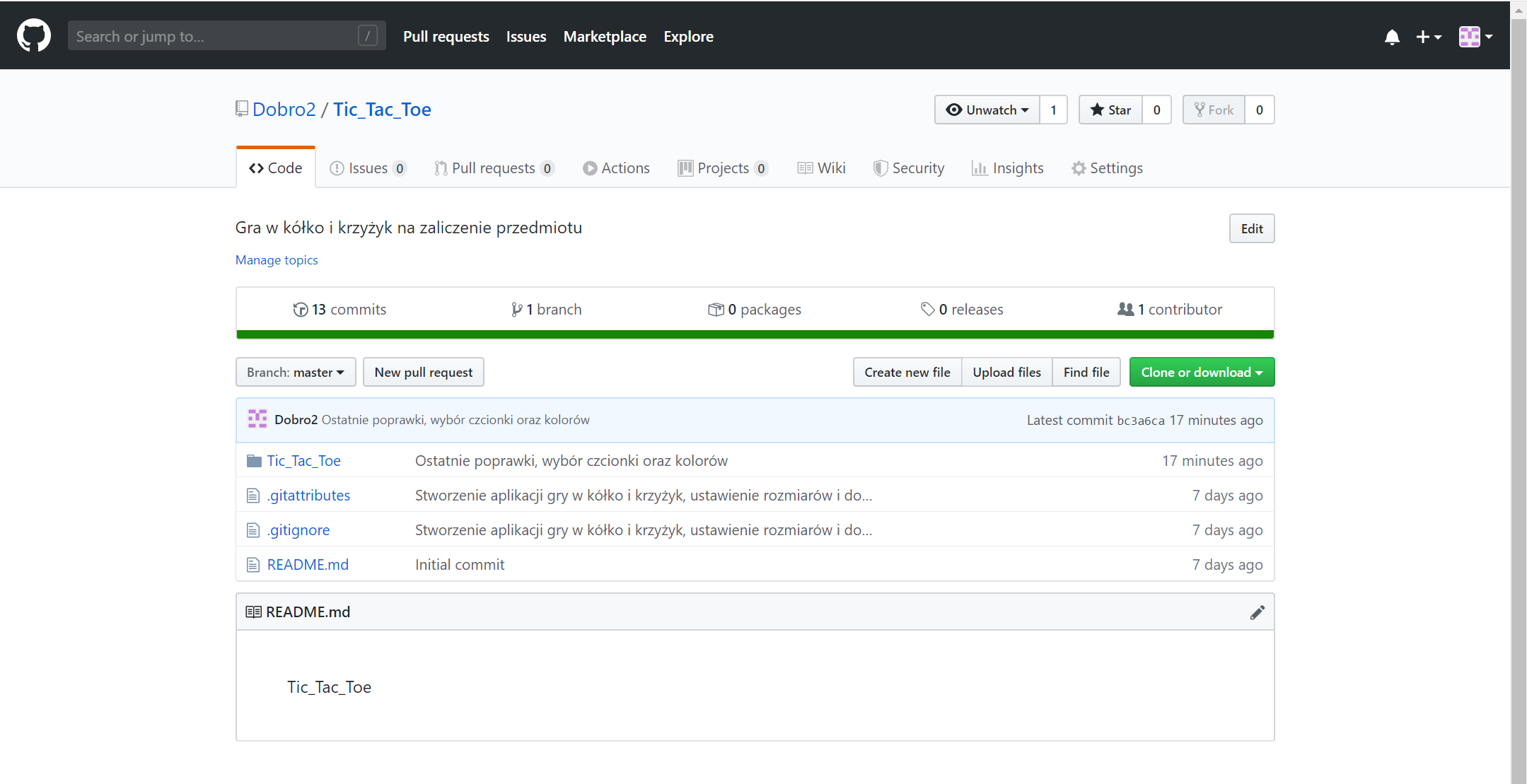
**I.Funkcjonalość programu:**

Gra w kółko i krzyżyk przeznaczona dla dwóch osób. Program podlicza punktację dla każdego gracza oraz pokazuję, który gracz ma obecnie ruch w turze. W przypadku remisu, planszę należy zresetować za pomocą przycisku Reset. Gra obejmuje 10 tur, po których pojawia się okienko z informacją o wygranej dla danego gracza. Dodatkowo po każdej turze, pojawia się informacja o zwycięzcy.

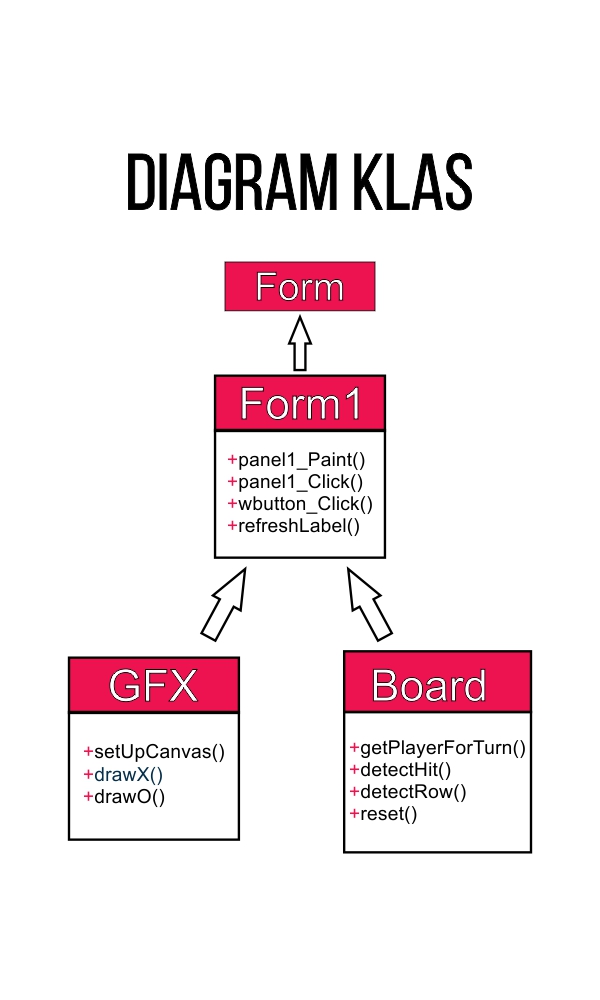


Program został napisany przy pomocy GitHub. Każdy umieszczony postęp, zmiana w programie została zarejestrowana.





**II. Diagram Klas**

****

**III. Metody**

Klasa GFX:

public static void setUpCanvas()

{

Brush bg = new SolidBrush(Color.WhiteSmoke);

Pen lines = new Pen(Color.Black, 5);

gObject.FillRectangle(bg, new Rectangle(0, 0,500,600));

gObject.DrawLine(lines, new Point(167, 0), new Point(167, 500));

gObject.DrawLine(lines, new Point(334, 0), new Point(334, 500));

gObject.DrawLine(lines, new Point(0, 167), new Point(500, 167));

gObject.DrawLine(lines, new Point(0, 334), new Point(500, 334));

gObject.DrawLine(lines, new Point(0, 500), new Point(500, 500));

}

Stworzenie pędźla oraz linii i narysowanie ich tak, aby tworzyły plansze

public static void drawX(Point loc)

{

Pen xPen = new Pen(Color.Lime, 5);

int xAbs = loc.X \* 167;

int yAbs = loc.Y \* 167;

gObject.DrawLine(xPen, xAbs+10, yAbs+10, xAbs + 147, yAbs + 147);

gObject.DrawLine(xPen, xAbs + 147, yAbs+10,xAbs+10, yAbs + 147);

}

public static void drawO(Point loc)

{

Pen oPen = new Pen(Color.MediumVioletRed,5);

int xAbs = loc.X \* 167;

int yAbs = loc.Y \* 167;

gObject.DrawEllipse(oPen,xAbs +10,yAbs + 10, 147,147);

}

Stworzenie znaku „X” oraz „O”

Klasa Board:

public int getPlayerForTurn()

{

return playersTurn;

}

Zwraca informacje o danej turze

public int getOwins()

{

return Owins;

}

Zwraca informacje o wygranych „O”

public int getXwins()

{

return Xwins;

}

Zwraca informacje o wygranych „X”

public void initBoard()

{

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

{

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

{

slots[x, y] = new Rectangle(x \* 167, y \* 167, 167, 167);

holders[x, y] = new Holder();

holders[x, y].setValue(B);

holders[x, y].setLocation(new Point(x, y));

}

}

}

Tworzenie planszy

public void detectHit(Point loc)

{

if (loc.Y <= 500)

{

int x = 0;

int y = 0;

if (loc.X < 167)

{

x = 0;

}

else if (loc.X > 167 && loc.X < 334)

{

x = 1;

}

else if (loc.X > 334)

{

x = 2;

}

if (loc.Y < 167)

{

y = 0;

}

else if (loc.Y > 167 && loc.Y < 334)

{

y = 1;

}

else if (loc.Y > 334 && loc.Y < 500)

{

y = 2;

}

movesMade++;

}

Metoda wykrywa kliknięcie myszki i przypisuje je do danego pola na planszy, zwiększa ruch gracza o 1,

if (movesMade % 2 == 0)

{

GFX.drawX(new Point(x, y));

holders[x, y].setValue(X);

if (detectRow())

{

MessageBox.Show("Brawo, wygrywa X!!");

Xwins++;

reset();

GFX.setUpCanvas();

}

playersTurn = O;

}

else

{

GFX.drawO(new Point(x, y));

holders[x, y].setValue(O);

if (detectRow())

{

MessageBox.Show("Wygrana dla O!!");

Owins++;

reset();

GFX.setUpCanvas();

}

playersTurn = X;

}

Metoda rysuje dany znak, jeśli zostaną wykryte 3 takie same znaki w linii pionowej, poziomej bądź pod skosem zwiększa punkt dla gracza oraz pokazuje informacje o wygranej w turze, następnie resetuje planszę i zaczyna nową turę

public bool detectRow()

{

bool isWon = false;

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

{

if (holders[x, 0].getValue() == X && holders[x, 1].getValue() == X && holders[x, 2].getValue() == X)

{

return true;

}

if (holders[x, 0].getValue() == O && holders[x, 1].getValue() == O && holders[x, 2].getValue() == O)

{

return true;

}

switch (x)

{

case 0:

if (holders[x, 0].getValue() == X && holders[x + 1, 1].getValue() == X && holders[x + 2, 2].getValue() == X)

{

return true;

}

if (holders[x, 0].getValue() == O && holders[x + 1, 1].getValue() == O && holders[x + 2, 2].getValue() == O)

{

return true;

}

break;

case 2:

if (holders[x, 0].getValue() == X && holders[x - 1, 1].getValue() == X && holders[x - 2, 2].getValue() == X)

{

return true;

}

if (holders[x, 0].getValue() == O && holders[x - 1, 1].getValue() == O && holders[x - 2, 2].getValue() == O)

{

return true;

}

break;

}

}

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

{

if (holders[0, y].getValue() == X && holders[1, y].getValue() == X && holders[2, y].getValue() == X)

{

return true;

}

if (holders[0, y].getValue() == O && holders[1, y].getValue() == O && holders[2, y].getValue() == O)

{

return true;

}

}

return isWon;

}

Metoda wykrywa 3 znaki w linii pionowej, poziomej bądź pod skosem

public void reset()

{

holders = new Holder[3, 3];

initBoard();

}

Metoda resetuje planszę

Klasa Form1:

public void refreshLabel()

{

String newText = "Tura dla:";

if (theBoard.getPlayerForTurn() == Board.X)

{

newText += "X";

}

else

{

newText += "O";

}

newText += "\n";

newText += "X wygrał:" + theBoard.getXwins() + " razy\n";

newText += "O wygrał:" + theBoard.getOwins() + " razy";

label1.Text = newText;

}

Informacje o danej turze oraz informacje końcowe