using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp5

{

public partial class Form1 : Form

{

Button[,] board;

int turn = 0;

int work = 0;

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

string s = textBox1.Text; //what is in the text box //MessageBox.Show(s); // show s

int number = int.Parse(s);

if (number < 6)

MessageBox.Show(" choose a bigger number ");

else if (number > 15)

MessageBox.Show(" choose a smaller number ");

if (number%2!=0)

{

MessageBox.Show(" choose an even number ");

}

else

{

board = new Button[number, number];

button1.Visible = false;

textBox1.Visible = false;

this.Height = panel1.Height + 700;

this.Width = panel1.Width +600;

int r, c;

r = 0;

int num = 0;

panel1.Width = number \* 70; // makes it squares

panel1.Height = number \* 70;

while (r < board.GetLength(0))

{

c = 0;

while (c < board.GetLength(1))

{

board[r, c] = new Button();

board[r, c].BackColor = Color.White;

board[r, c].Size = new Size(panel1.Width / number, panel1.Height / number);

board[r, c].Location = new Point(c \* panel1.Width / number, r \* panel1.Height / number);

board[r, c].Tag = num;// new thing to be abe to press the button

num++;

board[r, c].Click += Form1\_Click;

panel1.Controls.Add(board[r, c]);

c++;

}

r++;

}

board[number / 2, number / 2].Name = "x";

board[number / 2 - 1, number / 2 - 1].Name = "x";

board[number / 2, number / 2].BackColor = Color.Black;

board[number / 2 - 1, number / 2 - 1].BackColor = Color.Black;

board[number / 2-1, number / 2 - 1].Name = "o";

board[number / 2 + 1, number / 2].Name = "o";

board[number / 2, number / 2 -1].BackColor = Color.Red;

board[number / 2 - 1, number / 2].BackColor = Color.Red;

}

}

catch

{

MessageBox.Show("error"); // show s

}

}

//static int FindToeatB(int shura, int amuda,int shura1, int amuda1)// switch order of digits

//{

// //if (board[shura, amuda - 1].BackColor == Color.Black || board[shura, amuda + 1].BackColor == Color.Black || board[shura - 1, amuda - 1].BackColor == Color.Black || board[shura + 1, amuda - 1].BackColor == Color.Black || board[shura - 1, amuda].BackColor == Color.Black || board[shura + 1, amuda].BackColor == Color.Black)

// //{

// //}

// return shura;

//}

private void Form1\_Click(object sender, EventArgs e)

{

//throw new NotImplementedException();

string k = (((Button)(sender)).Tag).ToString();

int n = int.Parse(k);

int shura = n / board.GetLength(1); // finds point of button

int amuda = n % board.GetLength(1);

if (turn==0) // 0 is red //o

{

if (((Button)(sender)).BackColor==Color.White)

{

try

{

//if (board[shura, amuda - 1].BackColor == Color.Black || board[shura, amuda + 1].BackColor == Color.Black || board[shura - 1, amuda - 1].BackColor == Color.Black || board[shura + 1, amuda + 1].BackColor == Color.Black || board[shura - 1, amuda + 1].BackColor == Color.Black || board[shura + 1, amuda - 1].BackColor == Color.Black || board[shura - 1, amuda].BackColor == Color.Black || board[shura + 1, amuda].BackColor == Color.Black)

//{

int i = shura; int j = amuda;

if (board[shura, amuda - 1].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

j--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

j--;

}

work = 0;

}

else

MessageBox.Show("1");

}

if (board[shura, amuda + 1].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

j++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

j++;

}

}

work = 0;

}

if (board[shura-1, amuda ].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

i--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

i--;

}

}

work = 0;

}

if (board[shura + 1, amuda].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

i++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

i++;

}

}

work = 0;

}

if (board[shura + 1, amuda-1].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

j--;

i++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

//MessageBox.Show("ugh");

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

j--;

i++;

}

}

work = 0;

}

if (board[shura - 1, amuda - 1].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

j--;

i--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

j--;

i--;

}

}

work = 0;

}

if (board[shura + 1, amuda + 1].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

j++;

i++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

j++;

i++;

}

}

work = 0;

}

if (board[shura - 1, amuda + 1].BackColor == Color.Black)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

work = 1;

break;

}

j++;

i--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Red)

{

break;

}

if (board[i, j].BackColor == Color.Black)

{

board[i, j].BackColor = Color.Red;

board[shura, amuda].BackColor = Color.Red;

}

j++;

i--;

}

}

work = 0;

}

work = 0;

//}

}

catch { }

}

turn = 1;

}

else if (turn == 1) // 0 is red //o

{

if (((Button)(sender)).BackColor == Color.White)

{

try

{

//if (board[shura, amuda - 1].BackColor == Color.Red || board[shura, amuda + 1].BackColor == Color.Red || board[shura - 1, amuda - 1].BackColor == Color.Red|| board[shura - 1, amuda + 1].BackColor == Color.Red|| board[shura + 1, amuda + 1].BackColor == Color.Red || board[shura + 1, amuda - 1].BackColor == Color.Red || board[shura - 1, amuda].BackColor == Color.Red || board[shura + 1, amuda].BackColor == Color.Red)

//{

int i = shura; int j = amuda;

if (board[shura, amuda - 1].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

j--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

j--;

}

work = 0;

}

}

if (board[shura, amuda + 1].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

j++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

j++;

}

work = 0;

}

}

if (board[shura - 1, amuda].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

i--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

i--;

}

work = 0;

}

}

if (board[shura + 1, amuda].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

i++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

i++;

}

work = 0;

}

}

if (board[shura + 1, amuda - 1].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

j--;

i++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

j--;

i++;

}

work = 0;

}

}

if (board[shura - 1, amuda - 1].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

j--;

i--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

j--;

i--;

}

work = 0;

}

}

if (board[shura + 1, amuda + 1].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

j++;

i++;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

j++;

i++;

}

work = 0;

}

}

if (board[shura - 1, amuda + 1].BackColor == Color.Red)

{

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

work = 1;

break;

}

j++;

i--;

}

if (work == 1)

{

i = shura; j = amuda;

while (i < board.GetLength(1))

{

if (board[i, j].BackColor == Color.Black)

{

break;

}

if (board[i, j].BackColor == Color.Red)

{

board[i, j].BackColor = Color.Black;

board[shura, amuda].BackColor = Color.Black;

}

j++;

i--;

}

work = 0;

}

}

//}

}

catch { }

}

turn = 0;

}

}

}

}