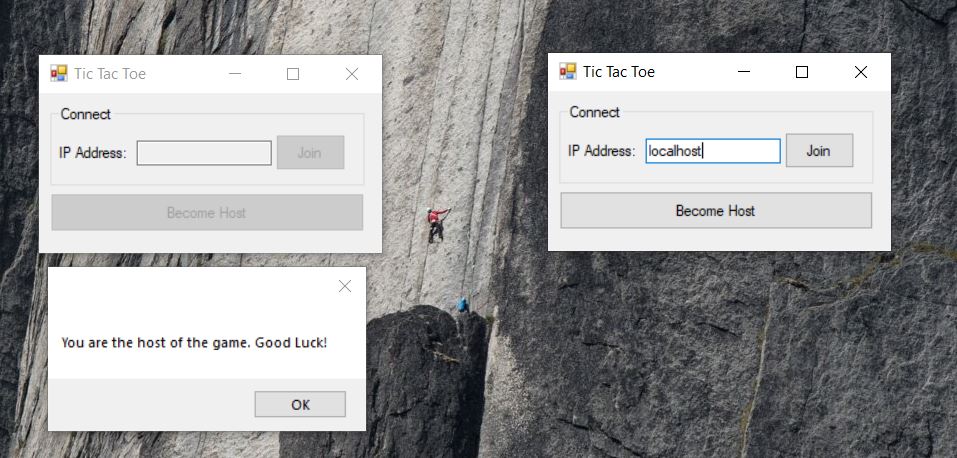
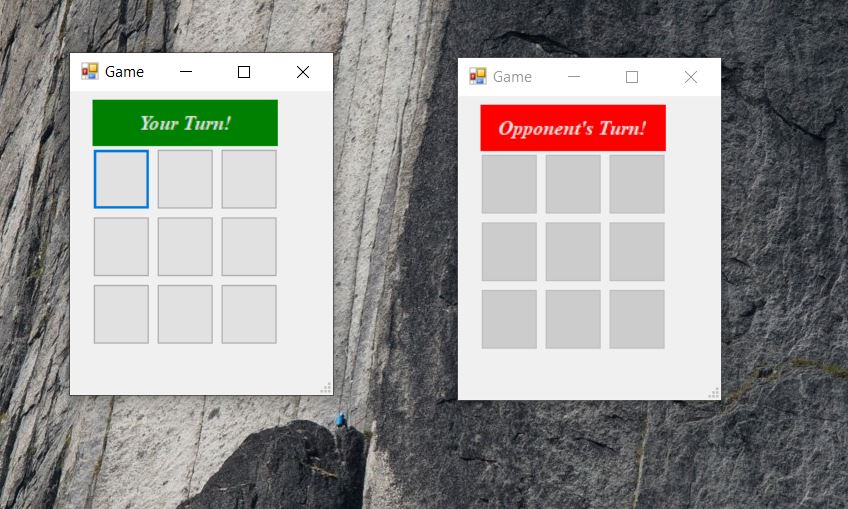
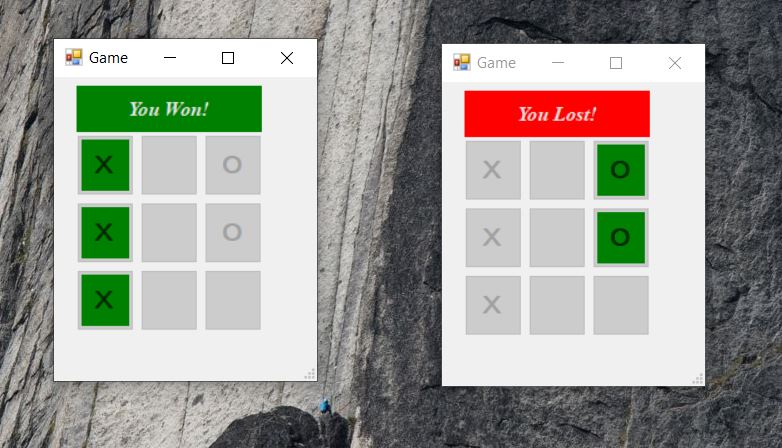
**Screen Captures with Brief Description**



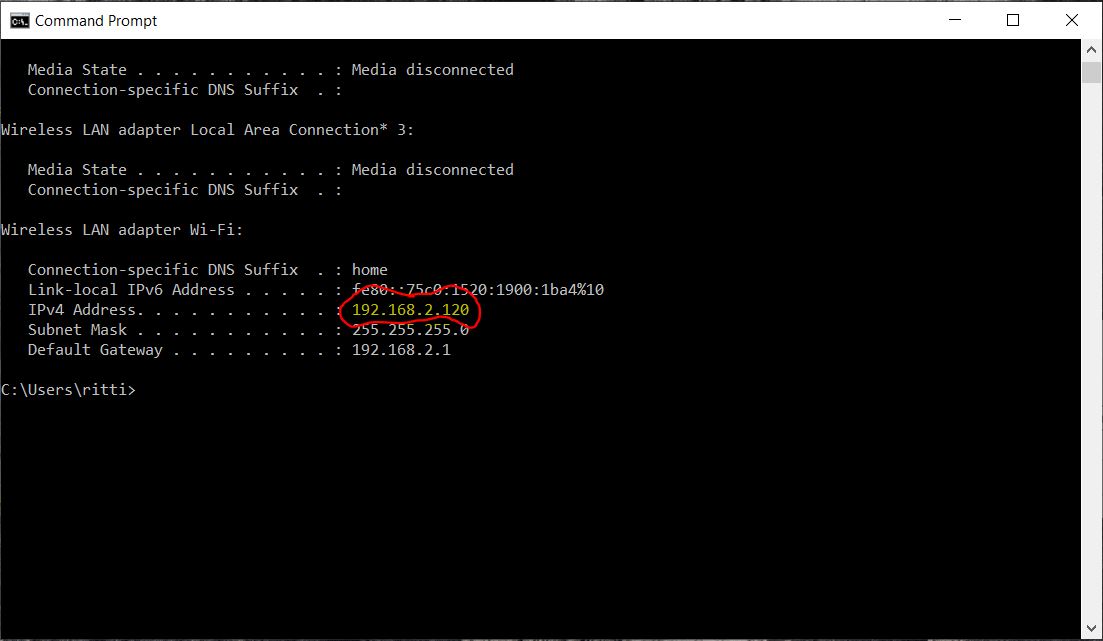
1. Server and client connected to each other.



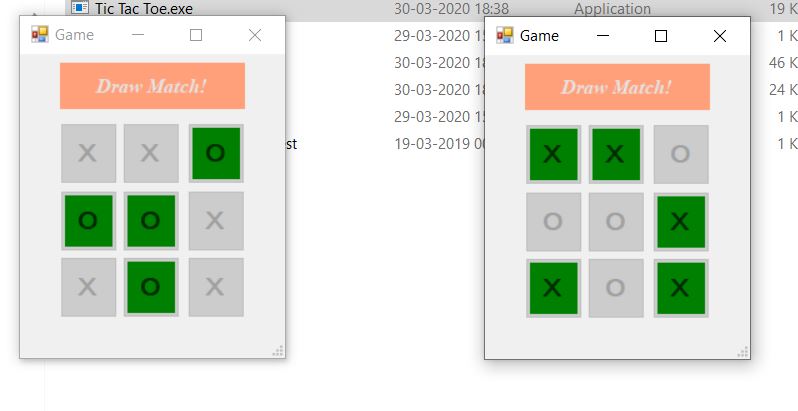
1. Start game page.



1. Win/Lost situation.



1. Local machine Ip address, client can connect to server by the Ip address as well.



1. Draw match situation.

**Source Code**

**Form1.cs**

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 Tic\_Tac\_Toe

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

if (textBox1.Text != "")

{

TicTacToe game = new TicTacToe(false, textBox1.Text);

Visible = false;

if (!game.IsDisposed)

game.ShowDialog();

if (game.endGame == true)

Application.Exit();

Visible = true;

}

else

MessageBox.Show("Enter valid IP Address or localhost");

}

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

{

textBox1.Enabled = false;

btnJoin.Enabled = false;

btnHost.Enabled = false;

MessageBox.Show("You are the host of the game. Good Luck!");

TicTacToe game = new TicTacToe(true);

Visible = false;

if (!game.IsDisposed)

game.ShowDialog();

if (game.endGame == true)

Application.Exit();

Visible = true;

}

}

}

**TicTacToe.cs**

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;

using System.Threading;

using System.Net.Sockets;

using System.Media;

namespace Tic\_Tac\_Toe

{

public partial class TicTacToe : Form

{

private char PlayerChar;

private char OpponentChar;

private Socket sock;

private BackgroundWorker Receiver = new BackgroundWorker();

private TcpListener server = null;

private TcpClient client;

public bool endGame = false;

private void playSound()

{

SoundPlayer splayer = new SoundPlayer(@"sound.wav");

splayer.Play();

}

private void finalSound()

{

SoundPlayer splayer = new SoundPlayer(@"final.wav");

splayer.Play();

}

public TicTacToe(bool isHost, string ip = null)

{

InitializeComponent();

Receiver.DoWork += Receiver\_Work;

CheckForIllegalCrossThreadCalls = false;

if (isHost)

{

PlayerChar = 'X';

OpponentChar = 'O';

server = new TcpListener(System.Net.IPAddress.Any, 4545);

server.Start();

sock = server.AcceptSocket();

}

else

{

PlayerChar = 'O';

OpponentChar = 'X';

try

{

client = new TcpClient(ip, 4545);

sock = client.Client;

Receiver.RunWorkerAsync();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

Close();

}

}

}

private void Receiver\_Work(object sender, DoWorkEventArgs e)

{

if (CurrentState())

return;

FreezeButtons();

lblTurn.BackColor = Color.Red;

lblTurn.Text = "Opponent's Turn!";

Turn();

lblTurn.BackColor = Color.Green;

lblTurn.Text = "Your Turn!";

if (!CurrentState())

UnfreezeButtons();

}

private void btn1\_Click(object sender, EventArgs e)

{

playSound();

btn1.BackColor = Color.Green;

try

{

Thread t1 = new Thread(() =>

{

byte[] number = { 1 };

sock.Send(number);

btn1.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t1.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void Turn()

{

byte[] buffer = new byte[1];

sock.Receive(buffer);

if (buffer[0] == 1)

btn1.Text = OpponentChar.ToString();

if (buffer[0] == 2)

btn2.Text = OpponentChar.ToString();

if (buffer[0] == 3)

btn3.Text = OpponentChar.ToString();

if (buffer[0] == 4)

btn4.Text = OpponentChar.ToString();

if (buffer[0] == 5)

btn5.Text = OpponentChar.ToString();

if (buffer[0] == 6)

btn6.Text = OpponentChar.ToString();

if (buffer[0] == 7)

btn7.Text = OpponentChar.ToString();

if (buffer[0] == 8)

btn8.Text = OpponentChar.ToString();

if (buffer[0] == 9)

btn9.Text = OpponentChar.ToString();

}

private void btn2\_Click(object sender, EventArgs e)

{

playSound();

btn2.BackColor = Color.Green;

try

{

Thread t2 = new Thread(() =>

{

byte[] number = { 2 };

sock.Send(number);

btn2.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t2.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void btn3\_Click(object sender, EventArgs e)

{

playSound();

btn3.BackColor = Color.Green;

try

{

Thread t3 = new Thread(() =>

{

byte[] number = { 3 };

sock.Send(number);

btn3.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t3.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void btn4\_Click(object sender, EventArgs e)

{

playSound();

btn4.BackColor = Color.Green;

try

{

Thread t4 = new Thread(() =>

{

byte[] number = { 4 };

sock.Send(number);

btn4.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t4.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void btn5\_Click(object sender, EventArgs e)

{

playSound();

btn5.BackColor = Color.Green;

try

{

Thread t5 = new Thread(() =>

{

byte[] number = { 5 };

sock.Send(number);

btn5.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t5.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void btn6\_Click(object sender, EventArgs e)

{

playSound();

btn6.BackColor = Color.Green;

try

{

Thread t6 = new Thread(() =>

{

byte[] number = { 6 };

sock.Send(number);

btn6.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t6.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void btn7\_Click(object sender, EventArgs e)

{

playSound();

btn7.BackColor = Color.Green;

try

{

Thread t7 = new Thread(() =>

{

byte[] number = { 7 };

sock.Send(number);

btn7.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t7.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void btn8\_Click(object sender, EventArgs e)

{

playSound();

btn8.BackColor = Color.Green;

try

{

Thread t8 = new Thread(() =>

{

byte[] number = { 8 };

sock.Send(number);

btn8.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t8.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void btn9\_Click(object sender, EventArgs e)

{

playSound();

btn9.BackColor = Color.Green;

try

{

Thread t9 = new Thread(() =>

{

byte[] number = { 9 };

sock.Send(number);

btn9.Text = PlayerChar.ToString();

Receiver.RunWorkerAsync();

Thread.Sleep(1000);

});

t9.Start();

}

catch (Exception ex)

{

Console.WriteLine(ex);

}

}

private void FreezeButtons()

{

btn1.Enabled = false;

btn2.Enabled = false;

btn3.Enabled = false;

btn4.Enabled = false;

btn5.Enabled = false;

btn6.Enabled = false;

btn7.Enabled = false;

btn8.Enabled = false;

btn9.Enabled = false;

}

private void UnfreezeButtons()

{

if (btn1.Text == "")

btn1.Enabled = true;

if (btn2.Text == "")

btn2.Enabled = true;

if (btn3.Text == "")

btn3.Enabled = true;

if (btn4.Text == "")

btn4.Enabled = true;

if (btn5.Text == "")

btn5.Enabled = true;

if (btn6.Text == "")

btn6.Enabled = true;

if (btn7.Text == "")

btn7.Enabled = true;

if (btn8.Text == "")

btn8.Enabled = true;

if (btn9.Text == "")

btn9.Enabled = true;

}

private void youWon()

{

finalSound();

lblTurn.BackColor = Color.Green;

lblTurn.Text = "You Won!";

FreezeButtons();

Thread.Sleep(15000);

endGame = true;

Application.Exit();

}

private void youLost()

{

finalSound();

lblTurn.BackColor = Color.Red;

lblTurn.Text = "You Lost!";

FreezeButtons();

Thread.Sleep(15000);

endGame = true;

Application.Exit();

}

private void DrawMatch()

{

finalSound();

lblTurn.BackColor = Color.LightSalmon;

lblTurn.Text = "Draw Match!";

FreezeButtons();

Thread.Sleep(15000);

endGame = true;

Application.Exit();

}

private bool CurrentState()

{

//Horizontals

if (btn1.Text == btn2.Text && btn2.Text == btn3.Text && btn3.Text != "")

{

if (btn1.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

else if (btn4.Text == btn5.Text && btn5.Text == btn6.Text && btn6.Text != "")

{

if (btn4.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

else if (btn7.Text == btn8.Text && btn8.Text == btn9.Text && btn9.Text != "")

{

if (btn7.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

//Verticals

else if (btn1.Text == btn4.Text && btn4.Text == btn7.Text && btn7.Text != "")

{

if (btn1.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

else if (btn2.Text == btn5.Text && btn5.Text == btn8.Text && btn8.Text != "")

{

if (btn2.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

else if (btn3.Text == btn6.Text && btn6.Text == btn9.Text && btn9.Text != "")

{

if (btn3.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

else if (btn1.Text == btn5.Text && btn5.Text == btn9.Text && btn9.Text != "")

{

if (btn1.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

else if (btn3.Text == btn5.Text && btn5.Text == btn7.Text && btn7.Text != "")

{

if (btn3.Text[0] == PlayerChar)

youWon();

else

youLost();

return true;

}

//Draw

else if (btn1.Text != "" && btn2.Text != "" && btn3.Text != "" && btn4.Text != "" && btn5.Text != "" && btn6.Text != "" && btn7.Text != "" && btn8.Text != "" && btn9.Text != "")

{

DrawMatch();

return true;

}

return false;

}

private void TicTacToe\_FormClosing(object sender, FormClosingEventArgs e)

{

Receiver.WorkerSupportsCancellation = true;

Receiver.CancelAsync();

if (server != null)

server.Stop();

this.Hide();

this.Parent = null;

e.Cancel = true;

}

}

}