**CELAL BAYAR ÜNİVERSİTESİ**

**HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ**

**YAZILIM SINAMA DERSİ**

**ÖDEV 1**

**1.Tic-Tac-Toe Oyunu**

**1.1.Amaç...............................................................2**

**1.2.Mantığı............................................................2**

**1.3.Program Kodu...................................................2-25**

**1.4.Ekran Çıktısı....................................................26-30**

**1.5.Test Caseler....................................................30-31**

**Hazırlayanlar: Cennet Gizem Söylemez(152805010)**

**Bekir Furkan Zadegil(152805009)**

**1.1.Amaç:**

Kullanıcı tarafından oluşturulan oyunda 3 seçenek vardır. Bilgisayara karşı oynama ,network üzerinden karşılıklı, aynı bilgisayardan karşılıklı. Oyuncu hangisinde oynamak istiyorsa onu seçer ‘X’ ve ’ O’ kutucuklara koyulur 3 yapan kazanır.

**1.2.Mantığı:**

Oyunun mantığı hem hızlı olmak hem de mantıklı düşünüp iyi hamleler yapmak. Öncelikle bilgisayara karşı oynamak (yapay zeka) bir hamleyi bizim yapıp sonra diğer hamleyi bilgisayarın yapmasıdır. Aynı bilgisayardan karşılıklı oynamak ise aynı ağ üzerinden iki kişinin oynamasıdır. Bir de aynı bilgisayardan karşılıklı var bu aynı ağ gerektirmeden oynan 2 kişi aynı oyunu aynı bilgisayrdan oynuyor. Bunların dışında oyun berabere bitti oyunu berabere bitirmek yerine oyunda bir süremiz var hangi oyuncu süresini daha verimli kullanırsa o kazanıyor .

**1.3.Program Kodu**

**Ana Form**

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 XOX\_Games

{

public partial class AnaForm : Form

{

public AnaForm()

{

InitializeComponent();

}

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

{

if(rbtnYeniServer.Checked ==true)

{

frmServer frm = new frmServer();

frm.Show();

}

if(rbtnAgaKatil.Checked == true)

{

frmClient frm2 = new frmClient();

frm2.Show();

}

}

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

{

groupBox2.Visible = false;

groupBox3.Visible = false;

}

private void rbtnbotakarsi\_CheckedChanged(object sender, EventArgs e)

{

groupBox2.Visible = true;

groupBox3.Visible = false;

}

private void rbtnIkiKisilik\_CheckedChanged(object sender, EventArgs e)

{

groupBox2.Visible = true;

groupBox3.Visible = false;

}

private void rbtnLan\_CheckedChanged(object sender, EventArgs e)

{

groupBox2.Visible = false;

groupBox3.Visible = true;

}

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

{

if (rbtnbotakarsi.Checked == true)

{

frmYapayZeka frm3 = new frmYapayZeka();

frm3.Show();

}

if (rbtnIkiKisilik.Checked == true)

{

frmikiKisi frm4 = new frmikiKisi();

frm4.Show();

}

}

}

}

**From Client**

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.Net;

using System.Net.Sockets;

namespace XOX\_Games

{

public partial class frmClient : Form

{

Socket \_socket;

byte[] buffer = new byte[1024];

public frmClient()

{

InitializeComponent();

\_socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

}

private void frmClient\_Load(object sender, EventArgs e)

{

\_socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

panel1.Hide();

setNames();

}

bool game;

string p1Name, p2Name;

char player = 'X';

int xScore = 1, oScore = 1;

List<int> X = new List<int>();

List<int> O = new List<int>();

public void Connect(string ipAddress, int port)

{

\_socket.BeginConnect(new IPEndPoint(IPAddress.Parse(ipAddress), port), ConnectCallBack, null);

}

private void ConnectCallBack(IAsyncResult result)

{

if (\_socket.Connected)

{

Invoke(new Action(() => lbConnection.Text = "Connected"));

Invoke(new Action(() => panel1.Show()));

buffer = new byte[1024];

\_socket.BeginReceive(buffer, 0, buffer.Length, SocketFlags.None, RecieveCallBack, null);

}

}

private void RecieveCallBack(IAsyncResult result)

{

try

{

int bufferSize = \_socket.EndReceive(result);

byte[] packet = new byte[bufferSize];

Array.Copy(buffer, packet, packet.Length);

string theMessageToReceive = Encoding.ASCII.GetString(packet);

if (theMessageToReceive.Contains("coordinate"))

{

check(int.Parse(theMessageToReceive.Replace("coordinate", "")));

}

buffer = new byte[1024];

\_socket.BeginReceive(buffer, 0, buffer.Length, SocketFlags.None, RecieveCallBack, null);

}

catch { }

}

public void SendMsg(byte[] msg)

{

\_socket.Send(msg);

}

public void setNames()

{

p1Name = "Server";

p2Name = "Client";

}

void check(int getSlotNr)

{

game = true;

this.panel1.Controls["pbSlot" + getSlotNr.ToString()].Text = player.ToString();

this.panel1.Controls["pbSlot" + getSlotNr.ToString()].Enabled = false;

if (player == 'X') X.Add(getSlotNr);

else O.Add(getSlotNr);

result();

if (game == true) player = (player == 'X') ? 'O' : 'X';

}

private void pbSlot1\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(1);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate1"));

}

}

private void pbSlot2\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(2);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate2"));

}

}

private void pbSlot3\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(3);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate3"));

}

}

private void pbSlot4\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(4);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate4"));

}

}

private void pbSlot5\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(5);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate5"));

}

}

private void pbSlot6\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(6);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate6"));

}

}

private void pbSlot7\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(7);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate7"));

}

}

private void pbSlot8\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(8);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate8"));

}

}

private void pbSlot9\_Click(object sender, EventArgs e)

{

if (player == 'O')

{

check(9);

\_socket.Send(Encoding.UTF8.GetBytes("coordinate9"));

}

}

private void button2\_Click(object sender, EventArgs e)

{

Connect(tbIP.Text, int.Parse(tbPort.Text));

button2.Enabled = false;

}

public void result()

{

int[,] win = new int[,] {

{1,2,3}, {4,5,6}, {7,8,9},

{1,4,7}, {2,5,8}, {3,6,9},

{1,5,9}, {7,5,3}

};

List<int>[] p = new List<int>[] { X, O };

int listPos = (player == 'X') ? 0 : 1;

for (int i = 0; i < 8; i++)

{

if (p[listPos].Contains(win[i, 0]) && p[listPos].Contains(win[i, 1]) && p[listPos].Contains(win[i, 2]))

{

((Label)this.panel1.Controls["lblP" + player.ToString() + "Score"]).Text = ((player == 'X') ? xScore++ : oScore++).ToString();

string winner = (player == 'X') ? p1Name : p2Name;

MessageBox.Show(winner + " won!", "Result");

clearBoard();

}

}

if (X.Count + O.Count == 9)

{

MessageBox.Show("It's a draw!", "Result");

clearBoard();

}

}

void clearBoard()

{

for (int i = 1; i <= 9; i++)

{

this.panel1.Controls["pbSlot" + i.ToString()].Text = "";

this.panel1.Controls["pbSlot" + i.ToString()].Enabled = true;

}

game = false;

player = 'X';

X.Clear();

O.Clear();

}

}

}

**İki Kişilik Form**

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 XOX\_Games

{

public partial class frmikiKisi : Form

{

public frmikiKisi()

{

InitializeComponent();

}

Button b;

bool turn = true;

int turn\_count = 0;

int saniye1, saniye2, toplamsaniye1, toplamsaniye2 = 0;

private void timer1\_Tick(object sender, EventArgs e)

{

saniye1 = saniye1 + 1;

toplamsaniye1 = toplamsaniye1 + saniye1;

saniye1 = 0;

lblPlayer1Timer.Text = toplamsaniye1.ToString();

}

private void frmikiKisi\_Load(object sender, EventArgs e)

{

}

private void timer2\_Tick(object sender, EventArgs e)

{

saniye2 = saniye2 + 1;

toplamsaniye2 = toplamsaniye2 + saniye2;

saniye2 = 0;

lblPlayer2Timer.Text = toplamsaniye2.ToString();

}

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

{

b = (Button)sender;

if (turn)

{

b.Text = "X";

timer2.Interval = 1000;

timer2.Start();

timer1.Stop();

}

else

{

b.Text = "O";

timer1.Interval = 1000;

timer1.Start();

timer2.Stop();

}

turn = !turn;

b.Enabled = false;

turn\_count++;

Kazananlar();

}

private void Kazananlar()

{

bool kazanan = false;

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

kazanan = true;

if ((btn4.Text == btn5.Text) && (btn5.Text == btn6.Text) && (!btn4.Enabled))

kazanan = true;

if ((btn7.Text == btn8.Text) && (btn8.Text == btn9.Text) && (!btn7.Enabled))

kazanan = true;

if ((btn1.Text == btn4.Text) && (btn4.Text == btn7.Text) && (!btn1.Enabled))

kazanan = true;

if ((btn2.Text == btn5.Text) && (btn5.Text == btn8.Text) && (!btn5.Enabled))

kazanan = true;

if ((btn3.Text == btn6.Text) && (btn6.Text == btn9.Text) && (!btn3.Enabled))

kazanan = true;

if ((btn1.Text == btn5.Text) && (btn5.Text == btn9.Text) && (!btn1.Enabled))

kazanan = true;

if ((btn3.Text == btn5.Text) && (btn5.Text == btn7.Text) && (!btn3.Enabled))

kazanan = true;

if (kazanan)

{

string winner = "";

if (turn)

winner = "O";

else

winner = "X";

timer1.Stop();

timer2.Stop();

MessageBox.Show(winner + "kazanan");

}

else

{

if (turn\_count == 9)

{

timer1.Stop();

timer2.Stop();

if (toplamsaniye1 < toplamsaniye2)

MessageBox.Show("X kazanan(süreyle)");

else if (toplamsaniye1 > toplamsaniye2)

MessageBox.Show("O kazanan(süreyle)");

else

MessageBox.Show("berabere(süreler eşit)");

timer1.Stop();

timer2.Stop();

}

}

}

}

}

**Server Form**

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.Net;

using System.Net.Sockets;

namespace XOX\_Games

{

public partial class frmServer : Form

{

Socket \_socket;

Socket clientSocket;

byte[] buffer = new byte[1024];

public frmServer()

{

InitializeComponent();

\_socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

}

private void frmServer\_Load(object sender, EventArgs e)

{

panel1.Hide();

Bind(6556);

Listen(500);

Accept();

setNames();

timer1.Interval = 1000;

}

bool game;

string p1Name, p2Name;

char player = 'X';

int xScore = 1, oScore = 1;

List<int> X = new List<int>();

List<int> O = new List<int>();

public void Bind(int port)

{

\_socket.Bind(new IPEndPoint(IPAddress.Any, port));

}

public void Listen(int backlog)

{

\_socket.Listen(backlog);

}

public void Accept()

{

\_socket.BeginAccept(AcceptCallBack, null);

}

private void AcceptCallBack(IAsyncResult result)

{

this.Invoke(new Action(() => State.Text = "Client Connected!"));

this.Invoke(new Action(() => panel1.Show()));

clientSocket = \_socket.EndAccept(result);

buffer = new byte[1024];

Accept();

clientSocket.BeginReceive(buffer, 0, buffer.Length, SocketFlags.None, RecieveCallBack, clientSocket);

}

private void RecieveCallBack(IAsyncResult result)

{

clientSocket = result.AsyncState as Socket;

int bufferSize = clientSocket.EndReceive(result);

byte[] packet = new byte[bufferSize];

Array.Copy(buffer, packet, packet.Length);

string msg = Encoding.ASCII.GetString(packet);

if (msg.Contains("coordinate"))

{

check(int.Parse(msg.Replace("coordinate", "")));

}

buffer = new byte[1024];

clientSocket.BeginReceive(buffer, 0, buffer.Length, SocketFlags.None, RecieveCallBack, clientSocket);

}

public void SendMsg(byte[] msg)

{

clientSocket.Send(msg);

timer1.Start();

}

public void setNames()

{

p1Name = "Server";

p2Name = "Client";

lblP1Name.Text = p1Name + ":";

lblP2Name.Text = p2Name + ":";

}

int secim = 0;

void check(int getSlotNr)

{

game = true;

this.panel1.Controls["pbSlot" + getSlotNr.ToString()].Text = player.ToString();

this.panel1.Controls["pbSlot" + getSlotNr.ToString()].Enabled = false;

if (player == 'X') X.Add(getSlotNr);

else O.Add(getSlotNr);

result();

if (game == true)

{

if (player == 'X')

{

player = 'O';

secim = 1;

timer1.Stop();

}

else

{

player = 'X';

secim = 0;

timer1.Start();

}

}

}

int saniye = 0, toplamsaniye = 0;

private void timer1\_Tick(object sender, EventArgs e)

{

saniye = saniye + 1;

toplamsaniye = toplamsaniye + saniye;

saniye = 0;

label1.Text = toplamsaniye.ToString();

}

public void result()

{

int[,] win = new int[,] {

{1,2,3}, {4,5,6}, {7,8,9},

{1,4,7}, {2,5,8}, {3,6,9},

{1,5,9}, {7,5,3}

};

List<int>[] p = new List<int>[] { X, O };

int listPos = (player == 'X') ? 0 : 1;

for (int i = 0; i < 8; i++)

{

if (p[listPos].Contains(win[i, 0]) && p[listPos].Contains(win[i, 1]) && p[listPos].Contains(win[i, 2]))

{

((Label)this.panel1.Controls["lblP" + player.ToString() + "Score"]).Text = ((player == 'X') ? xScore++ : oScore++).ToString();

string winner = (player == 'X') ? p1Name : p2Name;

MessageBox.Show(winner + " won!", "Result");

clearBoard();

}

}

if (X.Count + O.Count == 9)

{

MessageBox.Show("It's a draw!", "Result");

clearBoard();

}

}

void clearBoard()

{

for (int i = 1; i <= 9; i++)

{

this.panel1.Controls["pbSlot" + i.ToString()].Text = "";

this.panel1.Controls["pbSlot" + i.ToString()].Enabled = true;

}

game = false;

player = 'X';

X.Clear();

O.Clear();

}

private void pbSlot1\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(1);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate1"));

}

}

private void pbSlot2\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(2);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate2"));

}

}

private void pbSlot3\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(3);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate3"));

}

}

private void pbSlot4\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(4);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate4"));

}

}

private void pbSlot5\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(5);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate5"));

}

}

private void pbSlot6\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(6);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate6"));

}

}

private void pbSlot7\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(7);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate7"));

}

}

private void pbSlot8\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(8);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate8"));

}

}

private void pbSlot9\_Click(object sender, EventArgs e)

{

if (player == 'X')

{

check(9);

clientSocket.Send(Encoding.UTF8.GetBytes("coordinate9"));

}

}

}

}

**Yapay Zeka Form**

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 XOX\_Games

{

public partial class frmYapayZeka : Form

{

public frmYapayZeka()

{

InitializeComponent();

}

Button b;

bool turn = true;

int turn\_count = 0;

int saniye1, saniye2, toplamsaniye1, toplamsaniye2 = 0;

int playerskor = 0, cpuskor = 0, berabere = 0;

private void computermakemove()

{

Button move = null;

move = Look\_for\_win\_or\_block("O");

if(move==null)

{

move = Look\_for\_win\_or\_block("X");

if(move==null)

{

move = look\_for\_corner();

if (move == null)

{

look\_for\_open\_space();

}

}

}

move.PerformClick();

}

private Button Look\_for\_win\_or\_block(string mark)

{

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

{

return btn3;

}

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

{

return btn1;

}

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

{

return btn2;

}

if (btn4.Text == mark && btn5.Text == mark && btn6.Text == "")

{

return btn6;

}

if (btn5.Text == mark && btn6.Text == mark && btn4.Text == "")

{

return btn4;

}

if (btn4.Text == mark && btn6.Text == mark && btn5.Text == "")

{

return btn5;

}

if (btn7.Text == mark && btn8.Text == mark && btn9.Text == "")

{

return btn9;

}

if (btn8.Text == mark && btn9.Text == mark && btn7.Text == "")

{

return btn7;

}

if (btn7.Text == mark && btn9.Text == mark && btn8.Text == "")

{

return btn8;

}

if(btn1.Text==mark && btn4.Text==mark &&btn7.Text=="")

{

return btn7;

}

if (btn4.Text == mark && btn7.Text == mark && btn1.Text == "")

{

return btn1;

}

if (btn1.Text == mark && btn7.Text == mark && btn4.Text == "")

{

return btn4;

}

if (btn2.Text == mark && btn5.Text == mark && btn8.Text == "")

{

return btn8;

}

if (btn5.Text == mark && btn8.Text == mark && btn2.Text == "")

{

return btn2;

}

if (btn2.Text == mark && btn8.Text == mark && btn5.Text == "")

{

return btn5;

}

if (btn3.Text == mark && btn6.Text == mark && btn9.Text == "")

{

return btn9;

}

if (btn6.Text == mark && btn9.Text == mark && btn3.Text == "")

{

return btn3;

}

if (btn3.Text == mark && btn9.Text == mark && btn6.Text == "")

{

return btn6;

}

if (btn1.Text == mark && btn5.Text == mark && btn9.Text == "")

{

return btn9;

}

if (btn5.Text == mark && btn9.Text == mark && btn1.Text == "")

{

return btn1;

}

if (btn1.Text == mark && btn9.Text == mark && btn5.Text == "")

{

return btn5;

}

if (btn3.Text == mark && btn5.Text == mark && btn7.Text == "")

{

return btn7;

}

if (btn3.Text == mark && btn7.Text == mark && btn5.Text == "")

{

return btn5;

}

if (btn5.Text == mark && btn7.Text == mark && btn3.Text == "")

{

return btn3;

}

return null;

}

private Button look\_for\_corner()

{

if(btn1.Text=="O")

{

if (btn3.Text == "")

return btn3;

if (btn7.Text == "")

return btn7;

if (btn9.Text == "")

return btn9;

}

if (btn3.Text == "O")

{

if (btn1.Text == "")

return btn1;

if (btn7.Text == "")

return btn7;

if (btn9.Text == "")

return btn9;

}

if (btn9.Text == "O")

{

if (btn1.Text == "")

return btn1;

if (btn7.Text == "")

return btn7;

if (btn3.Text == "")

return btn3;

}

if (btn7.Text == "O")

{

if (btn1.Text == "")

return btn1;

if (btn3.Text == "")

return btn3;

if (btn9.Text == "")

return btn9;

}

if(btn1.Text=="")

{

return btn1;

}

if (btn3.Text == "")

{

return btn3;

}

if (btn7.Text == "")

{

return btn7;

}

if (btn9.Text == "")

{

return btn9;

}

return null;

}

private void frmYapayZeka\_Load(object sender, EventArgs e)

{

}

private Button look\_for\_open\_space()

{

Button b = null;

foreach(Control c in Controls)

{

b = c as Button;

if(b!=null)

{

if(b.Text == "")

{

return b;

}

}

}

return null;

}

private void timer1\_Tick(object sender, EventArgs e)

{

saniye1 = saniye1 + 1;

toplamsaniye1 = toplamsaniye1 + saniye1;

saniye1 = 0;

saniye2 = 0;

saniye2 = saniye2 + 1;

toplamsaniye2 = toplamsaniye2 + saniye2;

lblPlayer1Timer.Text = toplamsaniye1.ToString();

lblPlayer2Timer.Text = toplamsaniye2.ToString();

computermakemove();

timer2.Start();

timer1.Stop();

}

private void timer2\_Tick(object sender, EventArgs e)

{

saniye2 = saniye2 + 1;

toplamsaniye2 = toplamsaniye2 + saniye2;

saniye2 = 0;

lblPlayer2Timer.Text = toplamsaniye2.ToString();

if(!turn)

{

timer2.Stop();

timer1.Start();

}

}

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

{

b = (Button)sender;

if (turn)

{

b.Text = "X";

timer2.Interval = 1000;

timer2.Start();

timer1.Interval = 1000;

timer1.Start();

}

else

{

b.Text = "0";

}

turn = !turn;

b.Enabled = false;

turn\_count++;

Kazananlar();

}

private void Kazananlar()

{

bool kazanan = false;

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

kazanan = true;

if ((btn4.Text == btn5.Text) && (btn5.Text == btn6.Text) && (!btn4.Enabled))

kazanan = true;

if ((btn7.Text == btn8.Text) && (btn8.Text == btn9.Text) && (!btn7.Enabled))

kazanan = true;

if ((btn1.Text == btn4.Text) && (btn4.Text == btn7.Text) && (!btn1.Enabled))

kazanan = true;

if ((btn2.Text == btn5.Text) && (btn5.Text == btn8.Text) && (!btn5.Enabled))

kazanan = true;

if ((btn3.Text == btn6.Text) && (btn6.Text == btn9.Text) && (!btn3.Enabled))

kazanan = true;

if ((btn1.Text == btn5.Text) && (btn5.Text == btn9.Text) && (!btn1.Enabled))

kazanan = true;

if ((btn3.Text == btn5.Text) && (btn5.Text == btn7.Text) && (!btn3.Enabled))

kazanan = true;

if (kazanan)

{

string winner = "";

if (turn)

{

winner = "O";

cpuskor++;

}

else

{

winner = "X";

playerskor++;

}

timer1.Stop();

timer2.Stop();

MessageBox.Show(winner + "kazanan");

lblBerabere.Text = berabere.ToString();

lblCPUScore.Text = cpuskor.ToString();

lblPlayerSkor.Text = playerskor.ToString();

}

else

{

if (turn\_count == 9)

{

timer1.Stop();

timer2.Stop();

if (toplamsaniye1 < toplamsaniye2)

{

MessageBox.Show("X kazanan(süreyle)");

playerskor++;

}

else if (toplamsaniye1 > toplamsaniye2)

{

MessageBox.Show("O kazanan(süreyle)");

cpuskor++;

}

else

{

MessageBox.Show("berabere(süreler eşit)");

berabere++;

}

lblBerabere.Text = berabere.ToString();

lblCPUScore.Text = cpuskor.ToString();

lblPlayerSkor.Text = playerskor.ToString();

}

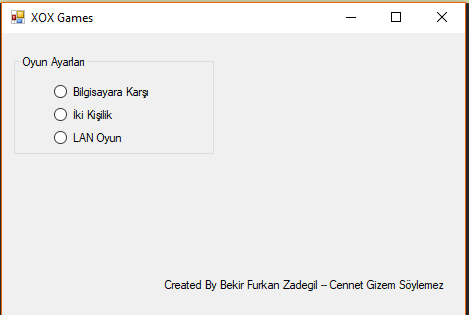
}

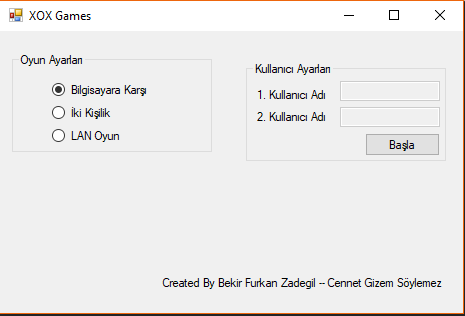
}

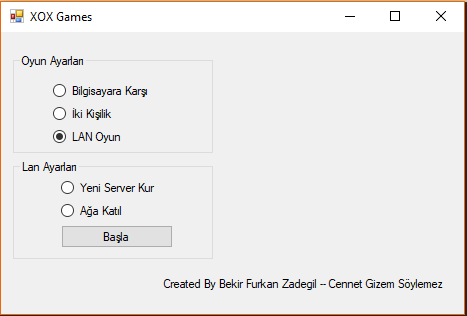
}

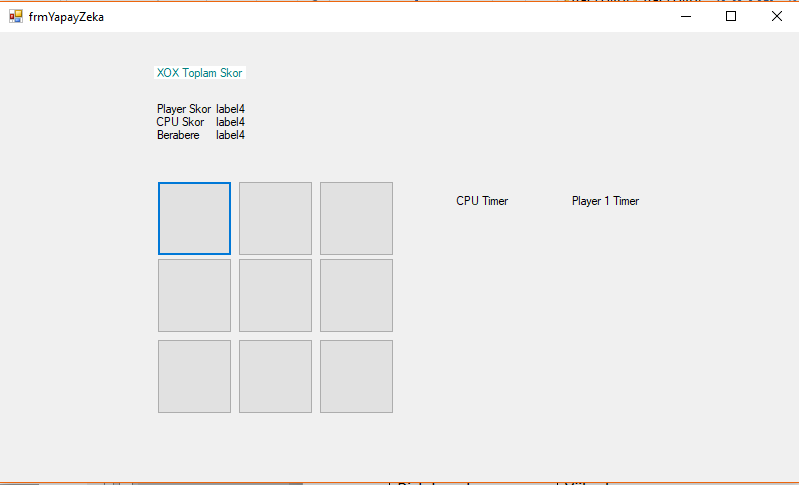
}

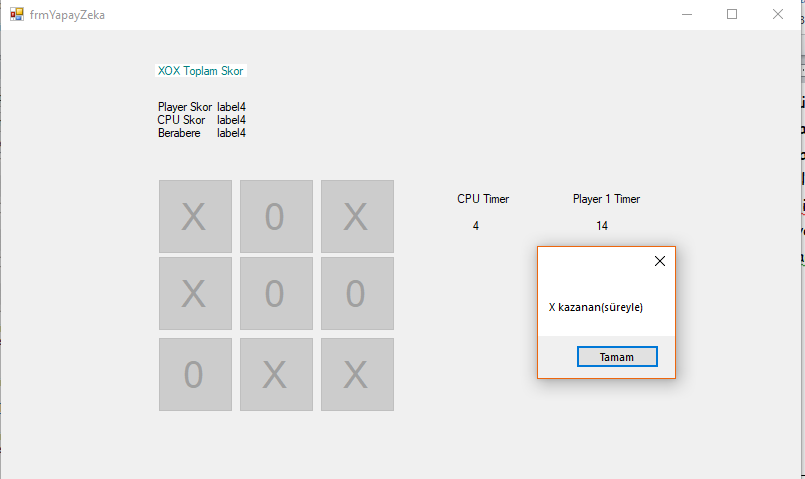
**1.4.Ekran Çıktısı**

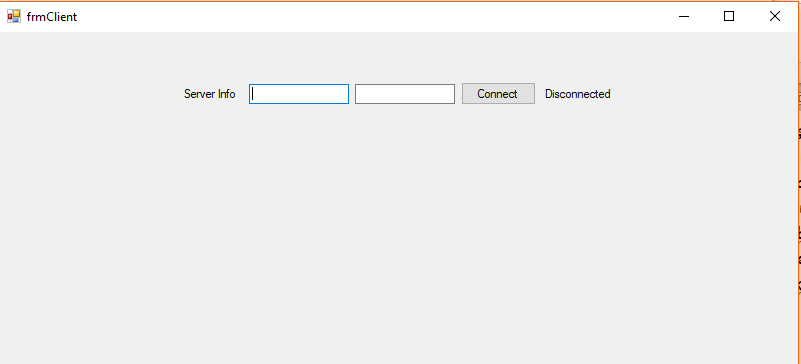
****

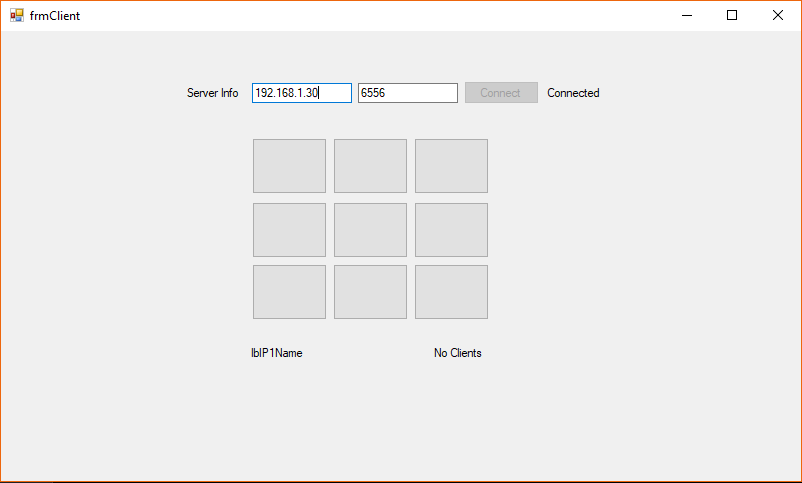
****

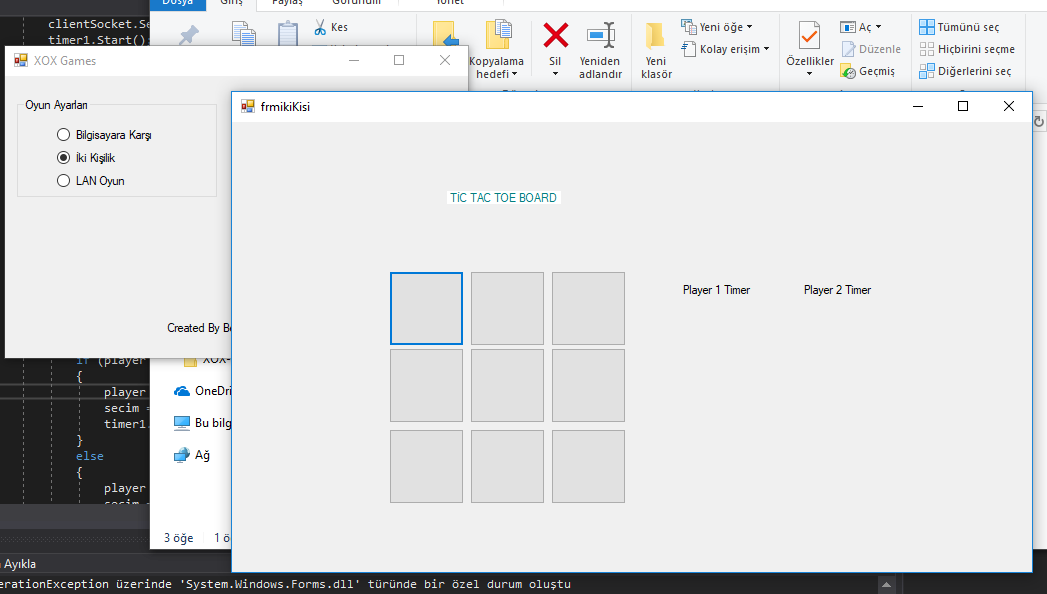
****

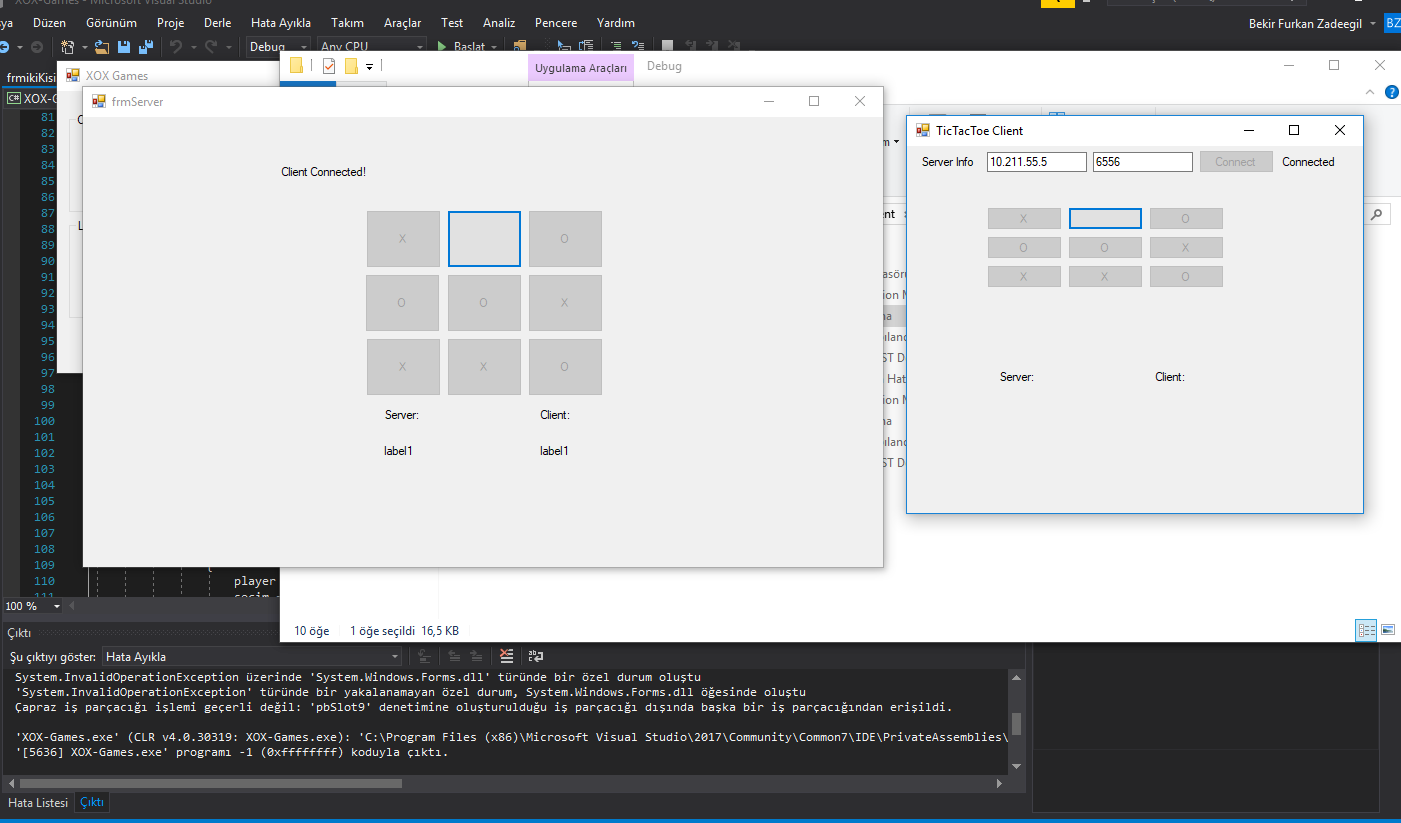
****

****

****

****

****

****

**1.5.Test Caseler**

TEST-1

Network test case

|  |  |
| --- | --- |
| Risk Level | Yüksek |
| Purpose | Bu test durumun başarılı bir şekilde oyunun test edilmesidir. |
| Inputs | Oyuncunun ilk hangi kutudan başladığı konum. |
| Expented Outputs | Oyunun sorunsuz çalışması. |
| Pass Criterias | Zaman olayının çalışmaması .(BAŞARILI)  Kullanıcı isimlerinin çalışmaması.(BAŞARILI) |
| Fail Criterias | Oyun oynanıyorken zaman uygulamasının sorunsuz çalışması.(BAŞARISIZ)  Kullanıcı isimlerinin sorunsuz çalışması.(BAŞARISIZ) |
| Text Procedure | Test kullanıcısı,yazılımı,desteklenen sistem ve cihazlarda çalıştırarak farenin peyniri belirlenen yol üzerinde başarılı bir şekilde bulma durumunun testini gerçekleştirmelidir.Test işlemi tamamlandıktan sonra test sonucunu Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır. |

TEST-2

Aynı Bilgisayardan Karşılıklı Test Case

|  |  |
| --- | --- |
| Risk Level | Yüksek |
| Purpose | Bu test durumun başarılı bir şekilde oyunun test edilmesidir. |
| Inputs | Oyuncunun ilk hangi kutudan başladığı konum. |
| Expented Outputs | Oyunun sorunsuz çalışması. |
| Pass Criterias | Oyun bittikten sonra yeni oyun başlatma.(BAŞARILI)  Kullanıcı isimlerinin çalışmaması.(BAŞARILI) |
| Fail Criterias | Oyun bittikten sonra sorunsuz bir şekilde yeni oyun butonuna tıklandığında yeni oyunun açılması.(BAŞARISIZ)  Kullanıcı isimlerinin sorunsuz çalışması.(BAŞARISIZ) |
| Text Procedure | Test kullanıcısı,yazılımı,desteklenen sistem ve cihazlarda çalıştırarak farenin peyniri belirlenen yol üzerinde başarılı bir şekilde bulma durumunun testini gerçekleştirmelidir.Test işlemi tamamlandıktan sonra test sonucunu Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır. |

TEST-3

Bilgisayara Karşı Test Case

|  |  |
| --- | --- |
| Risk Level | Yüksek |
| Purpose | Bu test durumun başarılı bir şekilde oyunun test edilmesidir. |
| Inputs | Oyuncunun ilk hangi kutudan başladığı konum. |
| Expented Outputs | Oyunun sorunsuz çalışması. |
| Pass Criterias | Oyun bittikten sonra yeni oyun başlatma.(BAŞARILI)  Kullanıcı isimlerinin çalışmaması.(BAŞARILI) |
| Fail Criterias | Oyun bittikten sonra sorunsuz bir şekilde yeni oyun butonuna tıklandığında yeni oyunun açılması.(BAŞARISIZ)  Kullanıcı isimlerinin sorunsuz çalışması.(BAŞARISIZ) |
| Text Procedure | Test kullanıcısı,yazılımı,desteklenen sistem ve cihazlarda çalıştırarak farenin peyniri belirlenen yol üzerinde başarılı bir şekilde bulma durumunun testini gerçekleştirmelidir.Test işlemi tamamlandıktan sonra test sonucunu Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır. |