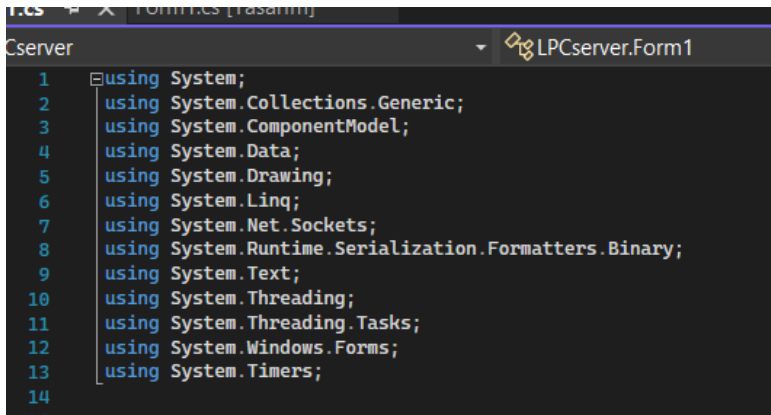


# COMPUTER NETWORK PROGRAMMING FİNAL ÖDEVİ

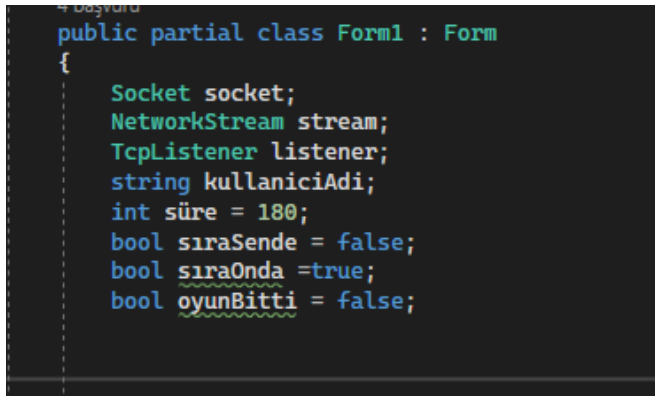
Projede Tcp bağlantıyla soket programlama yapılmaktadır. Server ve Client adlı 2 proje oluşturdum. Serverimiz soket oluşturup o soketi dinliyor. Clientimiz socketimize bağlanıyor.

## Server:



```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Net.Sockets;
8 using System.Runtime.Serialization.Formatters.Binary;
9 using System.Text;
10 using System.Threading;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13 using System.Timers;
14
```

Socket , Thread, Timers kullanıyoruz.



```
4 başlıdır
public partial class Form1 : Form
{
    Socket socket;
    NetworkStream stream;
    TcpListener listener;
    string kullanıcıAdi;
    int süre = 180;
    bool sıraSende = false;
    bool sıraOnda = true;
    bool oyunBitti = false;
}
```

Socket ,listener ve stream işlemleri serveri yaratmak için, kullanıcıAdi oyuncumuzun adını almak için, süre 3 dakika, sıranın kimde olduğunu belirtmek için sıraSende booleanı .

```

1 başvuru
public Form1()
{
    InitializeComponent();
    Control.CheckForIllegalCrossThreadCalls = false;
}

1 başvuru
private void Form1_Load(object sender, EventArgs e)
{
    timer1.Start();
}

```

Threadlerle ilgili sorunların için bu kod parçasığını buldum ve thread sorunları çözüldü..

Uygulama başlatıldığında Timer ımız başlatılıyor fakat Clientden bağlanan kullanıcı oyunu ilk başlayacağı için kodun devamında if bloğuna takılacağından şuan timer1 beklemede kalıyor.

```

private void startBtn_Click(object sender, EventArgs e)
{
    Thread serversibaslat = new Thread(fonkserveribaslat);
    serversibaslat.Start();
    listBox1.Items.Add("Oyuncu Bekleniyor...");
}

1 başvuru
void fonkserveribaslat()
{
    kullaniciAdi = kullaniciAditext.Text;
    listener = new TcpListener(5000);
    listener.Start();

    socket = listener.AcceptSocket();
    stream = new NetworkStream(socket);

    Thread thlistener = new Thread(fonklistener);
    thlistener.Start();
}

BinaryFormatter bf = new BinaryFormatter();
1 başvuru

```

Serverimiz start buttonuna tıkladığında serverbaslat tetikleniyor ve socket dinlenmeye başlanıyor.

BinaryFormatter – Data gönderimi için.

```

I Başvuru
void fonklistener()
{
    while (true)
    {
        string mesaj = bf.Deserialize(stream).ToString();

        Console.WriteLine(mesaj);

        if (mesaj.Substring(0, 3) == "100")//mesaj
        {
            listBox1.Items.Add(mesaj.Substring(3));
        }
        else if (mesaj.Substring(0, 3) == "101")//deneme
        {
            Console.WriteLine(mesaj.Substring(3));
        }
        else if (mesaj.Substring(0, 3) == "102")//sıra
        {
            rakibinSurelbl.Text=mesaj.Substring(3).ToString();
            sıraSende = true;
        }
        else if (mesaj.Substring(0, 3) == "103")
        {
            string gelen = mesaj.Substring(3).ToString();
            // int [] x = gelen.Split('=');
            //int[] y = gelen.Split('=');

            // movPicture1.Location = new Point(x[0], y[0]);
        }
    }
}

```

Listener bir while alınıp sürekli olarak veri alımını beklemektedir. Deserialize fonksiyonu ile gelen veriyi alıyoruz.

Gönderilen verilerin başlarında etiket olarak keyfi sayılar verdim.100 kodlu etiketle gelen veri Chat kısmına gelecek mesajı belirtiyor,102 kodlu etiket sırayı ve kalan süreyi belirtiyor.103 kodlu veri satrançın taşlarının konumlarının değişmesini sağlıyor. (Olası sorunlar yüzünden bu kısmı yetiştiremedim.(işletim sistemi yetersizliği,bilgisayar yetersizliği ,plan ve programlamada eksiklik.)).Gelen veriyi stringi parçalayarak gerekli yerlere gönderiyoruz 100 koduyla geleni listboxa,102 koduyla geleni sürenin yazdığı kısma ve sıranın geldiğini belirtmek için sırasende=true sekilde ayarlanılıyor.

```
1 başvuru
private void gonderBtn_click(object sender, EventArgs e)
{
    listBox1.Items.Add(kullaniciAdi + ":" + mesajtextBox1.Text);
    bf.Serialize(stream, "100" +kullaniciAdi+"-"+ mesajtextBox1.Text);
    stream.Flush();
    mesajtextBox1.Text = "";
}

1 başvuru
```

Gönder buttonuna basıldığında listboxa kendi yazdığımız mesajı ekliyoruz ve Serialize fonksiyonu ile etiketini önünde bırakarak mesajı gönderiyoruz.

```
1 başvuru
private void oynadimBtn_Click(object sender, EventArgs e)
{
    sıraSende = false;
    bf.Serialize(stream, "101" + "b4 to f8");
    bf.Serialize(stream, "102" + süre.ToString()); //süre gönder

    stream.Flush();
}
```

Oynadım buttonuna basıldığında sıra karşıya geçeceği için sıraSende booleanını false yapıyoruz ve süreyi gönderiyoruz.

```
1 başvuru
private void timer1_Tick(object sender, EventArgs e)
{
    if (sıraSende){
        süre--;
        seninSurenlbl.Text = süre.ToString();
        if (süre == 0)
        {
            timer1.Stop();
            MessageBox.Show("Oyun bitti rakip kazandı...");
            oyunBitti = true;
        }
    }
}
```

Boolean değerimizin sonucuna göre süreyi azaltıp ilgili labela yazıyoruz.

Süre bittiğinde ekrana bir uyarı çıkartıp oyunun bittiğini söylüyoruz.

Server kısmımız bitti şimdi Client Projesine geçiyoruz..

## Client:

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Timers;
using System.Windows.Forms;

namespace LCPClient
{
    4 başvuru
    public partial class Form1 : Form
    {
        ArrayList tasKonumları = new ArrayList();

        TcpClient tcpClient;
        NetworkStream stream;
        string kullanıcıAdı;
        bool sıraSende;
        int süre = 180;
        bool oyunBitti = false;
    }
}
```

Thread, Timers, Socket kullanılmakta.

Clientin bağlanabilmesi için tcpClient,stream; Clientin kullanıcı adı , sıranın kimde olduğu bilgisi ve süre değişkenlerini oluşturuyoruz.

```

BinaryFormatter bf = new BinaryFormatter();

1 başvuru
private void baslatBtn_Click(object sender, EventArgs e)
{
    string bilgisayarAdi = Dns.GetHostName();
    string ipAdresi = Dns.GetHostByName(bilgisayarAdi).AddressList[0].ToString();

    kullanıcıAdı = kullanıcıAdıtxt.Text;
    tcpClient = new TcpClient(ipAdresi, 500);
    stream = tcpClient.GetStream();

    Thread dinleyici = new Thread(fonklistener);
    dinleyici.Start();
    sıraKimdelbl.Text = "Sıra Sende:";
    sıraSende = true;
    timer1.Start();
}

```

Veri alma ve gönderme işlemleri için BinaryFormmatterımızı oluşturuyoruz.

Baslat butonunu basıldığında bilgisayarın ip adresini alıp bağlantı sağlanıyor. Serverde oluşturduğumuz 500 portuna bağlanılıyor ve timer1.start ile süre başlatılıyor.

```

1 başvuru
void fonklistener()
{
    while (true)
    {
        string mesaj = bf.Deserialize(stream).ToString();
        Console.WriteLine(mesaj.Substring(0, 3));

        if (mesaj.Substring(0, 3) == "100") //mesaj
        {
            listBox1.Items.Add(mesaj.Substring(3));
        }
        else if (mesaj.Substring(0, 3) == "101")
        {
            Console.WriteLine(mesaj.Substring(3));
        }
        else if (mesaj.Substring(0, 3) == "102") // sure
        {
            rakibinSurelbl.Text = mesaj.Substring(3);
            sıraSende = true;
        }
    }
}
}

```

Serverde kullandığım yöntemin aynısı burada da geçerli, belirli etiketlerle gelen data ilgili kısımlara ayrılıyor ve ilgili yerlere ekleniyor.

```
1 başyuru
private void gonderBtn_Click(object sender, EventArgs e)
{
    listBox1.Items.Add(kullaniciAdı + ":" + mesajtextBox1.Text);
    bf.Serialize(stream, "100" + kullaniciAdı + ":" + mesajtextBox1.Text);
    stream.Flush();
    mesajtextBox1.Text = "";
}

1 başyuru
private void oynadimBtn_Click(object sender, EventArgs e)
{
    sıraSende = false;
    bf.Serialize(stream, "101" + "b4 to f8");
    bf.Serialize(stream, "102" + süre.ToString());
    stream.Flush();

    if (tasKonumları[1].ToString() != movPicture1.Location.ToString())
    {
        bf.Serialize(stream, "103" + movPicture1.Location.ToString());
    }
}
```

Gönder butonuna tıklandığında yine serverdeki mantıkla aynı olarak etiketle veri servere gönderiliyor. Oynadım butonuna tıklandığında sıranın artık bizde olmadığı ,sürenin servere gönderilmesi ve taşların konumu belirleniyor(Olası sorunlar yüzünden bu kısmı yetiştiremedim.(İşletim sistemi yetersizliği,bilgisayar yetersizliği ,plan ve programlamada eksiklik.))



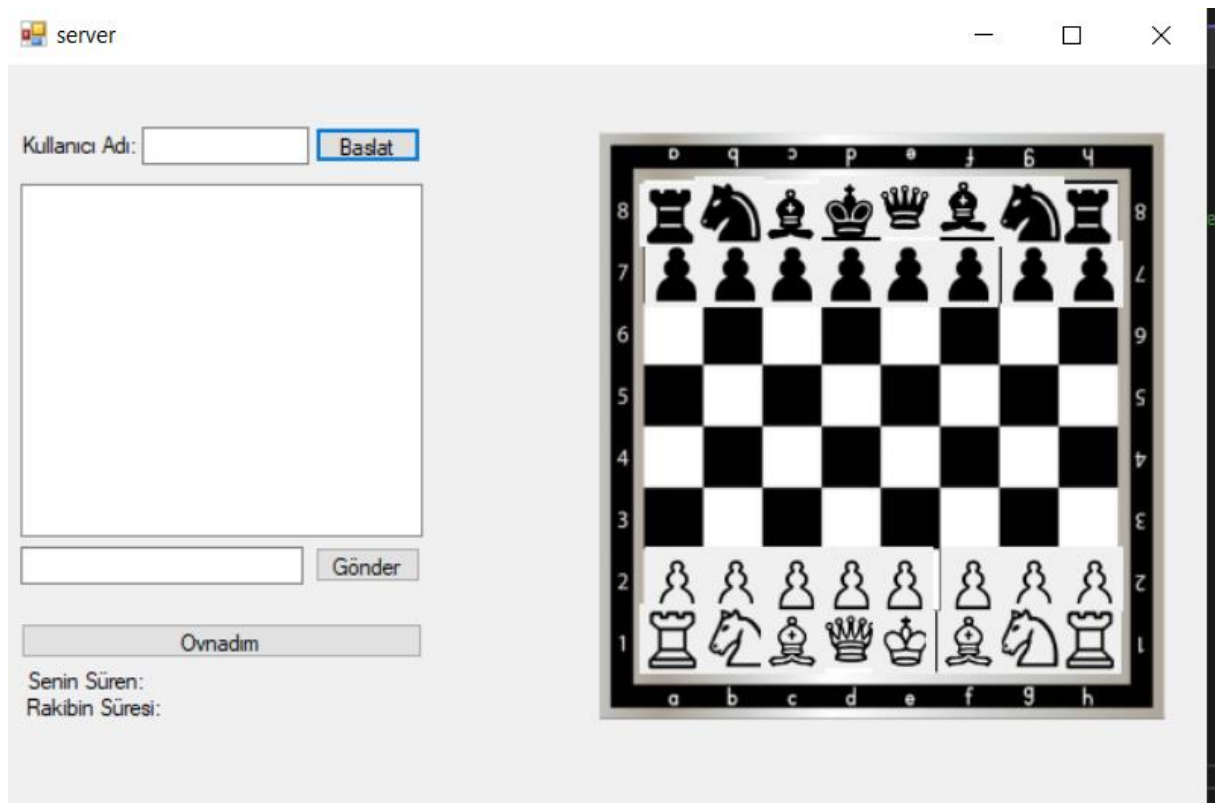
```

I Başvuru
private void timer1_Tick(object sender, EventArgs e)
{
    if (sıraSende){
        süre--;
        seninSurenlbl.Text = süre.ToString();
        if (süre == 0)
        {
            timer1.Stop();
            MessageBox.Show("Oyun bitti rakip kazandı...");
            oyunBitti = true;
        }
    }
}
}

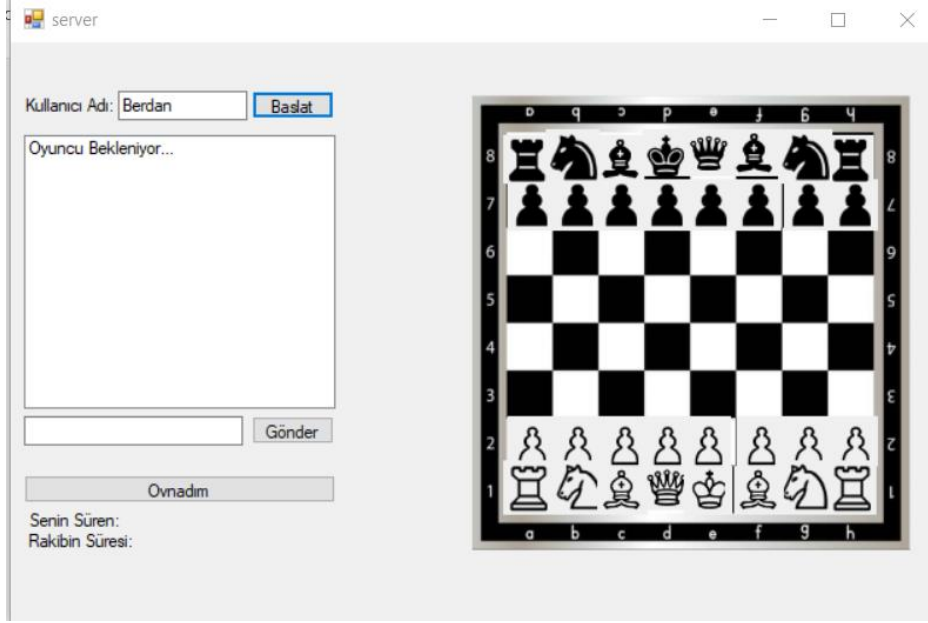
```

Timer1tick le boolean değere göre yine sıranın kimde olduğu bilgisi ve sürenin karşıya gönderilmesini sağlıyor.

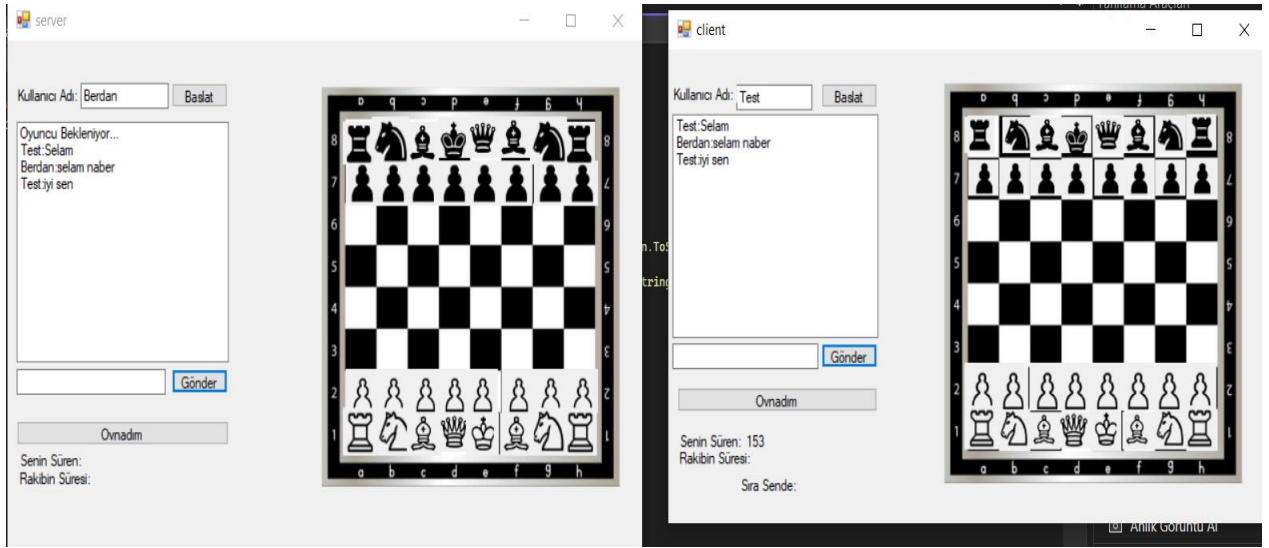
## PROGRAM:



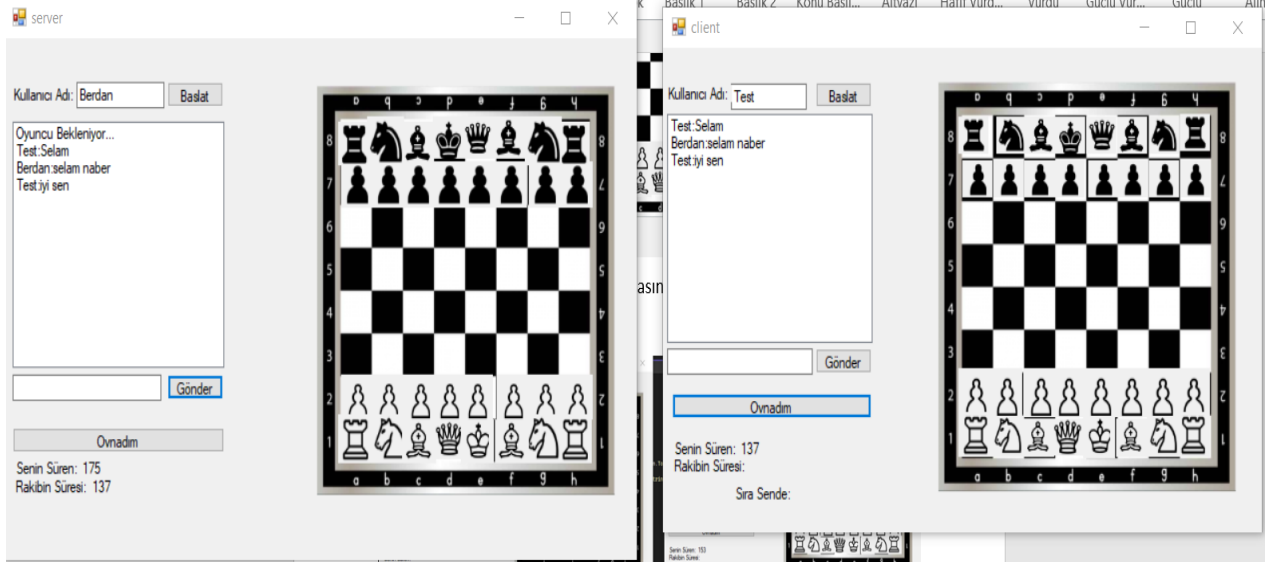
Server projesini açıyoruz ve bir kullanıcı adı girip serverimizi başatıyoruz.



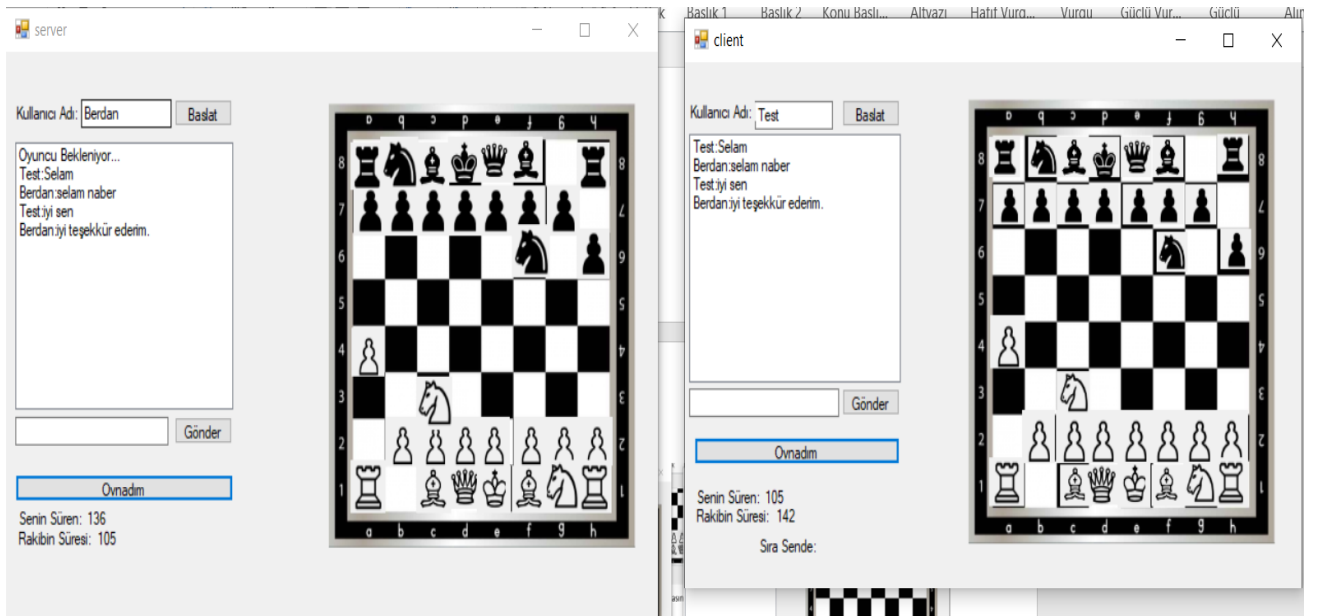
Serverimiz başlatıldı, Clientin bağlanmasını bekliyor.Portu dinliyor.



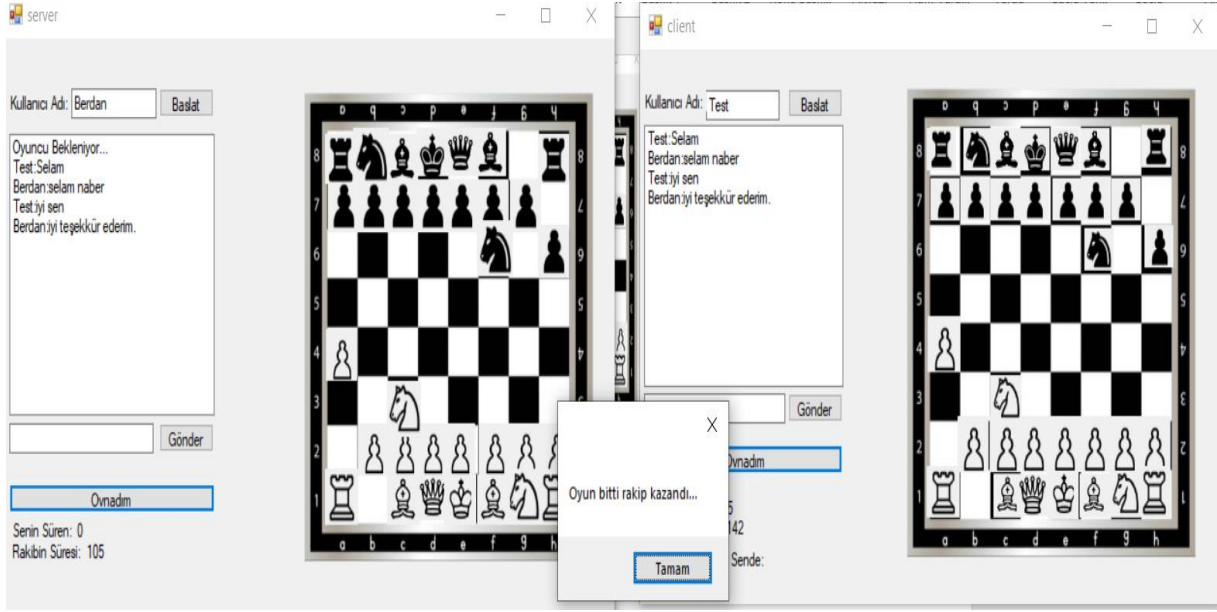
Client bağlandığında süre otomatik olarak başlıyor ve hamle yapması bekleniyor. ChatBox görüldüğü üzere iki taraflı düzgün bir şekilde çalışmaktadır.



Clientimiz oynadım butonuna basıyor ve süresi rakibe gidiyor, ve rakibin süresi başlıyor. Rakip ilgili işlemini yaptıktan sonra oynadım diyerek cliente bilgileri gönderiyor.



Görüldüğü üzere iki tarafa da bilgiler gitmektedir.



Sürenin bitmesiyle birlikte ekrana rakibin kazandığı bilgisi geliyor...

Ana projem anlattığım bu dökümandır(LCPserver ve LCPclient).Bunun dışında 2 farklı proje daha paylaşacağım, bunlar da denediğim ve başarılı olmadığım projelerdir.(TCPSocketProgChess ve TCPSocketProgClient bir proje , TcpipChess farklı bir proje(bu projede de backgroundWorkerla yapmaya çalıştım.) Bunun yanında bilgisayar eksikliğinden java ile yazmayı denedim fakat GUI kısmında chat kısmından öteye gidemedim.

**Berdan Özbeytemür**

**180316005**