



T.C.

ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

2021-2022 Eğitim-Öğretim Yılı

OBJECT ORIENTED PROGRAMMING 2-LAB

PROJE RAPORU

Projeyi Hazırlayanlar:

Mükrime SAĞIROĞLU, 152120191034, Bilgisayar Mühendisliği

Hazar NAMDAR, 152120191053, Bilgisayar Mühendisliği

HAZİRAN 2022

İçindekiler

1)Giriş Kısmı

Proje hakkında bilgi verilir.

2)Projeyi Geliştirme Aşamaları

1. Adım

2. Adım

3. Adım

4. Adım(Son Adım)

3)Görev Dağılımı

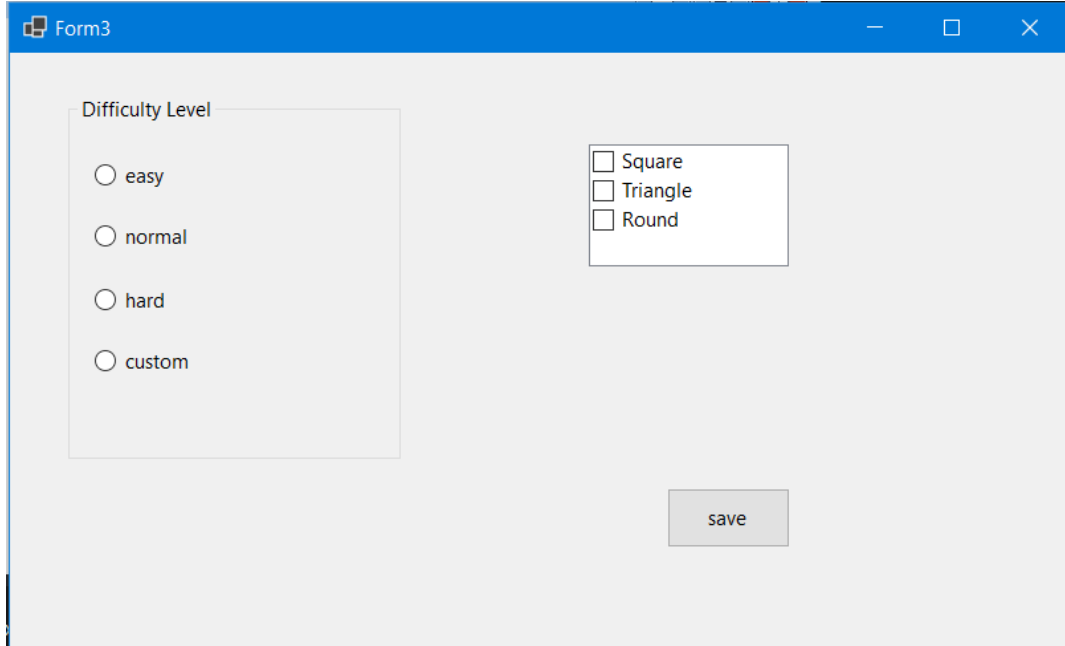
1)Giriş

Proje oyun tasarımına dayanmaktadır. Burada öncelikle kullanıcı adı ve şifreyle giriş sağlanır ardından oyun menüsünden; ayarlar, yardım ekranı, kullanıcı profili, oyun hakkında, tek oyunculu oyun ve iki kişilik oyun seçenekleri bulunmaktadır. Oyunda admin bulunmaktadır ve admin kullanıcı ekleme, silme, güncelleme ve kullanıcıların listesine bakabilme özelliklerine sahiptir. Oyunda zorluk düzeylerine göre puanlandırma yapılmaktadır. Kullanıcı bilgileri MSSQL veritabanında tutulmaktadır.

2)Proje Geliştirme Adımları

1.ADIM

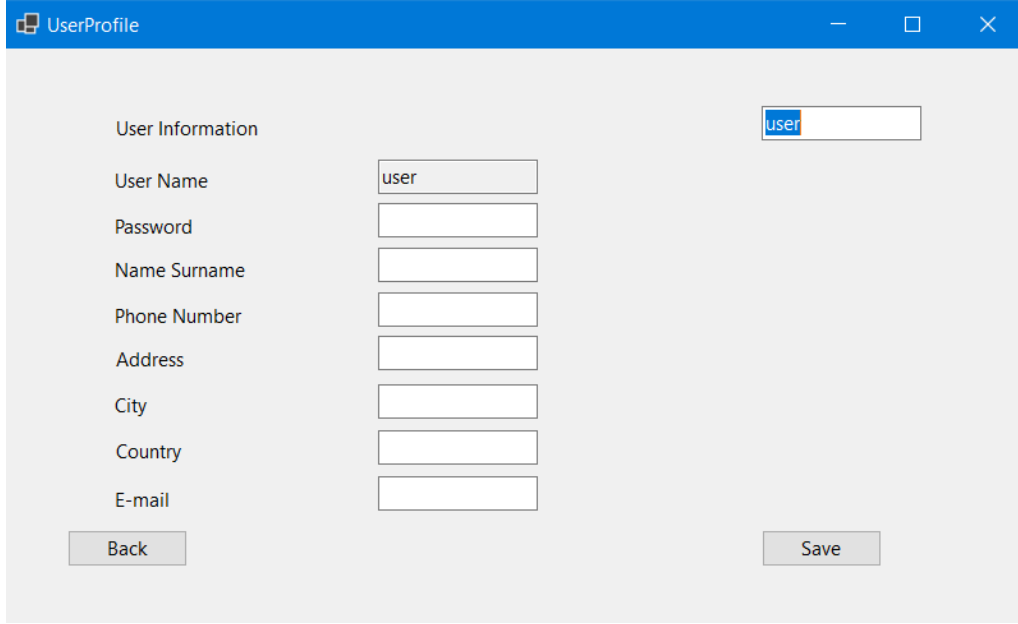
İlk ödevde projemizin temellerini attık.Bunlar log in screen oluşturmak ve ana menüde ayarlar ekranı oluşturmaktı. Ayarlar ekranına da aşağıdaki resimdeki gibi düzenlemeler yapıldı. İsterlere uygun hale getirildi.



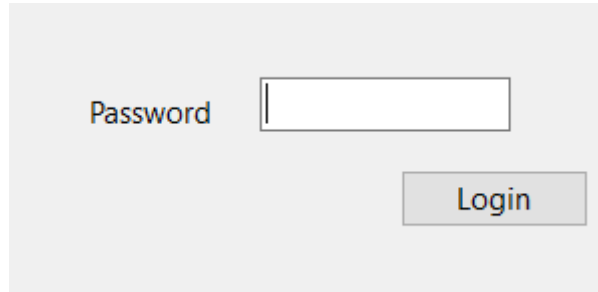
The screenshot shows a Windows application window with a blue title bar labeled 'Form3'. The main content area is light gray. On the left, there is a group box titled 'Difficulty Level' containing four radio buttons labeled 'easy', 'normal', 'hard', and 'custom'. To the right of this group box is a smaller white box containing three checkboxes labeled 'Square', 'Triangle', and 'Round'. At the bottom right of the window, there is a gray button labeled 'save'.

2.ADIM

Bu kısımda kullanıcıların kendi profillerini görebileceği bir form ekledik. Burada kullanıcılar kendi bilgilerini güncelleyip değiştirebilecekti. Aynı zamanda kullanıcıların şifrelerini SHA256 ya göre şifreleyip kullanıcıların güvenliği artırdık.

A screenshot of a web application window titled "UserProfile". The window has a blue header bar with standard window controls (minimize, maximize, close). The main content area is light gray. On the left, there is a "User Information" section with a list of labels: "User Name", "Password", "Name Surname", "Phone Number", "Address", "City", "Country", and "E-mail". To the right of these labels are corresponding input fields. The "User Name" field contains the text "user". The "Password" field is empty. The other fields are also empty. At the bottom left of the form is a "Back" button, and at the bottom right is a "Save" button. There is also a small "user" label in the top right corner of the form area.

Yukarıdaki formda da görüldüğü gibi kullanıcılar "username" hariç tüm bilgileri değiştirebilir. Kullanıcılar bilgilerini değiştirdikten sonra güvenlik için tekrardan şifre girilmesi istenir. Şifre doğruysa bilgiler güncellenir..Yukarıdaki resimde bahsettiğim pencere görünmemektedir. Save butonuna basıldıktan sonra "enable" özelliği aktif edilerek aşağıdaki gibi görünmektedir.

A screenshot of a web application showing a "Password" label next to an empty input field. Below the input field is a "Login" button.

Bu sayede ek bir form kullanmaya gerek duymadık.

```

<users>
  <userInformation>
    <id>0</id>
    <UserType>user</UserType>
    <Username>mukrime</Username>
    <Password>1208b1032164feb70c25eaa40d3bae2b582bd8cbcd067305aee607f3551e48aa</Password>
    <NameSurname>Mükrime Sağıroğlu</NameSurname>
    <PhoneNumber>05050657612</PhoneNumber>
    <Address>Eskişehir</Address>
    <City>Eskişehir</City>
    <Country>Turkey</Country>
    <Mail>mkrimesagrglu@gmail.com</Mail>
  </userInformation>
  <userInformation>
    <id>1</id>
    <UserType>user</UserType>
    <Username>hazar</Username>
    <Password>97dd35f40ccf272e330b7243e5b7468d6dee35f3b94e0d1c4da99a9e6c211ad0</Password>
    <NameSurname>Hazar Namdar</NameSurname>
    <PhoneNumber>05354353560</PhoneNumber>
    <Address>Eskişehir</Address>
    <City>Eskişehir</City>
    <Country>Turkey</Country>
    <Mail>hazarnamdarrr@gmail.com</Mail>
  </userInformation>
  <userInformation>
    <id>2</id>
    <UserType>user</UserType>
    <Username>aaa</Username>
    <Password>9834876dcfb05cb167a5c24953eba58c4ac89b1adf57f28f2f9d09af107ee8f0</Password>
    <NameSurname>aaa</NameSurname>
    <PhoneNumber>555555555</PhoneNumber>
    <Address>aaa</Address>
    <City>aaa</City>
    <Country>aaa</Country>
    <Mail>aaa</Mail>
  </userInformation>

```

Kullanıcıların bilgilerini tutmak için yukarıda bilgiler bulunan xml kullanıldı. Xml dosyasının kodları da aşağıdaki resimde verilmiştir. Burada da şifreleme için SHA256 kullandığımız şifre kısmında daha belirgin görünmektedir.

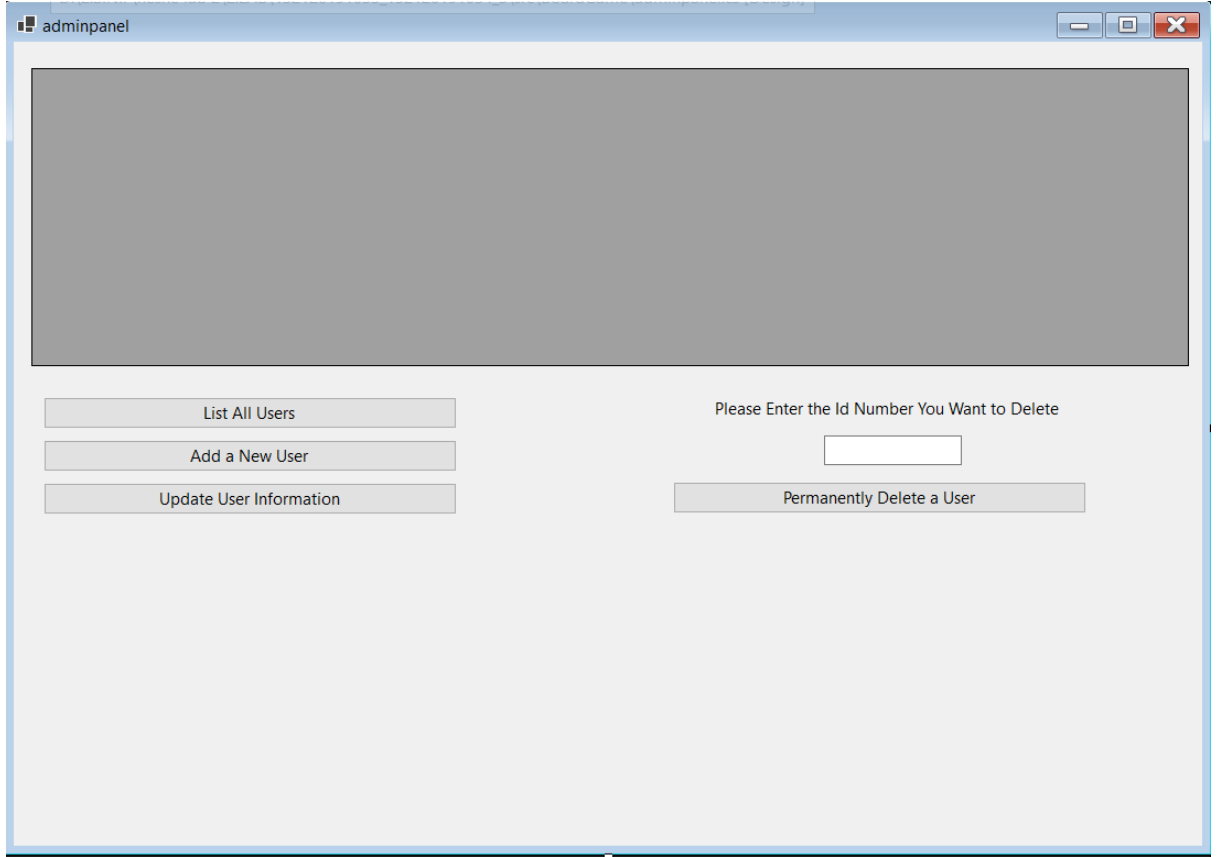
```

private void btnLogin_Click(object sender, EventArgs e)
{
  XDocument x = XDocument.Load(@"../../userinfo.xml");
  XElement update = x.Element("users").Elements("userInformation").FirstOrDefault(a => a.Element("Username").Value.Trim() == txtUsername.Text);
  if ((txtPassword.Text != "") && (txtNameSurname.Text != "") //tüm kullanıcı bilgilerini doldurması için
    && (txtPhoneNumber.Text != "") && (txtAddress.Text != "") && (txtCity.Text != "")
    && (txtCountry.Text != "") && (txtEmail.Text != ""))
  {
    if (update != null)
    {
      update.SetElementValue("UserType", txtUserType.Text);
      hPassword = sha256_hash(txtPassword.Text);
      update.SetElementValue("Password", hPassword);
      update.SetElementValue("NameSurname", txtNameSurname.Text);
      update.SetElementValue("PhoneNumber", txtPhoneNumber.Text);
      update.SetElementValue("Address", txtAddress.Text);
      update.SetElementValue("City", txtCity.Text);
      update.SetElementValue("Country", txtCountry.Text);
      update.SetElementValue("Mail", txtEmail.Text);
      x.Save(@"../../userinfo.xml");
    }
    label11.Text = "Saved";
  }
  else
  {
    label11.Text = "Error";
  }
}

//If ((txtPassword.Text == "user" || txtPassword.Text == "USER"))
//{
//  MessageBox.Show("Your Information Has Been Updated");
//}
//else
//  MessageBox.Show("Error! Please Check Your Password");
}

```

2. adımda yaptığımız bir diğer ayarlama ise admin paneliydi. Bu adımda kullanıcıları listeleyebilen, bilgilerini güncelleyebilen ve silebilen bir admin paneli oluşturduk.



Admin panelinin düzeni ve içeriği yukarıdaki gibidir. Burada da yeni kullanıcı ekleme ve güncelleme için yeni formlar oluşturduk. Kullanıcıları listeleme ve silme işlemini aynı formda tamamladık.

```
void listele()
{
    XmlDocument x = new XmlDocument();
    DataSet ds = new DataSet();
    XmlReader xmlfile;
    xmlfile = XmlReader.Create(@"../../../userinfo.xml", new XmlReaderSettings());
    ds.ReadXml(xmlfile);

    ds.Tables[0].Columns.Remove(ds.Tables[0].Columns["Password"]);

    dataGridView1.DataSource = ds.Tables[0];

    xmlfile.Close();
}
```

Adminin herhangi bir değişiklik yapmasından sonra datagridview de listelemesi için "listele()" fonksiyonunu oluşturduk. Buradaki tüm değişiklikler yine XML dosyamıza aktarılmaktadır.

Admin penceresinde görüldüğü gibi adminin kullanıcı ekleme kısmı aşağıdaki gibi görünmektedir.

The 'userAdd' window contains a form with the following fields and buttons:

- User Information:
- User ID:
- User Name:
- Password:
- Name Surname:
- Phone Number:
- Address:
- City:
- Country:
- E-mail:
- Buttons: Save, Back

Burada admin “User ID” dışındaki tüm özellikleri kendisi ekleyebilmektedir. “User ID” kısmı ise counter yardımıyla otomatik artırılmaktadır.

Admin penceresinde olan kullanıcı güncelleme kısmında ise ID numarası girildikten sonra kullanıcı bilgileri değiştirip girilen ID de güncel bilgilerin değiştirilmesi sağlanmıştır.

The 'userUpdate' window contains a search section and a form with the following fields and buttons:

- Search Section: Please Enter The ID Number Of The User You Want To Update, , Search button
- User Information:
- User ID:
- User Name:
- Password:
- Name Surname:
- Phone Number:
- Address:
- City:
- Country:
- E-mail:
- Buttons: Save, Back

Adminin kullanıcıları silme kısmında aşağıdaki resimde işaretlenmiş olan kısımda adminden silmek istediği kullanıcının ID'sini yazmasını istedik. Kullanıcı sil butonuna bastığında da kullanıcı listeden ve XML dosyamızdan kalıcı olarak silinmektedir.

	id	UserType	Username	NameSurname	PhoneNumber	Address	City	Country	Mail
	0	user	mukrime	Mükrim Sağı...	05050657612	Eskişehir	Eskişehir	Turkey	mkrimesagrgl...
	1	user	hazar	Hazar Namdar	05354353560	Eskişehir	Eskişehir	Turkey	hazarnamdarr...
	2	user	aaa	aaa	555555555	aaa	aaa	aaa	aaa
	3	user	mehmet	mehmet Kurt	555555555	sdgjsdlgjs	fbjlsjgl	fdblfdld	sbldl@gmail....
	4	user	ccc	m	m	m	m	m	m
	5	user	g	o	o	hu	hu	hu	hu
*									

List All Users

Add a New User

Update User Information

Please Enter the Id Number You Want to Delete

Permanently Delete a User

Yaptığımız bir diğer güncelleme ise kullanıcıların “settings” kısmında seçtiği bilgileri kaydedip bir sonraki girişte hatırlanmasını sağlamaktı. Bu bilgileri txt dosyasında aşağıdaki resimdeki düzende tuttuk.

userInfo - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

normal
shape
Square
Triangle
color
blue
pink

Burada kullanıcıların seçmiş olduğu zorluk düzeyi şekil ve renk mevcut.

Form3

Difficulty Level

☐ easy

☒ normal

☐ hard

☐ custom

☒ Square

☒ Triangle

☐ Round

☐ red

☒ blue

☒ pink

save

Yukarıdaki resimde de görüldüğü gibi ayarlar ekranımız güncellenmiş kullanıcıların seçimlerine göre bilgiler txt dosyasına eklenmiştir. Bu sayede kullanıcının seçimi bir sonraki girişte hatırlanmıştır.

3.ADIM

Bu adımda “newGame” adında form eklenerek oyun tasarlandı. Oyunun zorluk derecesine göre değişen boyutuna uyarlanacak “fillboard()” adında fonksiyon oluşturuldu.

```
fillBoard(dizi);

for (int i = 0; i < Row; i++)
{
    for (int j = 0; j < Colm; j++)
    {
        button[i, j].Enabled = true;

        if (button[i, j].BackgroundImage == null)
            button[i, j].Enabled = false;
    }
}
```

Yukarıda kodu verilen fillboard(), settings ekranında kullanıcıların seçtiği zorluk düzeyine göre şekillenmektedir.

```

outputFile = Application.StartupPath + @"../../../userinfo.txt"; //dosyayı okumak için
FileStream fileStream = new FileStream(outputFile, FileMode.OpenOrCreate, FileAccess.Read);
StreamReader reader = new StreamReader(fileStream);
string satir = reader.ReadLine();

if (satir == "" || satir == "normal") //Normal boyutu için uygun oyun tahtası oluşturuldu.
{
    Matris(9, 9);
    reader.Close();
}
if (satir == "easy") //Easy boyutu için uygun oyun tahtası oluşturuldu.
{
    Matris(15, 15);
    reader.Close();
}
if (satir == "hard") //Hard boyutu için uygun oyun tahtası oluşturuldu.
{
    Matris(6, 6);
    reader.Close();
}
if (satir == "custom") //Custom boyutu için uygun oyun tahtası oluşturuldu.
{
    string row = reader.ReadLine();
    string colm = reader.ReadLine();
    Matris(Convert.ToInt32(row), Convert.ToInt32(colm));
    reader.Close();
}
fileStream.Close();

```

Kullanıcı bilgilerini tuttuğumuz userinfo.txt 'den zorluk düzeyini çekip ona uygun matrisi if kullanarak oluşturduk.

```

public void buttonClick(object sender, EventArgs e)
{
    Button pushed = sender as Button;

    if (pushed.BackgroundImage != null) //Basılan buton şekilliyse
    {
        image = pushed.BackgroundImage;
        pushed.BackgroundImage = null;

        for (int i = 0; i < Row; i++)
        {
            for (int j = 0; j < Colm; j++)
            {
                button[i, j].Enabled = false;

                if (button[i, j].BackgroundImage == null)
                    button[i, j].Enabled = true;
            }
        }
    }
    else //Basılan buton boşsa
    {
        pushed.BackgroundImage = image;

        fillBoard(dizi);

        for (int i = 0; i < Row; i++)
        {
            for (int j = 0; j < Colm; j++)
            {
                button[i, j].Enabled = true;

                if (button[i, j].BackgroundImage == null)
                    button[i, j].Enabled = false;
            }
        }
    }
}

```

Bu kısımda hareket tanımlanmıştır. If kısmında basılan buton şekillidir. Else kısmında ise basılan buton boş butondur. If te resmi alma else de ise koyma işlemi yapılır.

Basılan butonun şekilli olup olmamasına göre ilk random 3 şekil atamayı sağlayan if else oluşturuldu ardından for döngüsüyle atama tamamlandı.

Bu kısmın öncesinde “images” adlı dosya oluşturulup renk ve seçeneklere göre resimler eklenmiştir.

```

public void Matris(int row, int colm)
{
    this.Row = row;
    this.Colm = colm;

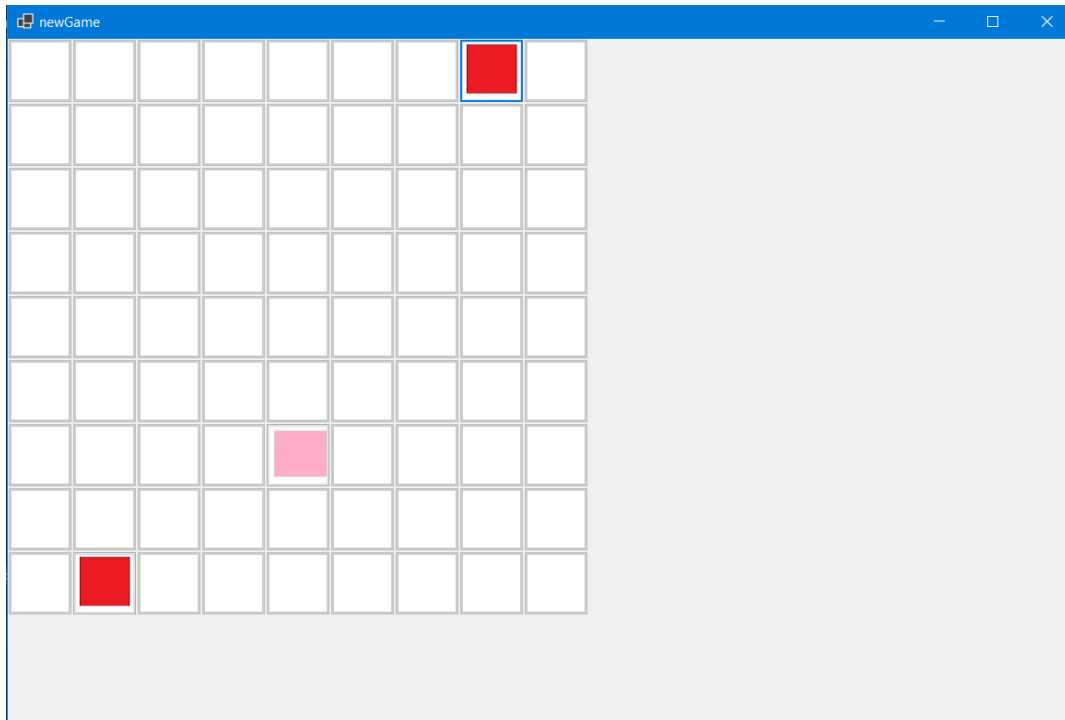
    int counter = 0;
    board = new Button[row, colm];
    int top = 0;
    int left = 0;

    int tagCount = 1; //Butonları işaretlemek için
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < colm; j++)
        {
            counter++;
            board[i, j] = new Button();
            board[i, j].Width = 35; //buton genisligi varsayılan olarak atandı.
            board[i, j].Height = 35; //buton yüksekligi varsayılan olarak atandı.
            board[i, j].Left = left;
            board[i, j].Top = top;
            board[i, j].Tag = tagCount++;
            this.Controls.Add(board[i, j]);
            left += 35;
            board[i, j].BackColor = Color.White; //butonlara varsayılan olarak beyaz renk atandı.
        }
        top += 35;
        left = 0;
    }

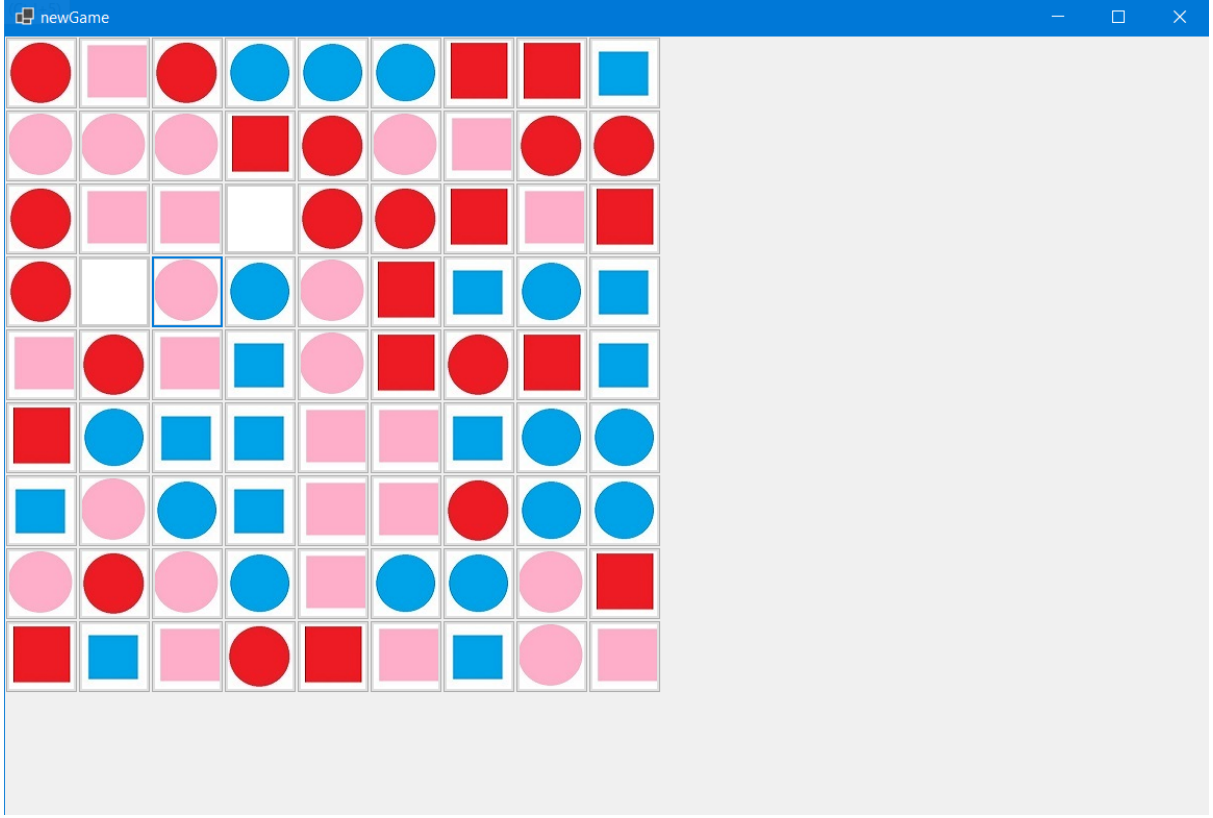
    fillBoard();
}

```

Oyunun matrisi yukarıdaki fonksiyonla oluşturulmuştur buradaki bilgiler fillboard fonksiyonu çağırarak uygulanmaktadır. fillBoard() metodu, tahtaya 3 adet rastgele renk ve şekilli resim ataması yapar.(Ayarlarda tercih edilenlerden) **fillBoard() metodu uzun bir süre boyunca yaklaşık 2000 satırlık yer kaplamıştı. Daha sonradan bu kodu spagetthi koddan daha optimize bir koda evirmeyi başardık.**

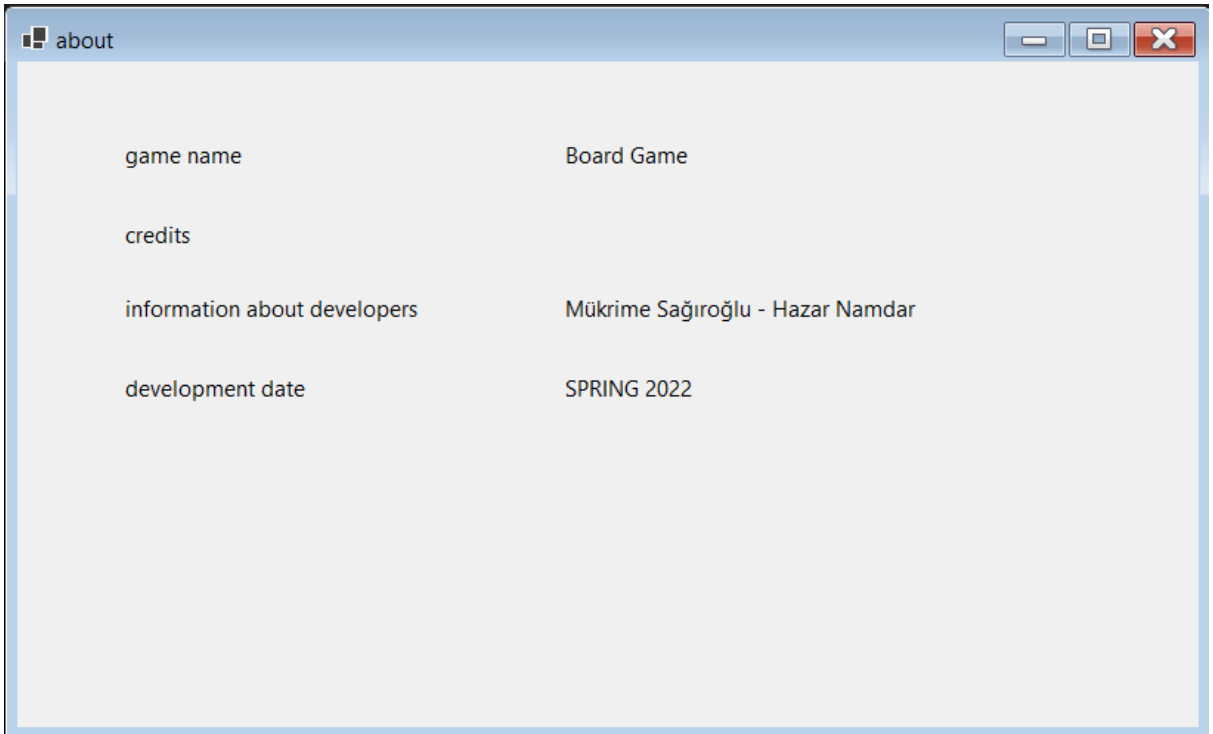


Kullanıcıların seçimlerine göre ilk 3 şekil random olarak yukarıdaki gibi atanmaktadır.



Oyun yukarıdaki resimdeki gibi tamamlanmaktadır. Burada iç içe “loop”larla atama sağlanmış boşluklar ayarlardaki seçimlere göre doldurulmaktadır.

Ayrıca oyun hakkında bilgiler olan about kısmı da laboratuvar dersinde eklenmiştir.



4.ADIM

Bu kısımda daha önce XML dosyasında tuttuğumuz verileri Access veritabanına taşıdık.

```

using System.Data.OleDb;

namespace BoardGame
{
    25 references
    public partial class adminpanel : Form
    {
        4 references
        public adminpanel()
        {
            InitializeComponent();
        }
        public static OleDbConnection baglanti; //Kullanıcı verileri access veri tabanına tasindi.
        public static OleDbCommand komut;
        public static OleDbDataAdapter da;

        2 references
        void listele()
        {
            baglanti = new OleDbConnection("Provider=Microsoft.ACE.OleDb.12.0;Data Source=../../userinfo.accdb");
            baglanti.Open();
            da = new OleDbDataAdapter("SELECT *FROM userinfo", baglanti);
            DataTable tablo = new DataTable();
            da.Fill(tablo);
            dataGridView1.DataSource = tablo;
            baglanti.Close();
            //XmlDocument x = new XmlDocument();
            //DataSet ds = new DataSet();
            //XmlReader xmlfile;
            //xmlfile = XmlReader.Create(@"../../userinfo.xml", new XmlReaderSettings());
            //ds.ReadXml(xmlfile);

            //ds.Tables[0].Columns.Remove(ds.Tables[0].Columns["Password"]);

            //dataGridView1.DataSource = ds.Tables[0];

            //xmlfile.Close();
        }
    }
    1 reference

```

Resimde görülen değişimlerle adminpanel ve userprofile kısmındaki bilgileri veritabanına aktarıldı.

id	UserType	Username	Passwords	NameSurnan	PhoneNumbr	Address	City	Country	Mail	Ekleme İçin Tıklayın
5	user	asd	asd	asd	asd	asd	asd	asd	asd	
6	user	sfd	ffds	sfd	sf	fd	f	fds	fds	
9	user	mmm	mmm	mmm	mm	m	m	mm	m	
10	user	bb	bb	bb	bb	bb	bb	bb	bb	
14	user	bb	bb	bb	bb	bb	bb	bb	bb	
16	user	bbb	bbb	bbb	bbb	bbb	bbb	bbb	bbb	
17	user	sssssss	sssssss	ssssss	ssssssss	ssssssssss	s	s	s	
18	user	eeeeee	aa	aa	a	a	a	a	a	
Yeni										

Oluşturduğumuz veritabanımız yukarıdaki gibi istelere uygun şekilde düzenlenmiştir.

```

/**
 * @brief Program yarım saniye bekletildi daha sonra fillBoard() metodu cagirildi.
 */
Thread.Sleep(500);
fillBoard();

/**
 * @brief Uygun sartlar karsilanirsa puan aldirma calisacak.
 */
for (int j = 0; j < this.Colum; j++)
    this.getPointVertical(j);
for (int i = 0; i < this.Row; i++)
    this.getPointHorizontal(i);

```

Oyuna puan aldirma eklenmiştir. bu kısım buttonClick içindeki else bölümündedir. Mantığı şöyle işler, getPointVertical(int) içine parametre olarak sütun değeri alır. Bu aldığı

sütun değerindeki her butonu kontrol eder. 5 veya 5'ten fazla aynı renk ve şekilden resim varsa, patlama gerçekleşir ve puan alınır.

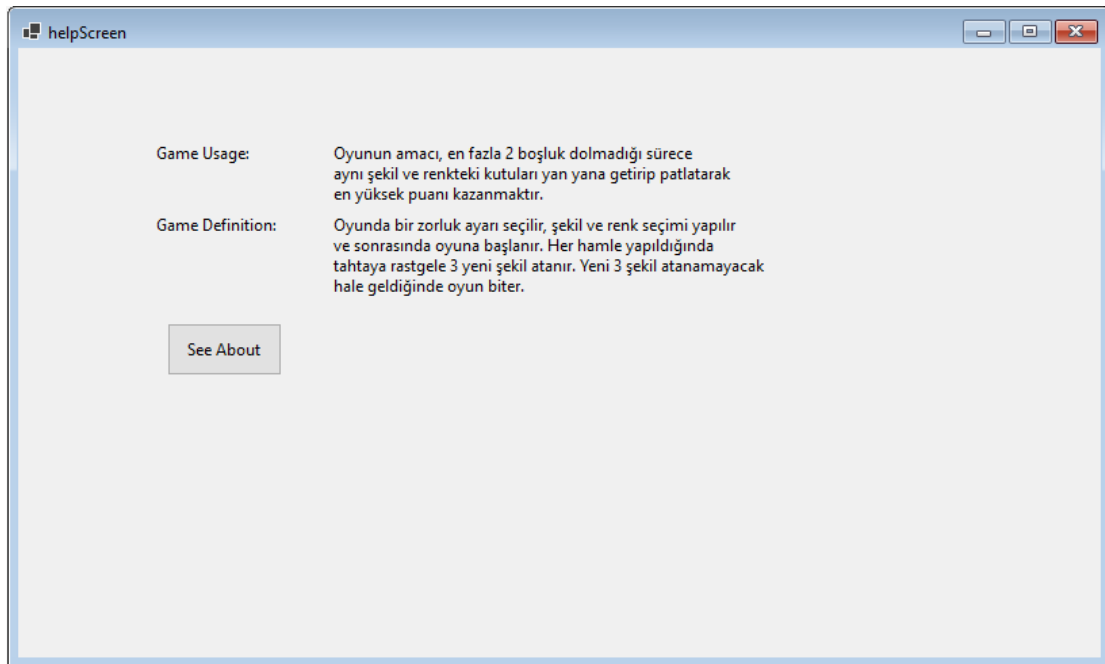
Aynı şekilde `getPointHorizontal(int)` içine satır değeri alır. Aldığı satır değerindeki her butonu kontrol eder ve 5 veya 5'ten fazla aynı renk ve şekilde resim tespit ederse onları yok eder ve puan ekler.

Her iki puan aldırmanın da algoritmasında ilk elemanla diğer elemanlar kıyaslanır ve bir `counter(pointCondition)` yardımıyla patlatma sağlanır. Eğer `counter` gerekli değere ulaşmazsa ilk eleman bir shift edilir, `counter` yeniden set edilir ve algoritma tekrar denir. Detaylı bilgi, koddan bakılarak da anlaşılabilir.

Oyuna ses özelliği eklenmiştir hamle yapıldığında ve puan kazanıldığında ses çalmaktadır.

```
}  
SoundPlayer ses = new SoundPlayer();//hamleden sonra ses eklendi.  
string konum = Application.StartupPath + @"../../sounds/hamle.wav";  
ses.SoundLocation = konum;  
ses.Play();
```

Ayrıca oyun hakkında yardımcı olacak bilgiler `helpScreen` kısmında verilmiştir. Bu kısım laboratuvar dersinde eklenmiştir.



```

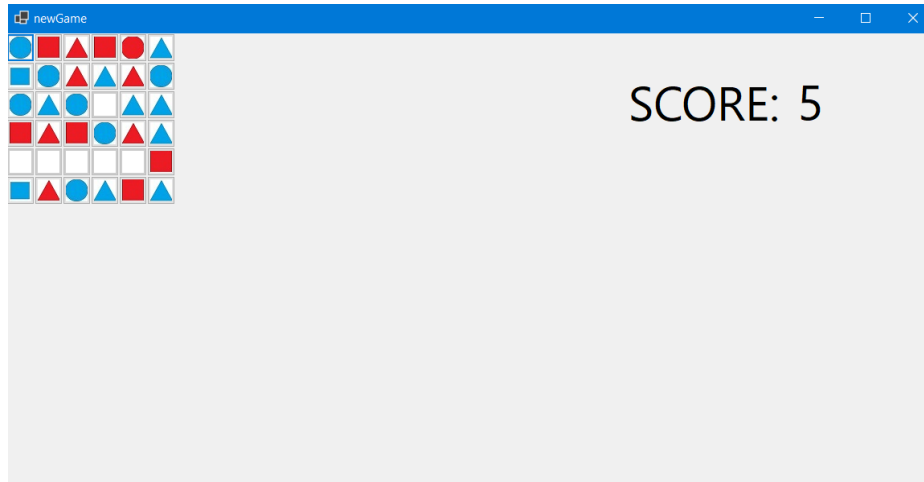
}
if (txtToArray.Contains("Square") && txtToArray.Contains("Triangle") && txtToArray.Contains("Round")) sekil = rnd.Next(1, 4);

//seçilebilecek renklere göre random renk atama kombinasyonu uyarlandı.
if (txtToArray.Contains("red") && !txtToArray.Contains("blue") && !txtToArray.Contains("pink")) renk = 1; //1=red 2=blue 3=pink
if (txtToArray.Contains("blue") && !txtToArray.Contains("red") && !txtToArray.Contains("pink")) renk = 2;
if (txtToArray.Contains("pink") && !txtToArray.Contains("red") && !txtToArray.Contains("blue")) renk = 3;
if (txtToArray.Contains("red") && txtToArray.Contains("blue") && !txtToArray.Contains("pink")) renk = rnd.Next(1, 3);
if (txtToArray.Contains("blue") && txtToArray.Contains("pink") && !txtToArray.Contains("red")) renk = rnd.Next(2, 4);
if (txtToArray.Contains("red") && txtToArray.Contains("pink") && !txtToArray.Contains("blue"))
{
    while (true) //blue bastırmamak için yapıldı.
    {
        renk = rnd.Next(1, 4);
        if (renk != 2)
        {
            break;
        }
    }
}
if (txtToArray.Contains("red") && txtToArray.Contains("blue") && txtToArray.Contains("pink")) renk = rnd.Next(1, 4);

```

Arraylist yardımıyla seçilen renk ve şekillere göre random atama sağlandı. Ardından uygun resimler eklendi.

Son Adımda(4. Adım):



Bu kısımda en az beş tane aynı renk ve şekil yan yana geldiğinde patlatma sağlandı ve zorluk düzeyine göre skor yukarıdaki resimdeki gibi sağ tarafta kullanıcıya gösterildi.

```

public void getScore(string dt)
{
    switch (dt)
    {
        case "easy":
            score += 1;

            break;
        case "normal":
            score += 3;

            break;
        case "hard":
            score += 5;

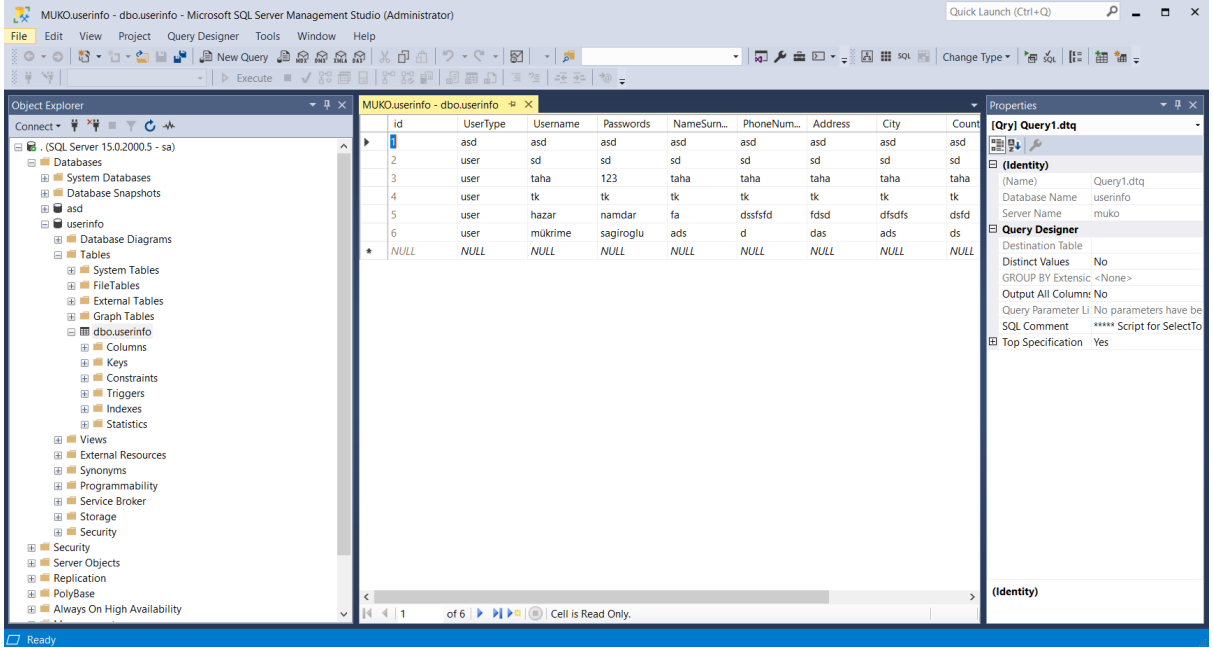
            break;
        case "custom":
            score += 2;

            break;
    }
}

```

“getScore()” fonksiyonuyla isterlere uygun puanlama sağlandı. Burada seçilen zorluğa göre puanlama yapılmıştır.

Kullanıcı bilgileri Access veritabanından microsoft SQL veritabanına aktarıldı.



Uygun kütüphane ve methodlar eklenerek access veritabanı mssql veritabanına dönüştürüldü.

```
using System.Data.SqlClient;

namespace BoardGame
{
    23 references
    public partial class adminpanel : Form
    {
        3 references
        public adminpanel()
        {
            InitializeComponent();
        }
        public static SqlConnection baglanti; //Kullanıcı verileri access veri tabanına tasindi.
        public static SqlCommand komut;
        public static SqlDataAdapter da;

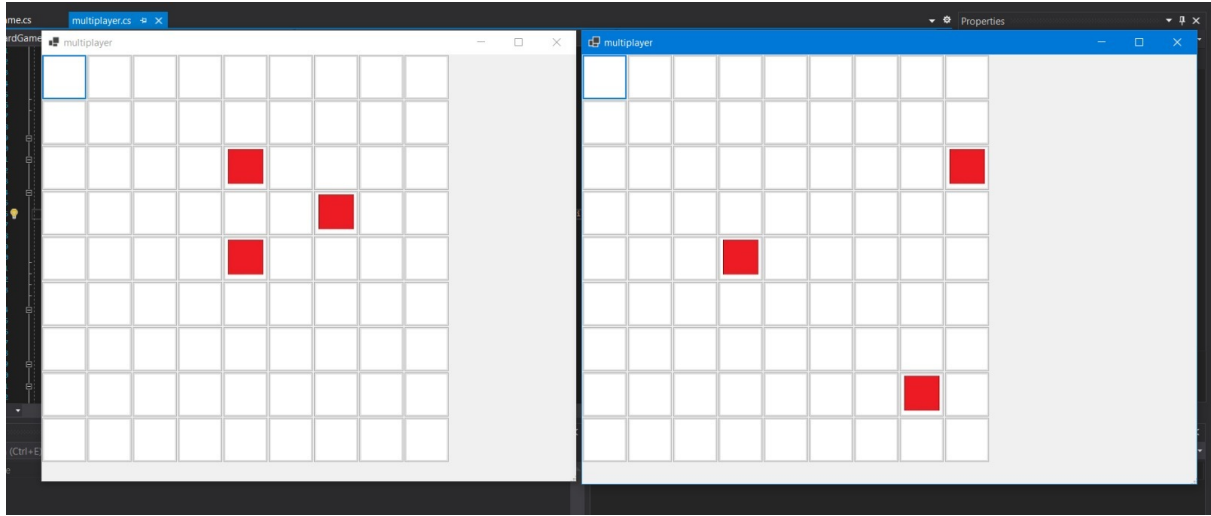
        2 references
        void listele()
        {
            baglanti = new SqlConnection("server=.; Initial Catalog=userinfo;Integrated Security=SSPI");
            baglanti.Open();
            da = new SqlDataAdapter("SELECT *FROM userinfo", baglanti);
            DataTable tablo = new DataTable();
            da.Fill(tablo);
            dataGridView1.DataSource = tablo;
            baglanti.Close();
            //XmlDocument x = new XmlDocument();
            //DataSet ds = new DataSet();
            //XmlReader xmlfile;
            //xmlfile = XmlReader.Create(@"../../userinfo.xml", new XmlReaderSettings());
            //ds.ReadXml(xmlfile);

            //ds.Tables[0].Columns.Remove(ds.Tables[0].Columns["Password"]);

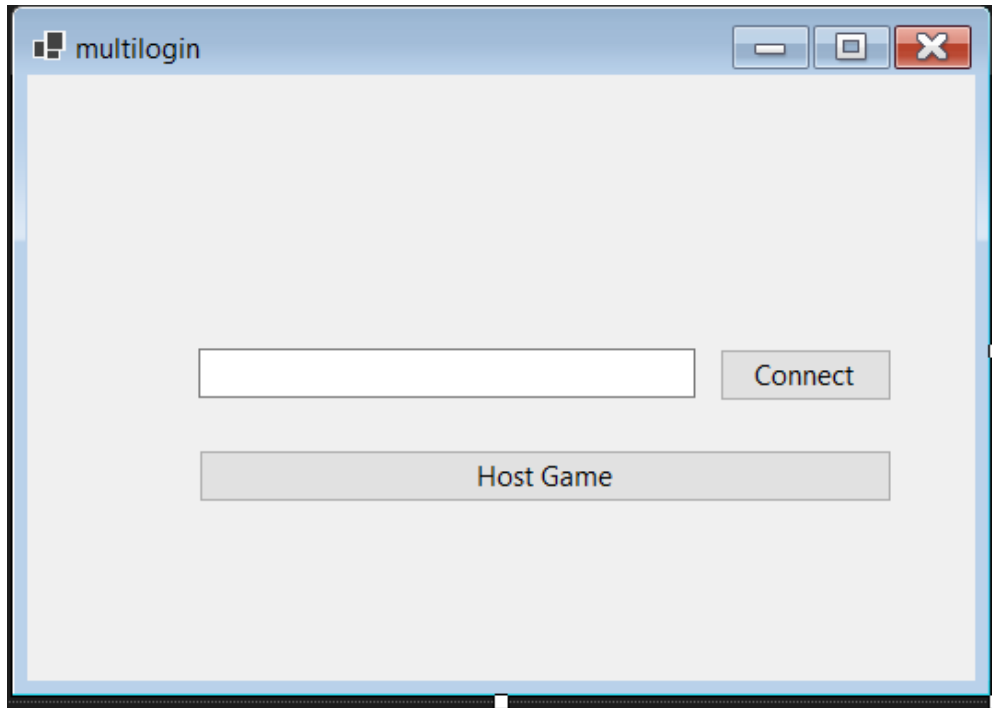
            //dataGridView1.DataSource = ds.Tables[0];

            //xmlfile.Close();
        }
    }
}
```

Son olarak multiplayer kısmında iki ekranın da aynı atama yapması sağlandı. Aşağıda programın çalışır haldeki ekran görüntüsü verilmiştir.



Multiplayer kısmında kullanıcıların IP adreslerini eşitlemek için "localhost" girişi kabul edilmiştir.



Yukarıdaki ekranda kullanıcılardan biri host diğeri de client kısmı olarak oyunu oynamaktadır.

3)Görev Dağılımı

Kodu yazarken bitbucket kullandık. Projemize public olarak aşağıdaki linkten ulaşabilir görev dağılımını inceleyebilirsiniz.

https://bitbucket.org/mukrime/oop_lab2/src/develop/