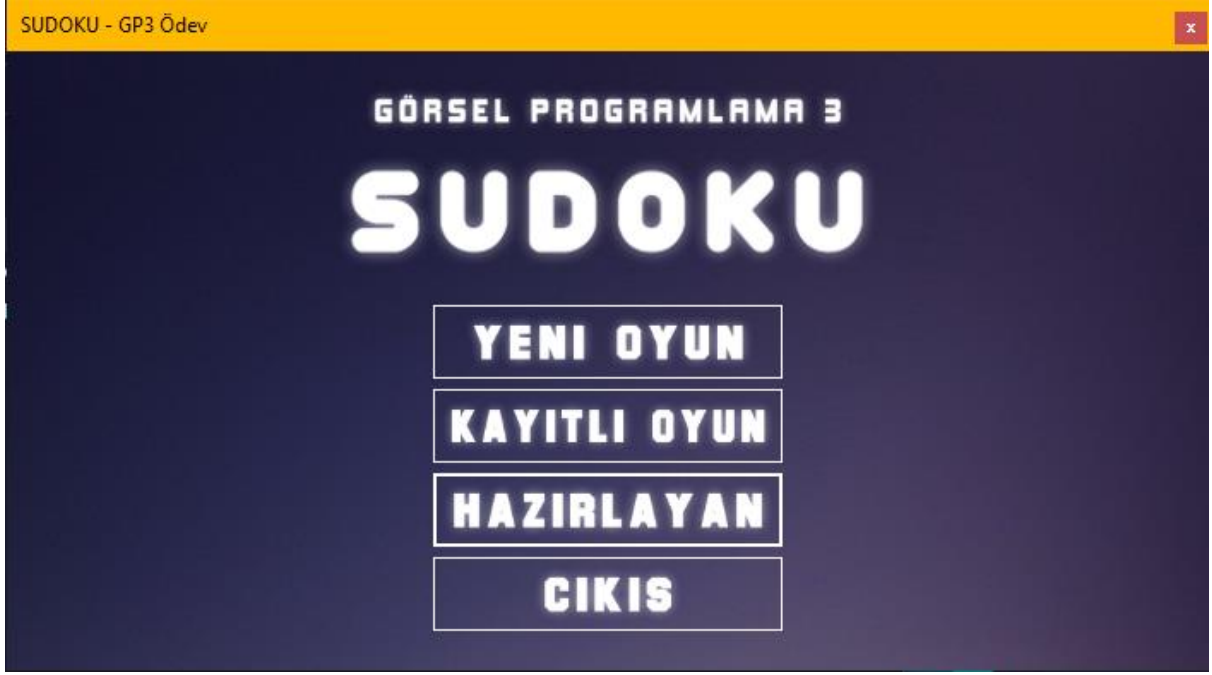


içindekiler

- Uygulama Görüntüleri..... 2
- Zorlandığım Kısımlar..... 4
- Kullanılan Metodlar..... 9
- Tasarım..... 15
- Kaynaklar..... 16

Uygulama Görüntüleri

Ana Menü :



Bu menüde 4 seçenek bulunur;

YENİ OYUN seçeneği, sudoku oyununu başlatır ve uzak xml bağlantısından çektiği verilere göre hücreleri doldurur.

KAYITLI OYUN seçeneği seçildiğinde kullanıcıya dosya seçim ekranı gösterilir. Kullanıcının seçeceği, daha önceden kaydetmiş olduğu oyun kayıt dosyasındaki (xml formatında) verilere göre hücreleri doldurur ve oyunu başlatır.

HAZIRLAYAN seçeneği, kullanıcıya uygulamayı hazırlayan öğrencinin bilgilerini MessageBox penceresinde gösterir.

ÇIKIŞ seçeneği uygulamayı kapatır.

Oyun Ekranı :



Burası kullanıcının oyunu oynayacağı ekrandır. 3x3'lük 9 adet hücre mevcuttur. Sudokunun kuralı 1'den 9'a kadar olan rakamları her satır ve sütun; ayrıca her 3x3'lük hücre grubunda her rakamı 1 defa kullanmak şartı ile yerleştirmektir. Kullanıcı tüm hücreleri doldurduğunda oyun kazanılır. Kullanıcı hücreye harf veya giremeyeceği rakamı yazdığında hata ile karşılaşır ve hücre bilgisi silinir. Kontrol bu şekilde sağlanır.

Ayrıca 3 seçenek mevcuttur;

YENİ OYUN kullanıcıya yeni oyun başlatır.

OYUNU KAYDET kullanıcının kayıt dosyasını kaydetmesi için dosya yolu seçmesini ister ve xml formatında mevcut ilerlemeyi kaydeder.

ANA MENÜ kullanıcıyı ana menüye yönlendirir.

Zorlandığım Kısımlar

Hücreleri oluşturma : Zorlandığım kısımlardan biri hücreleri oluşturmak oldu. Neye göre ve hangi sırada oluşturacağıma karar vermek epey zamanımı aldı. Oluşturacağım düzen çok önemliydi çünkü hücrelerin kontrolünü zorlaştırabilir veya kolaylaştırabilirdi. Bu yüzden hücreler için textbox kullandım ve bu textBox'ları oluştururken sudo adı ile tanımladığım çok boyutlu textbox dizisine ekledim. Index değerleri kullanarak textBox'ları döngülerde kullandım. Bu sayede hücre kontrol işlemlerini daha kısa hale getirebildim. Ayrıca oluşturduğum textBox'lara bazı özellikler ekledim. Formda düzenli durması için de kodda location komutları kullandım.

```
34 public void olustur()
35 {
36     int locationa = 30;
37     int locationb = 30;
38     int sayaca = 0;
39     int sayacb = 0;
40     Font tfont = button1.Font;
41     for (int i = 0; i < 9; i++)
42     {
43         for (int j = 0; j < 9; j++)
44         {
45             sudo[i, j] = new TextBox();
46             sudo[i, j].TextChanged += new EventHandler(giris);
47             Controls.Add(sudo[i, j]);
48             sudo[i, j].Width = 30;
49             sudo[i, j].Font = new Font(tfont.FontFamily, 15);
50             sudo[i, j].MaxLength = 1;
51             sudo[i, j].Location = new Point(locationa, locationb);
52             sudo[i, j].TextAlign = HorizontalAlignment.Center;
53             sayaca++;
54             locationa += 32;
55             if (sayaca == 3) { sayaca = 0; locationa += 5; }
56         }
57         locationa = 30;
58         locationb += 32;
59         sayacb++;
60         if (sayacb == 3) { sayacb = 0; locationb += 5; }
61     }
62     locationb = 30;
63 }
64
```

XML okuma ve hücre doldurma : Bu bölümde zorlanmamın nedeni daha önce hiç dosya işlemleri ile ilgili bir proje yapmamış olmamdı. İlk olarak xml bağlantısından veri okumak için kaynak kodlar buldum. Kod kullanımlarına baktım ve bana verilen bağlantıdaki xml formatındaki veriyi bir stringe aktardım.

```
<puzzle>
  <rows>
    <row>0,3,1,5,0,0,9,0,0</row>
    <row>0,0,2,0,0,0,7,0,0</row>
    <row>0,0,0,0,0,0,0,0,0</row>
    <row>0,0,0,0,0,0,0,0,7</row>
    <row>0,0,0,0,0,0,6,4,0</row>
    <row>0,0,0,4,0,0,8,0,0</row>
    <row>0,8,0,0,0,0,0,0,0</row>
    <row>9,0,4,0,0,0,0,0,0</row>
    <row>0,0,0,0,8,0,0,0,0</row>
  </rows>
</puzzle>
```

Elimde bir sonuç stringi vardı. Bu string 0,3,1,5,0,0,9,...,0,0,0 şeklinde düzensiz bir ifade tutuyordu. Ayrıca xml verisi çekerken xml versiyonu ve encoding bilgisi gibi gereksiz bir ifadeyi de stringe ekliyordu. Bunu önlemek için ilk olarak xml verisinin ilk satırını string içerisine eklemedim. Geriye sadece sayılar ve virgüller kaldı. Replace(); kullanarak string içerisindeki virgülleri kaldırdım. Bu sayede sadece sayıların bulunduğu bir string elde ettim. Ardından hücreleri doldururken SubString(); ile string içerisindeki numaraları sırayla hücrelere aktardım.

Kod (xml okuma ve hücreleri doldurma)

```
string dizi = ""; // sonuç stringi.

bool sil = false; // ilk satır silmek için kontrol değişkeni

String URLString = " https://www.noordpress.com/sudoku.php"; // xml link

XmlTextReader reader = new XmlTextReader(URLString); // xml okumak için reader tanımladık.

while (reader.Read()) // okuma işlemi
{
    dizi += reader.Value.Replace(",", string.Empty); // veriyi dizi adlı string içinde birleştir.

    if (sil == false) { sil = true; dizi = ""; } // ilk satırla gelen sürüm bilgilerini dahil etme.
}

int sayac = 0; // hücrelere numaraları aktarırken kullanacağımız parametre değişkeni.

for (int i = 0; i < 9; i++) // 9x9 hücre için for döngüsü
{
    for (int j = 0; j < 9; j++)
    {
        if (dizi.Substring(sayac, 1) != "0") // eğer numara 0 ise hücreyi boş bırak.
        {
            sudo[i, j].Text = dizi.Substring(sayac, 1);

            sudo[i, j].Enabled = false; // doldurulan hücreyi düzenlemeyi kapat
        }

        sayac++; // sayacı arttır.
    }
}
```

Doldurulan Hücreler

			6	9			1	
						5		
		4						8
	7			8		3		
		5			3			
				5				
								2
	1				2			

Hücrelerin kontrolü : Uygulamada en çok zorlandığım yer buydu. Çünkü sudoku kurallarına göre her satır ve sütun için ayrıca 3x3 lür her hücre grubu içinde aynı rakamların birden fazla kullanılmaması gerekiyordu. Ek olarak uygulamada tüm kontroller başarılı olunca “KAZANDINIZ” bildirimi göstermesi gerekiyordu. İlk olarak kontrol() adlı bir metod oluşturdum. Bu metod ilk olarak her satır ve sütun için tek kullanım kontrolü ve 3x3 hücre grubu için tek kullanım kontrolü yapıyordu. Bu kontrolleri yaparken; örneğin bir satırdaki rakamların tek kullanım kontrolünü yapmak için, satırdaki her rakamı, oluşturduğum int dizisindeki kendi parametresine denk gelen elemanın değerini 1 arttırıyordu. Yani eğer bir rakam, birden fazla kullanılırsa, dizideki elemanın değeri bir noktada 2 oluyor. Her eleman arttırma işleminde dizide 2 değeri aranıyor. Dizide 2 değeri bulunursa yani bir rakam birden fazla kullanıldığı anda metod false döndürüyor. Yani girilen sayı hata kontrolünden geçemiyor. Bu kontrolü her satır,sütun ve 3x3 hücre grubu için yapıyordu. Girilen sayı bu kontrollere takılmadan geçebiliyorsa metod true döndürüyor. Ardından giris() adında bir metod oluşturdum. Bu metod ise hücrelere girilmemesi gereken ifadelerin (0 sayısı, sayı olmayan ifadeler,vb.) kontrolü içeriyordu. Sayfa 4'te vermiş olduğum kodun 46 numaralı satırında olduğu üzere bu metodu oluşturduğum her textBox'un TextChanged event'i

olarak ayarladım. Bu sayede kullanıcı hücrelere her veri girdiğinde bu metod çalışacaktı. Bu metod içerisinde de kontrol() metodunu kullandım. Sonuç olarak kullanıcı birden fazla rakam kullanıldığında, 0 rakamını kullandığında veya harf girişi yaptığında uyarı mesajı alacak ve girdiği veri textBox'tan silinecekti. Kullanıcının o anki değişiklik yaptığı textBox'ın hangi textbox olduğunu döngü içinde Focused komutu kullanarak buldum. Odaklanılmış textBox'ın parametrelerini başka bir değişkene aldım ve slime hata slime vb. İşlemlerini bu parametreler yardımı ile textBox'lara ulaşarak yaptım. Son olarak kazandınız bildirimi kaldı. Tüm hücrelerin dolu olması, girilen tüm verilerin tüm hata kontrollerinden başarılı bir şekilde geçmiş olması demekti. Yani buradan sudokunun doğru değerler verilerek başarılı bir şekilde tamamlandığını çıkarabiliriz. Bunu düşünerek ayrı bir bölüme doluMu() adlı bir metod oluşturdum. Bu metod tüm hücrelerin dolu olup olmadığını veriyordu. Bu metodu giris() metodunun en alt satırında kullandım. Metodun en alt satırına, eğer tüm hücreler dolu ise "Tebrikler Oyunu Kazandınız" bildirimi gösterecek şekilde gerekli kodu yazdım. Uygulamanın hücre kontrolü bu şekilde tamamlanmış oldu. Belirttiğim metodları sonraki sayfalarda görebilirsiniz.

Kullanılan Metodlar

kontrol() metodu

```
1 reference
public bool kontrol()
{
    for (int i = 0; i < 9; i++) // her satır için tekrar kullanım kontrolü
    {
        int[] ctrl = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
        for (int j = 0; j < 9; j++)
        {
            if (sudo[i, j].Text != "")
            {
                ctrl[Convert.ToInt32(sudo[i, j].Text)]++;
                // tekrar kullanımda false döndürür
                if (ctrl.Contains(2) == true) { return (false); }
            }
        }
    }

    for (int i = 0; i < 9; i++) // her sütun için tekrar kullanım kontrolü
    {
        int[] ctrl = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
        for (int j = 0; j < 9; j++)
        {
            if (sudo[j, i].Text != "")
            {
                ctrl[Convert.ToInt32(sudo[j, i].Text)]++;
                // tekrar kullanımda false döndürür
                if (ctrl.Contains(2) == true) { return (false); }
            }
        }
    }

    // her 3x3 hücre grubu için döngü
    for (int y = 0; y < 7; y += 3)
    {
        for (int x = 0; x < 7; x += 3)
        {
            int[] ctrl = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    if (sudo[(i + y), (j + x)].Text != "")
                    {
                        ctrl[Convert.ToInt32(sudo[(i + y), (j + x)].Text)]++;
                        // tekrar kullanımda false döndürür
                        if (ctrl.Contains(2) == true) { return (false); }
                    }
                }
            }
        }
    }

    // hatalara takılmaz ise true döndür
    return (true);
}
```

giris() metodu (textbox TextChanged event)

```
278 protected void giris(object sender, EventArgs e)
279 {
280     int ti=0, tj=0; // mevcut textBox parametrelerini tutmak için değişken oluşturduk.
281     for (int i = 0; i < 9; i++) // 9x9 hücre döngüsü
282     {
283         for (int j = 0; j < 9; j++)
284         {
285             if (sudo[i, j].Focused) { ti = i; tj = j; break; } // odaklanmış ise parametreyi değişkenlere al
286         }
287     }
288
289     // kullanıcının sayı dışında veri girmesine karşı..
290     long val; // long değişkeni oluşturduk
291     try{
292         // girilen değer boş değilse Int64 çevirme işlemi yap
293         if (sudo[ti, tj].Text != "") { val = System.Convert.ToInt64(sudo[ti, tj].Text); }
294     }
295     catch{ // hata veriyorsa sayı değildir
296         MessageBox.Show("Sadece rakam girebilirsiniz."); // gerekli uyarıyı ver
297         sudo[ti, tj].Clear(); // mevcut textBox verisini temizle
298     }
299
300     if (!kontrol()) { // kontrol metodu false değer döndürürse yani tekrar kullanım yapılmış ise
301         MessageBox.Show("Girdiğiniz rakam bu hücre içerisinde tekrar kullanılamaz."); // uyarı ver
302         sudo[ti, tj].Clear(); // mevcut textBox verisini temizle
303     }
304
305     // kullanıcı 0 girerse
306     if (sudo[ti, tj].Text == "0") { MessageBox.Show("Yalnız 1'den 9'a kadar olan rakamları kullanabilirsiniz."); sudo[ti, tj].Clear(); }
307
308     // tüm hücreler hatalara takılmadan doldurulmuş ise oyunBitti metodunu çalıştır
309     if (doluMu()) { MessageBox.Show("Tebrikler. Oyunu Kazandınız !"); oyunBitti(); }
310 }
311
```

doluMu() metodu

```
312 public bool doluMu()
313 {
314     for (int i = 0; i < 9; i++)
315     {
316         for (int j = 0; j < 9; j++)
317         {
318             // textBox dolu mu kontrolü.
319             if (sudo[i, j].Text == "") { return (false); }
320         }
321     }
322
323     // tüm hücreler dolu ise true döndür
324     return (true);
325 }
```

olustur() metodu

```
1 reference
34 public void olustur()
35 {
36     int locationa = 30;
37     int locationb = 30;
38     int sayaca = 0;
39     int sayacb = 0;
40     Font tfont = button1.Font;
41     for (int i = 0; i < 9; i++)
42     {
43         for (int j = 0; j < 9; j++)
44         {
45             sudo[i, j] = new TextBox();
46             sudo[i, j].TextChanged += new EventHandler(giris);
47             Controls.Add(sudo[i, j]);
48             sudo[i, j].Width = 30;
49             sudo[i, j].Font = new Font(tfont.FontFamily, 15);
50             sudo[i, j].MaxLength = 1;
51             sudo[i, j].Location = new Point(locationa, locationb);
52             sudo[i, j].TextAlign = HorizontalAlignment.Center;
53             sayaca++;
54             locationa += 32;
55             if (sayaca == 3) { sayaca = 0; locationa += 5; }
56         }
57         locationa = 30;
58         locationb += 32;
59         sayacb++;
60         if (sayacb == 3) { sayacb = 0; locationb += 5; }
61     }
62     locationb = 30;
63 }
```

sifirla() metodu

```
3 references
341 public void sifirla() // tüm hücreleri temizler
342 {
343     for (int i = 0; i < 9; i++)
344     {
345         for (int j = 0; j < 9; j++)
346         {
347             sudo[i, j].Clear();
348         }
349     }
350 }
```

yeniOyun() metodu

```
65 public void yeniOyun()  
66 {  
67     sifirla();  
68     for (int i = 0; i < 9; i++)  
69     {  
70         for (int j = 0; j < 9; j++)  
71         {  
72             sudo[i, j].Text = "";  
73             sudo[i, j].Enabled = true;  
74             sudo[i, j].BackColor = default;  
75         }  
76     }  
77  
78     string dizi = ""; // sonuç stringi.  
79     bool sil = false; // ilk satır silmek için kontrol değişkeni  
80     String URLString = " https://www.noordpress.com/sudoku.php"; // xml link  
81     XmlTextReader reader = new XmlTextReader(URLString); // xml okumak için reader tanımladık.  
82     while (reader.Read()) // okuma işlemi  
83     {  
84         dizi += reader.Value.Replace(",", string.Empty); // veriyi dizi adlı string içinde birleştir.  
85         if (sil == false) { sil = true; dizi = ""; } // ilk satırla gelen sürüm bilgilerini dahil etme.  
86     }  
87     int sayac = 0; // hücrelere numaraları aktarırken kullanacağımız parametre değişkeni.  
88     for (int i = 0; i < 9; i++) // 9x9 hücre için for döngüsü  
89     {  
90         for (int j = 0; j < 9; j++)  
91         {  
92             if (dizi.Substring(sayac, 1) != "0") // eğer numara 0 ise hücreyi boş bırak.  
93             {  
94                 sudo[i, j].Text = dizi.Substring(sayac, 1);  
95                 sudo[i, j].Enabled = false; // doldurulan hücreyi düzenlemeyi kapat  
96             }  
97             sayac++; // sayacı arttır.  
98         }  
99     }
```

oyunBitti() metodu

```
316 public void oyunBitti()  
317 {  
318     for (int i = 0; i < 9; i++)  
319     {  
320         for (int j = 0; j < 9; j++)  
321         {  
322             // oyun kazanıldıysa tüm hücreler için düzenlemeyi kapat  
323             // tüm hücreleri yeşile boy :)  
324             sudo[i, j].Enabled = false;  
325             sudo[i, j].BackColor = Color.Green;  
326         }  
327     }  
328 }
```

kaydet() metodu

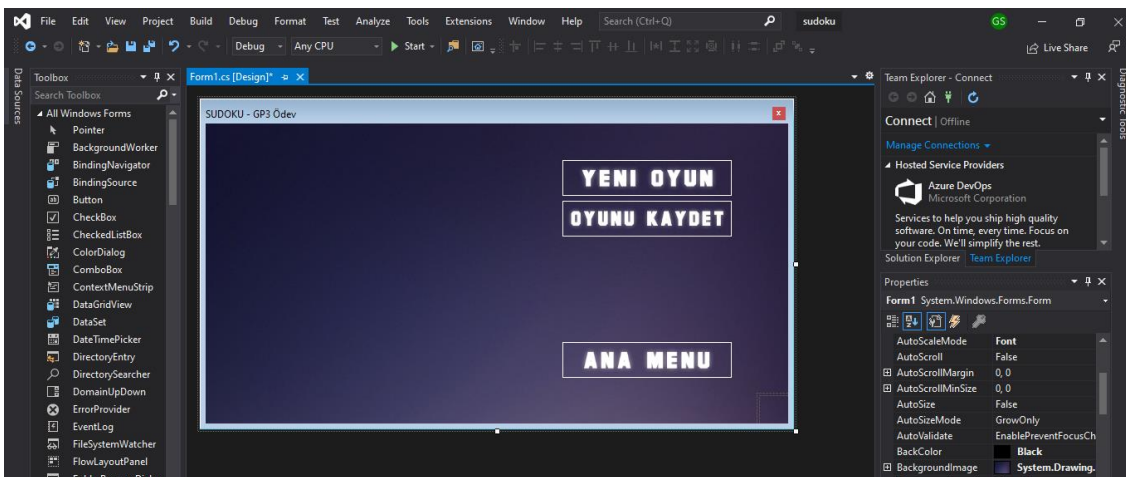
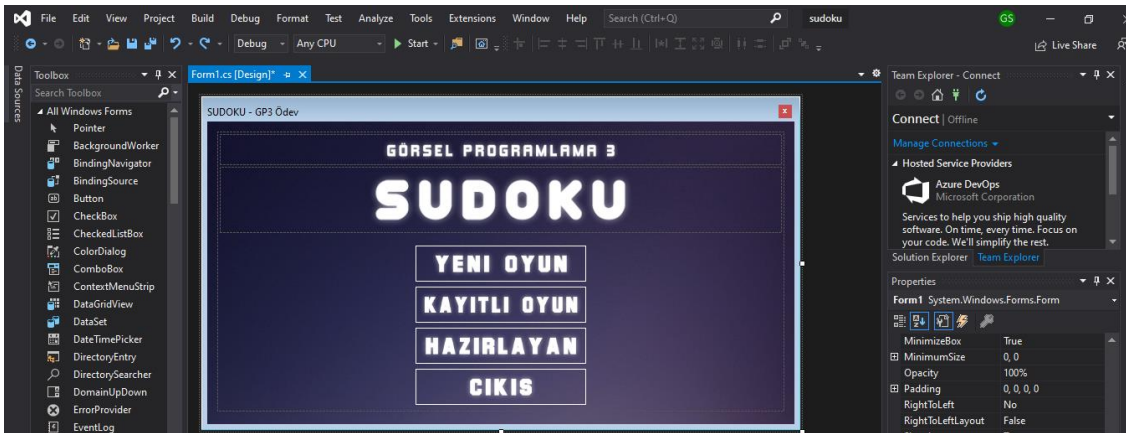
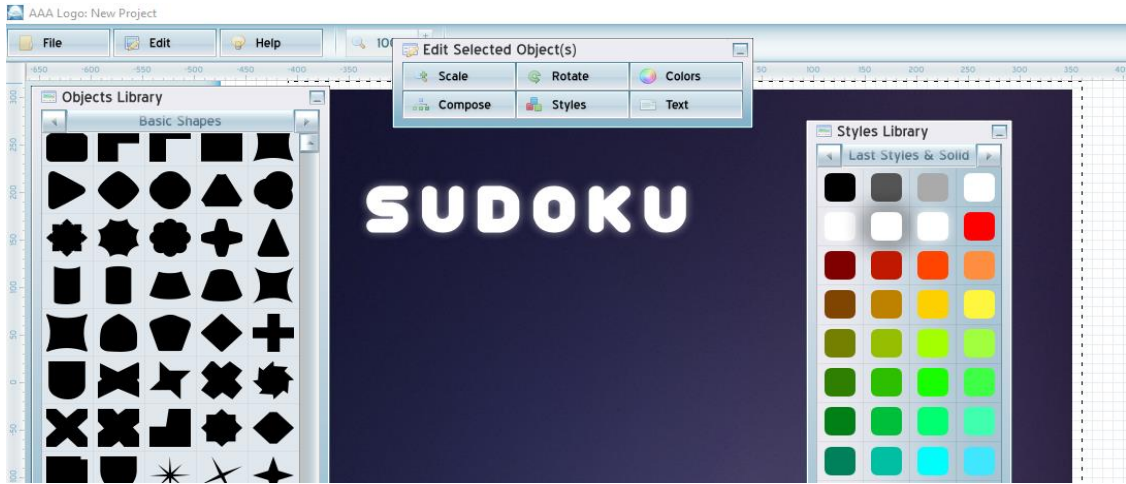
```
1 reference
161 public void kaydet()
162 {
163     SaveFileDialog save = new SaveFileDialog();
164     save.Filter = "XML Dosyası|*.xml";
165     save.OverwritePrompt = true;
166     save.CreatePrompt = true;
167     save.Title = "İlerlemeni Kaydet";
168
169     if (save.ShowDialog() == DialogResult.OK)
170     {
171         string text = "";
172         XmlWriter xmlYazici = XmlWriter.Create(save.FileName);
173
174         xmlYazici.WriteStartDocument();
175         xmlYazici.WriteStartElement("puzzle");
176         xmlYazici.WriteStartElement("rows");
177         for (int i = 0; i < 9; i++)
178         {
179             xmlYazici.WriteStartElement("row");
180             for (int j = 0; j < 9; j++)
181             {
182                 if (sudo[i, j].Text == "") {
183                     text += "0";
184                 }
185                 else
186                 {
187                     text += sudo[i, j].Text;
188                     if (j != 8) { text += ","; }
189                 }
190             }
191             xmlYazici.WriteString(text);
192             xmlYazici.WriteEndElement();
193             text = "";
194         }
195         xmlYazici.WriteEndElement();
196         xmlYazici.WriteEndElement();
197         xmlYazici.WriteEndDocument();
198         xmlYazici.Close();
199     }
200 }
```

kayitac() metodu

```
202 public void kayitac()  
203 {  
204     OpenFileDialog file = new OpenFileDialog();  
205     file.RestoreDirectory = true;  
206     file.Filter = "XML Dosyası|*.xml";  
207     file.Title = "Kayıtlı Oyunun XML Dosyasını Seçin.";  
208     file.ShowDialog();  
209  
210     string DosyaYolu = file.FileName;  
211  
212     sifirla();  
213     for (int i = 0; i < 9; i++)  
214     {  
215         for (int j = 0; j < 9; j++)  
216         {  
217             sudo[i, j].Text = "";  
218             sudo[i, j].Enabled = true;  
219         }  
220     }  
221  
222     string dizi = "";  
223     bool sil = false;  
224     String URLString = file.FileName;  
225     XmlTextReader reader = new XmlTextReader(URLString);  
226     while (reader.Read())  
227     {  
228         dizi += reader.Value.Replace(",", string.Empty);  
229         if (sil == false) { sil = true; dizi = ""; } // ilk satırla gelen sürüm bilgilerini sildik.  
230     }  
231     int sayac = 0;  
232     for (int i = 0; i < 9; i++)  
233     {  
234         for (int j = 0; j < 9; j++)  
235         {  
236             if (dizi.Substring(sayac, 1) != "0")  
237             {  
238                 sudo[i, j].Text = dizi.Substring(sayac, 1);  
239                 sudo[i, j].Enabled = false;  
240             }  
241             sayac++;  
242         }  
243     }  
244 }
```

Tasarım

Uygulama tasarımında pictureBox araçları kullandım ve butonların arkaplan resimleri için 'AAA Logo' adlı basit bir tasarım programından yardım aldım.



Kaynaklar

Uygulamayı Hazırlarken Bilgi Topladığım Web Siteleri

Sanalkurs.net

Hikmetokumus.com

Yazilimkodlama.com

Stack Overflow

W3Schools

Microsoft

Ek Olarak Uygulamayı Test Etmek İçin

Matematiktutkusu

Sudoku çözücü