



**HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ**

**ÖDEV – 3**

**Satranç At Oyunu**

**YAZILIM SINAMA**

**Hazırlayanlar:**

**Kadir Sefa ÜNAL – 132805008**

**Kadir MUTLU - 132805013**

---

# *İçindekiler*

---

1. Amaç .....	3
2. Girdiler .....	3
3. Mantığı .....	3
4. Program Kodu .....	3
5. Ekran Çıktısı .....	8
6. Test Senaryoları .....	9

# 1. Amaç

---

Satranç At Oyununda amaç satranç oyunundaki at taşının hareketleri baz alınarak tüm kareleri gezmesi ile belirlenen alanda gezilmedik alan kalmayacak şekilde oyunu bitirmek ve oyunun değişik durumlarda verdiği sonuçları karşılaştırarak test işlemine tabi tutmak amaçlanmıştır.

## 2. Girdiler

---

- En: Kullanıcının eni ve boyu belirlediği girdidir. Satranç da kare şeklinde olduğu için en boy oranı eşittir.
- Hareket Edilen Kare: Kullanıcının başladığı noktadan açık olan yeşil kareler tıklama girdisi.

## 3. Mantığı

---

Oyunda öncelikle oynanacak alan boyutları belirlenir. Bu alanda bir kare seçilerek başlangıç noktası belirlenir.

Başlangıç noktasına tıklandığında satranç oyunundaki at taşının hareket bölgelerine göre yeşil kareler belirlenir. Yeşil kareler kullanıcıyı oynayabileceği alanlardır. Bir kareden en fazla bir kere geçilebilir ve bu kareler kırmızı renkte gösterilir.

Hareket alanı bitene kadar oyun devam eder. Eğer oyun boş alan kalmayacak şekilde bitirilirse başarı ile tamamlanır.

## 4. Program Kodu

---

```
public partial class frmSatranc : Form
{
    public frmSatranc()
    {
        InitializeComponent();

        //Daha önce geçmediyse tag = 0
        //Daha önce geçtiyse tag = 1
        //Klavuz noktalar ise tag = 2
        Button[,] butonlar;
        int en, boy;
        int skor;

        private void btnOlustur_Click(object sender, EventArgs e)
        {
            if (txtEn.Text != "" && txtBoy.Text != "" &&
```

```

        SayiMi(txtEn.Text) && SayiMi(txtBoy.Text))
    {
        en = Convert.ToInt32(txtEn.Text);
        boy = Convert.ToInt32(txtBoy.Text);

        if (en > 4 && en < 11)
        {
            Olustur();
            btnOlustur.Enabled = false;
        }
        else
        {
            MessageBox.Show("Lütfen aralık arasında değer girin!(5-10)",
"UYARI",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        else
        {
            MessageBox.Show("Lütfen sayı girin!", "UYARI",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }

    private void Olustur()
    {
        butonlar = new Button[boy, en];
        Button btn;
        for (int i = 0; i < boy; i++)
        {
            for (int j = 0; j < en; j++)
            {
                btn = new Button();
                btn.Name = "btn" + i + "_" + j;
                btn.Width = 30;
                btn.Height = 30;
                btn.Top = 70 + (i * 30);
                btn.Left = 13 + (j * 30);
                btn.FlatStyle = FlatStyle.Flat;
                btn.Tag = 2;
                btn.Click += Btn_Click1;
                butonlar[i, j] = btn;
            }
        }
        foreach (var item in butonlar)
        {
            Controls.Add(item);
        }
    }

    private void Btn_Click1(object sender, EventArgs e)
    {
        Button btn = (sender as Button);
        ButonKonumBul(btn);
        if (Convert.ToInt32(btn.Tag) == 2)
        {
            skor++;
            btn.Tag = 1;
            btn.BackColor = Color.Red;
            foreach (Button item in butonlar)
            {
                if (Convert.ToInt32(item.Tag) != 1)
                {
                    item.Tag = 0;
                    item.BackColor = default(Color);
                }
            }
        }
    }

```

```

        }
        KlavuzNoktaOlustur();
        Kontrol();
    }
    else
        MessageBox.Show("Buraya hareket edemezsiniz!");
}

int x, y;
private void ButonKonumBul(Button buton)
{
    for (int i = 0; i < boy; i++)
    {
        for (int j = 0; j < en; j++)
        {
            if (buton == butonlar[i, j])
            {
                x = i;
                y = j;
            }
        }
    }
}

private void KlavuzNoktaOlustur()
{
    int a, b;

    a = x - 2;
    if (a >= 0)
    {
        b = y - 1;
        if (b >= 0)
            KlavuzYap(a, b);
        b = y + 1;
        if (b < en)
            KlavuzYap(a, b);
    }

    a = x + 2;
    if (a < boy)
    {
        b = y - 1;
        if (b >= 0)
            KlavuzYap(a, b);
        b = y + 1;
        if (b < en)
            KlavuzYap(a, b);
    }

    b = y - 2;
    if (b >= 0)
    {
        a = x - 1;
        if (a >= 0)
            KlavuzYap(a, b);
        a = x + 1;
        if (a < boy)
            KlavuzYap(a, b);
    }
}

```

```

        b = y + 2;
        if (b < en)
        {
            a = x - 1;
            if (a >= 0)
                KlavuzYap(a, b);
            a = x + 1;
            if (a < boy)
                KlavuzYap(a, b);
        }
    }

    private void KlavuzYap(int a, int b)
    {
        if (Convert.ToInt32(butonlar[a, b].Tag) == 0)
        {
            butonlar[a, b].BackColor = Color.LightGreen;
            butonlar[a, b].Tag = 2;
        }
    }

    private void txtEn_KeyUp(object sender, KeyEventArgs e)
    {
        txtBoy.Text = txtEn.Text;
    }

    private void Kontrol()
    {
        bool klavuz = false, tam = true, bittiMi = false;
        foreach (Button btn in butonlar)
        {
            if (Convert.ToInt32(btn.Tag) == 2)
            {
                klavuz = true;
                break;
            }
        }
        foreach (Button btn in butonlar)
        {
            if (Convert.ToInt32(btn.Tag) != 1)
            {
                tam = false;
                break;
            }
        }

        if (tam)
        {
            MessageBox.Show("Tebrikler! Kazandınız.\nSkorunuz: " +
                skor.ToString());
            bittiMi = true;
        }
        else if (!klavuz && !tam)
        {
            MessageBox.Show("Kaybettiniz! \nHareket edebileceğiniz yer
                kalmadı!\nSkorunuz: " + skor.ToString());
            bittiMi = true;
        }

        if (bittiMi)
        {

```

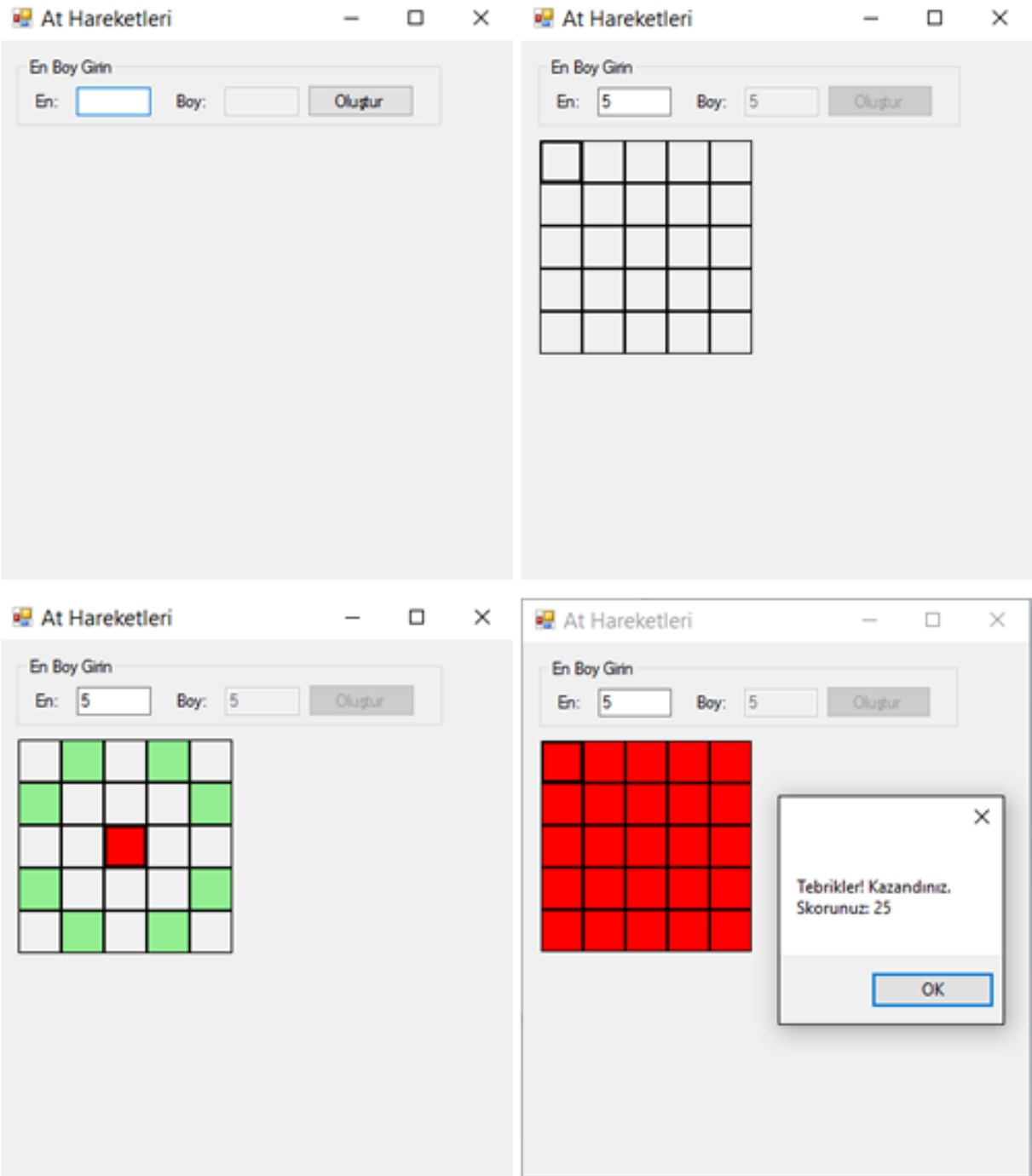
```
        DialogResult dr = MessageBox.Show("Tekrar oynamak ister misiniz?",
"Yeniden dene!",
                                                MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
        if (dr == DialogResult.Yes)
            Temizle();
        else
            Application.Exit();
    }

}

public bool SayiMi(string ifade)
{
    foreach (char chr in ifade)
    {
        if (!Char.IsNumber(chr)) return false;
    }
    return true;
}

private void Temizle()
{
    skor = 0;
    en = 0;
    boy = 0;
    txtEn.Clear();
    txtBoy.Clear();
    btnOlustur.Enabled = true;
    if (butonlar != null)
    {
        foreach (Button btn in butonlar)
        {
            Controls.Remove(btn);
        }
    }
}
}
```

## 5. Ekran Çıktısı





## 6. Test Senaryoları

---

- ✓ Girdilerin sayı olup olmadığı kontrolü başarılı
- ✓ Oyun alanındaki yeşil kare seçenekleri dışında hiçbir alana kullanıcı tıklayamaz.
- ✓ Yeşil alan dışında bir yere tıklandığında uyarı verir.
- ✓ Hareket alanı bittiğinde boş kare kalırsa oyun başarısızlıkla sonuçlanır.
- ✓ Bütün kareler doldurulur ise oyun başarı ile tamamlanır.
- ✓ Oyun başladıktan sonra tekrar alanın en boy oranı seçimi yapılamaz.
- ✓ Alan oluşturmak için kullanılan oluştur butonu oyunun en başında kullanılır. Oyun sırasında kullanılamaz.