

**MANİSA CELAL BAYAR ÜNİVERSİTESİ
HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

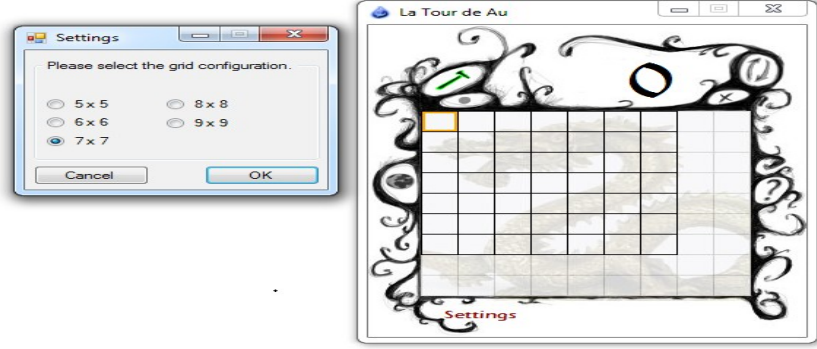
YZM 2122 Yazılım Yapımı Dersi ÖDEV – 2 (MART-2018)

Danışman : Yrd. Doç. Dr. Emin BORANDAĞ

Hazırlayan: Orçun ÖZDİL 172803065

ÖDEV-1

Oyun Alanı 5*5
ila 9*9 arasında
olacaktır.



Oyuncu
yapacağı her bir
hamle ile grid
üzerinde bir
alanı
kapatır. Amacı
daha önce
kullanılmayan
boş alanları
kullanarak
oyunda bulunan
bütün alanları
tamamlamaktır



Oyun
Tamamlandığın
da bir mesaj
ekranının alt
kısımında
kendini
gösterecektir.



Ödevin yapım aşamaları :

1. Oyun alanının oluşturulması:

Oyun alanı olarak 9 kolon ve 9 satırdan oluşan bir datagridview kullandım. Bunların yükseklik ve enlerini ayrlayarak düzgün bir görüntü sağladım. Bir form vasıtasıyla oyun alanını 5X5 den 9X9 a kadar boyutlandırma işlemi yaptım.

2. Renk ayıraçları ve click eventi:

Hücreleri birbirinden ayırmak için 4 tip renk belirledim.

pasif_renk (oyun alanı dışı hücre),

bloke_renk (hareket edilemeyen hücre),

uygun_renk (hareket edilebilecek hücre),

dolu_renk (önceden hamle yapılmış hücre).

İşlemleri datagridview click eventında yaptım.

3. Harket şekillerinin belirlenmesi:

Bulunan hücreden x-ekseninde +-2 y-ekseninde+- 1 veya x-ekseninde +-1 y-ekseninde+- 2 şeklinde hamleleri belirleyip bir diziye atadım. Datagridviewın kolonlarını X, satırlarını Y eksenini kullanarak her hamlede hücrelerin uygunluğunu kontrol ettim.

4. Hakkında Formu:

Bu form ile oyuncuya oyun hakkında kısa bir bilgi sundum

5. Skor tespiti:

Her hamle sayısını bir sayaçta tutarak kullanıcıya gösterdim. Properties.Settings.Default alanında tuttuğum highscore değişkeni ile yüksek skor kontrolü ve güncellemesi yaptım.

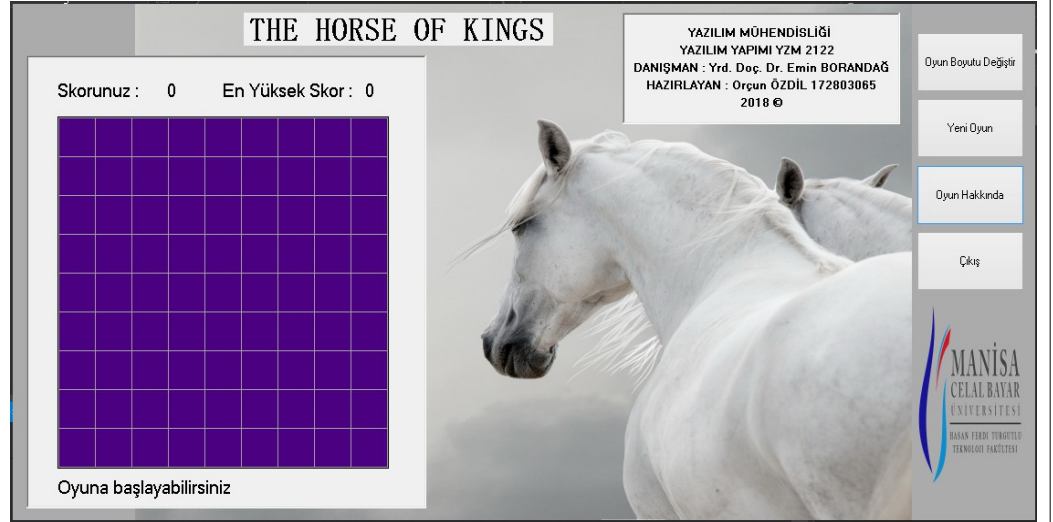
6. Oyun Sonucu:

Artık hamle yapılacak alan kalmadığında boardun tamamlanıp tamamlanmadığına bakar, oyun sonucunu oyuncuya gösterdim.(Kazandınız-Kaybettiniz şeklinde)

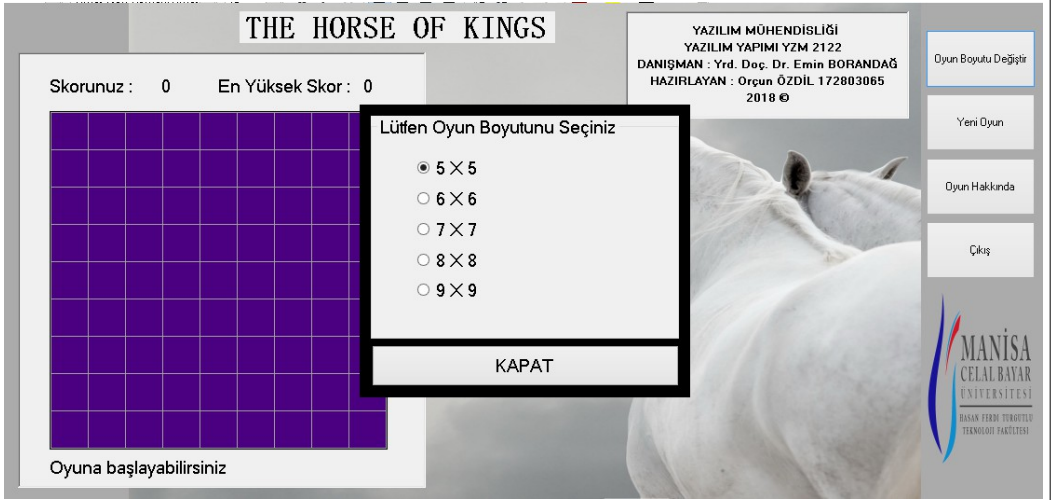
7. <https://github.com/Oozdil/Yzm-2122-HorseOfKings2>

Ödevin görselleri:

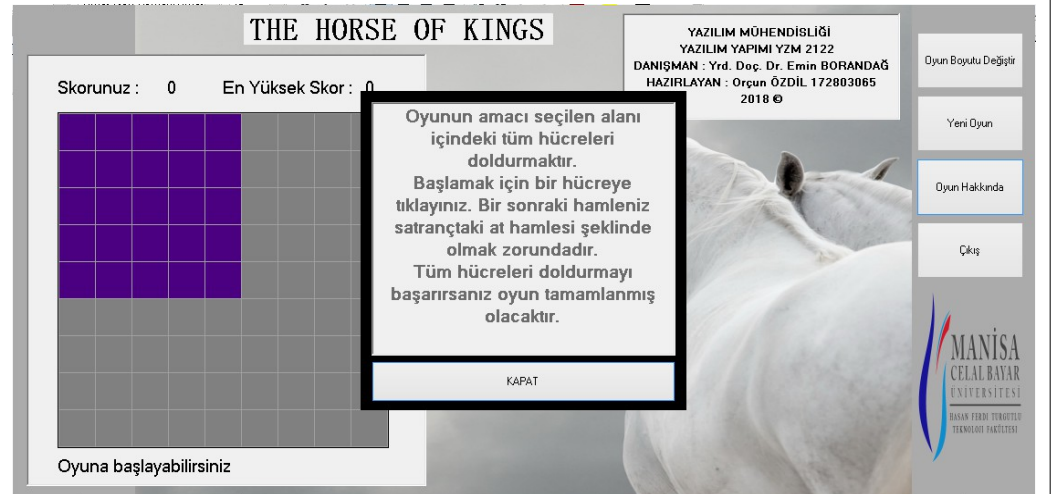
Açılış Ekranı



Alan boyutu seçimi



Oyun hakkında



Oyunun oynanışı

THE HORSE OF KINGS

Skorunuz : 1

En Yüksek Skor : 0

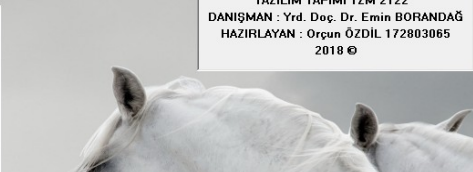
YAZILIM MÜHENDİSLİĞİ

YAZILIM YAPIMI YZM 2122

DANIŞMAN : Yrd. Doç. Dr. Emin BORANDAĞ

HAZIRLAYAN : Orçun ÖZDİL 172803065

2018 ©




Oyun Boyutu Değişir

Yeni Oyun

Oyun Hakkında

Çıkış



MANİSA
CELAL BAYAR
ÜNİVERSİTESİ
BAŞAN FİRKİ TEKNOLOJİ
TEKNOLOJİ FAKÜLTESİ

Oyuna başlayabilirsiniz

Kazandınız Örneği

[illegible]

Kaybettiniz Örneği

THE HORSE OF KINGS

Skorunuz : 19
En Yüksek Skor : 25

3	16	7	10	5					
	11	4	15	8					
17	2	9	6						
12			19	14					
1	18	13							

Üzgünüzü kaybettiniz

YAZILIM MÜHENDİSLİĞİ
YAZILIM YAPIMI YZM 2122
DANIŞMAN : Yrd. Doç. Dr. Emin BORANDAĞ
HAZIRLAYAN : Orçun ÖZDİL 172803065
2018 ©

Oyun Boyutu Değiştir

Yeni Oyun

Oyun Hakkında

Çıkış

MANİSA
CELAL BAYAR
ÜNİVERSİTESİ
BİLGİ TEKNOLOJİLERİ
FAKÜLTESİ

Çıkış İşlemi

[illegible]

Kaynak Kodlar :

Form1.cs

```
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 HorseOfKings
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        #region Degiskenler
        Color pasif_renk = System.Drawing.Color.Gray;
        Color bloke_renk = System.Drawing.Color.SandyBrown;
        Color uygun_renk = System.Drawing.Color.Indigo;
        Color dolu_renk = System.Drawing.Color.GreenYellow;
        int hamleSayisi;
        int[,] hareket_yerleri = { { -2, 1 }, { 1, 2 }, { 2, -1 }, { -1, -2 } //saat yönünde dönüşler
                                   { -2, -1 }, { -1, 2 }, { 2, 1 }, { 1, -2 } //saat yönünün tersinde dönüşler
                                   };
        #endregion

        private void Form1_Load(object sender, EventArgs e)
        {
            OyunAlaniAyarla();
        }

        private void OyunAlaniAyarla()
        {
            dGv_Game.ColumnCount = 9;
            dGv_Game.RowCount = 9;
            for (int i = 0; i < 9; i++)
            {
                dGv_Game.Columns[i].Width = 40;
                dGv_Game.Rows[i].Height = 40;
            }
            dGv_Game.ClearSelection();
            int boardsize = Properties.Settings.Default.BoardSize;
            for (int i = 0; i < 9; i++)
```

```

{
    for (int j = 0; j < 9; j++)
    {

        dGv_Game.Rows[i].Cells[j].Style.BackColor = uygun_renk;
        dGv_Game.Rows[i].Cells[j].Value = "";
        if (i >= boardsize || j >= boardsize)
        {
            dGv_Game.Rows[i].Cells[j].Style.BackColor = pasif_renk;
        }
    }
}
dGv_Game.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

}
dGv_Game.DefaultCellStyle.Font = new Font("Tahoma", 15);
hamleSayisi = 0;
lbl_score.Text = hamleSayisi.ToString();
lbl_high_score.Text = Properties.Settings.Default.HighScore.ToString();
lbl_message.Text = "Oyuna başlayabilirsiniz";
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    //Hangi butona basıldığını tespit ettik

    int cell_x = e.RowIndex;
    int cell_y = e.ColumnIndex;
    DataGridViewCell secilen_cell = dGv_Game.Rows[cell_x].Cells[cell_y];
    if (secilen_cell.Style.BackColor != dolu_renk && secilen_cell.Style.BackColor !=
pasif_renk && secilen_cell.Style.BackColor != bloke_renk)
    {
        dGv_Game.Rows[cell_x].Cells[cell_y].Style.BackColor = dolu_renk;
        dGv_Game.Rows[cell_x].Cells[cell_y].Value = (++hamleSayisi).ToString();
        lbl_score.Text = hamleSayisi.ToString();

        // dolu olmayan ve oyun alanı içinde olan tüm butonları bloke ettik

        foreach (DataGridViewRow row in dGv_Game.Rows)
        {
            foreach (DataGridViewCell cell in row.Cells)
            {
                if (cell.Style.BackColor != dolu_renk && cell.Style.BackColor != pasif_renk)
                    cell.Style.BackColor = bloke_renk;
            }
        }
    }

    //yapılabilecek hamleleri buluyoruz
    int kalanhamle = 0;
    for (int i = 0; i < 8; i++)

```



```

{
    int cell_new_x = cell_x + hareket_yerleri[i, 0];
    int cell_new_y = cell_y + hareket_yerleri[i, 1];

    //buton uyona alanında ve dolu değilse uygundur
    try
    {
        DataGridViewCell cell = dGv_Game.Rows[cell_new_x].Cells[cell_new_y];

        if (cell.Style.BackColor != dolu_renk && cell.Style.BackColor != pasif_renk)
        {
            cell.Style.BackColor = uygun_renk;
            kalanhamle++;
        }
    }
    catch
    {
    }
}

if (kalanhamle == 0)
{
    int score = Convert.ToInt32(lbl_score.Text);
    if (score == Math.Pow(Properties.Settings.Default.BoardSize, 2))
    {
        // MessageBox.Show("Tebrikler kazandınız");
        lbl_message.Text = "Tebrikler kazandınız";
    }
    else
    {
        // MessageBox.Show("Üzgünüm kaybettiniz");
        lbl_message.Text = "Üzgünüm kaybettiniz";
    }

    if (score > Properties.Settings.Default.HighScore)
    {
        Properties.Settings.Default.HighScore = hamleSayisi;
        Properties.Settings.Default.Save();
        lbl_high_score.Text = Properties.Settings.Default.HighScore.ToString();
    }
}

dGv_Game.ClearSelection();
}

private void btn_exit_Click(object sender, EventArgs e)
{

```

```

        DialogResult karar = MessageBox.Show("Çıkmak istediğinize emin misiniz ?", "DİKKAT
ÇIKIŞ YAPILACAK",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
        if (karar == DialogResult.Yes)
            this.Close();
    }

    private void btn_about_Click(object sender, EventArgs e)
    {
        AboutForm aboutform = new AboutForm();
        aboutform.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
        aboutform.MaximizeBox = false;
        aboutform.MinimizeBox = false;
        aboutform.ShowDialog();
    }

    private void btn_board_size_Click(object sender, EventArgs e)
    {
        BoardSizeForm boardsizeform = new BoardSizeForm();
        boardsizeform.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
        boardsizeform.MaximizeBox = false;
        boardsizeform.MinimizeBox = false;
        boardsizeform.ShowDialog();
        OyunAlaniAyarla();
    }

    private void btn_new_game_Click(object sender, EventArgs e)
    {
        OyunAlaniAyarla();
    }

}
}

```


AboutForm.cs

```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace HorseOfKings
{
    public partial class AboutForm : Form
    {
        public AboutForm()
        {
            InitializeComponent();

            private void AboutForm_Load(object sender, EventArgs e)
            {
                string Mesaj = @"Oyunun amacı seçilen alanı içindeki tüm hücreleri
doldurmaktır."+Environment.NewLine+
                @"Başlamak için bir hücreye tıklayınız. Bir sonraki hamleniz satrançtaki at
hamlesi şeklinde olmak zorundadır. "+Environment.NewLine+
                @"Tüm hücreleri doldurmayı başarırsanız oyun tamamlanmış olacaktır.";
                richTextBox1.Text = Mesaj;
                richTextBox1.Enabled = false;
                richTextBox1.SelectAll();
                richTextBox1.SelectionAlignment = HorizontalAlignment.Center;

            }

            private void btn_close_Click(object sender, EventArgs e)
            {
                this.Close();
            }
        }
    }
}
```