

Özet - Bu projede nesneye yönelik programlama ile C# ve Unity kullanarak sporcu kart oyunu tasarladık.

Anahtar Kelimeler - Nesneye Yönelik Programlama, C#, Unity, Kart Oyunu, Strateji.

I. GİRİŞ

Bu proje temelde projeyi yapan kişilerin nesneye yönelik programlama yeteneklerini geliştirmeye amaçlanmıştır.

Bu projeye başladığımızda hangi dili kullanacağımızı ve oyunun görselini nasıl yapacağımızı düşündük. İlk önce oyunu konsolda çalışacak şekilde java yazılım diliyle yazdık akabinde görselleştirme için Unity kullanmak istediğimizden dolayı yazdığımız kodu C# yazılım dilinde yazdık.

II. KARŞILAŞILAN PROBLEMLER

İlk karşılaştığımız problem C# yazılım dilinde super() komutunun javaya göre biraz farklı olmasıydı biraz araştırma sonucu bu sorunu hallettik.

İkinci karşılaştığımız problem ise bilgisayarın kart seçiminde seçtiği kartı tekrar kullanmasıydı, bunu da kullanıldımı boolean değerini bir döngüde kullanıp çözdük.

III. GENEL YAPI

KULLANICI KISMI

Proje başlatıldığında ekrana Unity ile yaptığımız oyun çıkar. Kartlarımız önümüze dizilir ve her kartın altında kart seçme butonu vardır, kartların üstünde özellikleri yer almaktadır, kartı seçtikten sonra oyna butonuna bastığınızda kartımızı oyuna sürüklemiş oluruz. Ekranın solunda bulunan tekrar oyna butonuna basarak yeni bir deste dağıtıp oyuna tekrar başlayabiliriz.



KOD KISMI

1. SPORCU SINIFI

```
public class Sporcu : MonoBehaviour
{
    string sporculisim;
    string sporcufakim;

    //baguru
    public Sporcu(string sporculisim, string sporcufakim)
    {
        this.sporculisim = sporculisim;
        this.sporcufakim = sporcufakim;
    }

    //baguru
    public Sporcu()
    {
    }

    //baguru
    public string Sporculisim
    {
        get
        {
            return sporculisim;
        }
        set
        {
            sporculisim = value;
        }
    }

    //baguru
    public string SporcuTakim
    {
        get
        {
            return sporcufakim;
        }
        set
        {
            sporcufakim = value;
        }
    }

    //baguru
    public virtual void sporcuPuanigoster()
    {
        Debug.Log("Sporcu puanı gösteriliyor!");
    }
}
```

Bize verilen doğrultuda constructor ve getter-setter metotlarını futbolcu ve basketbolcu sınıflarının kullanmak için sporcu sınıfı için oluşturduk.

2. FUTBOLCU SINIFI

```
public class Futbolcu : Sporcu
{
    public int penaltı;
    public int serbestAtış;
    public int kaleciKartıKariye;
    public bool isued = false;

    public string futbolculad;
    public string futbolcufakim;

    public GameObject gameobject;

    //baguru
    public Futbolcu(string sporculisim, string sporcufakim, int penaltı, int serbestAtış, int kaleciKartıKariye, bool isued, GameObject gameobject) : base(sporculisim, sporcufakim)
    {
        this.futbolculad = sporculisim;
        this.futbolcufakim = sporcufakim;
        this.penaltı = penaltı;
        this.serbestAtış = serbestAtış;
        this.kaleciKartıKariye = kaleciKartıKariye;
        this.isued = isued;
        this.gameobject = gameobject;
    }

    //baguru
    public Futbolcu()
    {
    }

    //baguru
    public override void sporcuPuanigoster()
    {
        base.sporcuPuanigoster();
        Debug.Log("Penaltı : " + this.penaltı);
        Debug.Log("Serbest Atış : " + this.serbestAtış);
        Debug.Log("Kaleci Kartı Kariye : " + this.kaleciKartıKariye);
    }

    //baguru
    public string futbolculad
    {
        get
        {
            return futbolculad;
        }
        set
        {
            futbolculad = value;
        }
    }

    //baguru
    public string futbolcufakim
    {
        get
        {
            return futbolcufakim;
        }
        set
        {
            futbolcufakim = value;
        }
    }

    //baguru
    public int penaltı
    {
        get
        {
            return penaltı;
        }
        set
        {
            penaltı = value;
        }
    }

    //baguru
    public int serbestAtış
    {
        get
        {
            return serbestAtış;
        }
        set
        {
            serbestAtış = value;
        }
    }

    //baguru
    public int kaleciKartıKariye
    {
        get
        {
            return kaleciKartıKariye;
        }
        set
        {
            kaleciKartıKariye = value;
        }
    }

    //baguru
    public bool isued
    {
        get
        {
            return isued;
        }
        set
        {
            isued = value;
        }
    }
}
```

Futbolcu verilerini tutan, verilere uygun getter setter metotları içeren ve Sporcu sınıfından kalıtılan bir Futbolcu sınıfını yazdık.

Sporcu sınıfından override ettiğimiz sporcuPuanınıGöster() fonksiyonu ile her karta özel bilgilerin ekrana bastırılmasını sağladık.

Ekran çıktısının görseli yukarıda verilmiştir.

Override kartSec(int a) fonksiyonu: Oyuncu sınıfındaki bu fonksiyonu bilgisayar sınıfımızda override ediyoruz ve gelen parametreye göre iki sayı aralığında rastgele sayı alıyoruz, iki sayı aralığında rastgele sayı almamızın sebebi kullanıcı futbol kartını seçince futbolcu kartı seçmek ya da aynı şekilde basketbolcu seçilince basketbolcu kartı seçmek.

TEST SINIFI

```
void Start()
{
    OyunHazirla();
}

1 bayram
public void OyunHazirla()
{
    listehazirlaFutbol();
    listehazirlaBasket();
    oyuncu.listeKaristir();
    listeDir();
}
```

Oyunun başlangıcında OyunHazirla() fonksiyonun çağırıyoruz.
OyunHazirla() fonksiyonun içinde 4 tane farklı fonksiyon bulunuyor.

```
void listFutbolClub(FutbolClub*)
{
    FutbolClub futbolClub = new FutbolClub("Barcelona", "Messi", 10, 100, 100, false, futbolClub(0));
    FutbolClub futbolClub2 = new FutbolClub("Cruzeiro", "Reis", 17, 50, 50, false, futbolClub(1));
    FutbolClub futbolClub3 = new FutbolClub("Vasco", "Tomaz", 10, 50, 50, false, futbolClub(2));
    FutbolClub futbolClub4 = new FutbolClub("Goiia", "Romarinho", 50, 50, 50, false, futbolClub(3));
    FutbolClub futbolClub5 = new FutbolClub("Grana Filmes", "Washington", 70, 70, 70, false, futbolClub(4));
    FutbolClub futbolClub6 = new FutbolClub("Grash Berol", "Ramsden", 60, 50, 50, false, futbolClub(5));
    FutbolClub futbolClub7 = new FutbolClub("Grash Berol", "Reis", 50, 50, 50, false, futbolClub(6));
    FutbolClub futbolClub8 = new FutbolClub("Real Madrid", "Benzema", 50, 50, 50, false, futbolClub(7));
    FutbolClub futbolClub9 = new FutbolClub("Real Madrid", "Benzema", 50, 50, 50, false, futbolClub(8));

    FutList.Add(futbolClub);
    FutList.Add(futbolClub2);
    FutList.Add(futbolClub3);
    FutList.Add(futbolClub4);
    FutList.Add(futbolClub5);
    FutList.Add(futbolClub6);
    FutList.Add(futbolClub7);
    FutList.Add(futbolClub8);
    FutList.Add(futbolClub9);

    for (int i = 0; i < FutList.Count; i++)
    {
        System.Console.WriteLine(FutList.ElementAt(i));
    }
}

void listBasketBall(BasketBall*)
{
    BasketBall basketBall = new BasketBall("Vasco Grana", "L. L. Lima", 100, 100, 100, false, basketBall(0));
    BasketBall basketBall2 = new BasketBall("Grash Berol", "Collins", 77, 50, 50, false, basketBall(1));
    BasketBall basketBall3 = new BasketBall("Grash Berol", "Collins", 40, 50, 50, false, basketBall(2));
    BasketBall basketBall4 = new BasketBall("Michael Jordan", "L. L. Lima", 50, 50, 50, false, basketBall(3));
    BasketBall basketBall5 = new BasketBall("Michael Jordan", "L. L. Lima", 50, 50, 50, false, basketBall(4));
    BasketBall basketBall6 = new BasketBall("Cedric Gaud", "L. L. Lima", 50, 50, 50, false, basketBall(5));
    BasketBall basketBall7 = new BasketBall("Vasco Grana", "George Mello", 67, 60, 70, false, basketBall(6));
    BasketBall basketBall8 = new BasketBall("L. L. Lima", 40, 50, 50, false, basketBall(7));

    basketList.Add(basketBall);
    basketList.Add(basketBall2);
    basketList.Add(basketBall3);
    basketList.Add(basketBall4);
    basketList.Add(basketBall5);
    basketList.Add(basketBall6);
    basketList.Add(basketBall7);
    basketList.Add(basketBall8);

    for (int i = 0; i < basketList.Count; i++)
    {
        System.Console.WriteLine(basketList.ElementAt(i));
    }
}
```

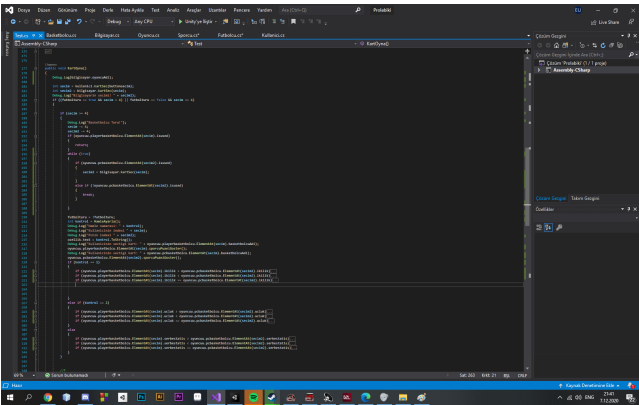
listehazirlaFutbol() ve listehazirlaBasket() fonksiyonları ile kartlarımızı yaratıyor ve onlara değerlerini verip oyuncu sınıfındaki listeye ekliyoruz.

```

public void listBizi()
{
    for (int i = 0; i < 4; i++)
    {
        oyuncuu.playerbasketbolcu.ElementAt(i).gameObject.transform.position = slots[i+4].position;
    }
    for (int i = 0; i < 4; i++)
    {
        oyuncuu.playerfutbolcu.ElementAt(i).gameObject.transform.position = slots[i].position;
    }
    for (int i = 0; i < 4; i++)
    {
        oyuncuu.pcbasketbolcu.ElementAt(i).gameObject.transform.position = slotspc[i + 4].position;
        oyuncuu.pcbasketbolcu.ElementAt(i).gameObject.transform.rotation = slotspc[i + 4].rotation;
    }
    for (int i = 0; i < 4; i++)
    {
        oyuncuu.pcfutbolcu.ElementAt(i).gameObject.transform.position = slotspc[i].position;
        oyuncuu.pcfutbolcu.ElementAt(i).gameObject.transform.rotation = slotspc[i].rotation;
    }
}

```

listeDiz() fonksiyonu ile oyun motorumuzla iletişime geçip her kartın ait olduğu oyun objesini pozisyonunu ve rotasyonunu alarak kart konumlarına gönderiyoruz.



Oyun içinde oyna butonuna tıkladığımızda çalışan fonksiyonu budur. Buton seçim değeri tıkladığımız butonlara göre verilir ve kullanıcının kartsec fonksiyonuna index olarak gönderilir ve bu seçim değeri bilgisayarın da seçimini etkileyeceğinden dolayı bilgisayarın kartsec fonksiyonun parametresi kullanıcının seçtiği kart indexidir.

Kart seçiminden sonra kartın daha önce kullanılıp kullanılmadığı test edilir. Seçilen kartın tur şekline uyup uymadığı kontrol edilir, eğer bu kontrollerden geçemezse oyna tuşuna basınca herhangi bir etkileşim olmayacaktır. Eğer kontrolden başarılı bir şekilde geçerse oyun başlayacaktır seçilen kartların özellikleri karşılaştırılacaktır. Karşılaştırılacak özelliği seçmek için hamleAyarla() fonksiyonu kullanılır ve bu 1,2,3 değerlerini döndürür, döndürdüğü değere göre kartların o sıradaki özellikleri birbiriyle kıyaslanır ve hangi oyuncunun kartı kazanırsa o oyuncunun skoru artırılır, o kart kullanılmış olarak ayarlanır ve arayüz güncellenerek skoru da eşzamanlı görmemiz sağlanır.

```

if (Control == 1)
{
    if (opnumcu.playerbacketholic.ElementAt(secin).IsNil() == opnumcu.pchasetholic.ElementAt(secin2).IsNil())
    {
        nullmilci.shor == 1;
        nullmilci.shor = 1;
        arrayGameCnic1Ethasetholic[secin, secin2];
        nullmilci.trest, text = nullmilci.shor.Yndring();
        Sound, lang"Voice from All"??;
        opnumcu.playerbacketholic.ElementAt(secin).Issued = true;
        opnumcu.pchasetholic.ElementAt(secin2).Issued = true;

        Destroy(opnumcu.playerbacketholic.ElementAt(secin).gameObject);
        Destroy(opnumcu.pchasetholic.ElementAt(secin2).gameObject);
    }
    if (opnumcu.playerbacketholic.ElementAt(secin).IsNil() == opnumcu.pchasetholic.ElementAt(secin2).IsNil())
    {
        nulligayer.shor == 1;
        arrayGameCnic1Ethasetholic[secin, secin2];
        actext.text = nulligayer.shor.Yndring();
        Sound, lang"Voice from All"??;
        opnumcu.playerbacketholic.ElementAt(secin).Issued = true;
        opnumcu.pchasetholic.ElementAt(secin2).Issued = true;

        Destroy(opnumcu.playerbacketholic.ElementAt(secin).gameObject);
        Destroy(opnumcu.pchasetholic.ElementAt(secin2).gameObject);
    }
    if (opnumcu.playerbacketholic.ElementAt(secin).IsNil() == opnumcu.pchasetholic.ElementAt(secin2).IsNil())
    {
        if (layer == 7)
        {
            soundManager.Play(controller);
        }
        Destroy(lang"Voice")
        arrayGameCnic1Ethasetholic[secin, secin2];
        opnumcu.playerbacketholic.ElementAt(secin).Issued = false;
        opnumcu.pchasetholic.ElementAt(secin2).Issued = false;
    }
}

```

Karşılaştırma kodları yukarıda görüldüğü gibidir.

```

public void arayuzGuncelleBasketbolcu(int secim,int secim2)
{
    k1.text = oyuncuu.playerbasketbolcu.ElementAt(secim).ikilik.ToString();
    k2.text = oyuncuu.playerbasketbolcu.ElementAt(secim).ucluk.ToString();
    k3.text = oyuncuu.playerbasketbolcu.ElementAt(secim).serbestatis.ToString();

    p1.text = oyuncuu.pcbasketbolcu.ElementAt(secim2).ikilik.ToString();
    p2.text = oyuncuu.pcbasketbolcu.ElementAt(secim2).ucluk.ToString();
    p3.text = oyuncuu.pcbasketbolcu.ElementAt(secim2).serbestatis.ToString();
}
}
# Bayguru
public void arayuzGuncelleFutbolcu(int secim,int secim2)
{
    k1.text = oyuncuu.playerfutbolcu.ElementAt(secim).penalti.ToString();
    k2.text = oyuncuu.playerfutbolcu.ElementAt(secim).serbestatis.ToString();
    k3.text = oyuncuu.playerfutbolcu.ElementAt(secim).kaleciKarsiKarsiya.ToString();

    p1.text = oyuncuu.pcfutbolcu.ElementAt(secim2).penalti.ToString();
    p2.text = oyuncuu.pcfutbolcu.ElementAt(secim2).serbestatis.ToString();
    p3.text = oyuncuu.pcfutbolcu.ElementAt(secim2).kaleciKarsiKarsiya.ToString();
}
}
# Bayguru

```

Arayüz güncelleme kodu yukarıdaki gibidir. Textlerin Unity'nin UnityEngine.UI kütüphanesi sayesinde kullanılır.

```

1 public void sonDurumBeraberlik(int kontrol1,int kontrol2)
2 {
3     while (true)
4     {
5         KartOyna();
6         if (kontrol1 != kontrol2)
7         {
8             break;
9         }
10    }
11 }
12
13 2 basvuru
14 int HamleAyarla()
15 {
16     int hamle = Random.Range(1, 4);
17     return hamle;
18 }

```

Son durumda beraberlik olduğunda aynı kartlar geri gelmez onun yerine o kartların başka bir özelliği kıyaslanır ve bunu da sonDurumBeraberlik fonksiyonunda yapıyoruz.

Hamle ayarla fonksiyonunda da rastgele hamleyi seçiyoruz.

KAYNAKÇA

<https://docs.microsoft.com/tr-tr/dotnet/csharp/>

<https://docs.unity3d.com/Manual/index.html>

KULLANILAN TEKNOLOJİLER

Unity Oyun Motoru, C# Yazılım Dili ve IDE olarak Visual Studio kullandık.

