

**GAZİ ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**



**2021-2022 GÜZ DÖNEMİ**

**BMT-410 OYUN PROGRAMLAMA**

**ARAŞTIRMA ÖDEVİ**

**OYUN PROGRAMLAMADA SEVİYE ATLAMA**

**04.11.2021**

**ÖĞRENCİ BİLGİLERİ**

**191816768**

**EMRE ÇİFTÇİ**

**KASIM 2021**

## İÇİNDEKİLER

Resimler Tablosu.....	2
Şekiller Tablosu.....	3
Giriş. ....	4
Kullanılan Teknolojiler.....	4
Sistem Gereksinimleri.....	4
Kullanım Senaryoları, Fonksiyonlar ve Oyun Tasarımı.....	5
Kodlar .....	9
Kaynaklar .....	11

## RESİMLER LİSTESİ

Resim 3.1: Oyun Programlama geliştirirken kullanılan bilgisayarın sistem bilgisi.....	5
Resim 4.1: Package Manager'ın ekran görüntüsü.....	5
Resim 4.2: Bodyguard karakterinin ekran görüntüsü.....	6
Resim 4.3: Virtual Kamera'nın ekran görüntüsü.....	6
Resim 4.4: Terrain ile hazırlanan oyun platformunun ekran görüntüsü.....	6
Resim 4.5: Para modelinin bulunduğu yerin ekran görüntüsü.....	7
Resim 4.6: OyunAyar objesinin ayarları.....	7
Resim 4.7: Pasif olan bitisPaneli görüntüsü.....	8
Resim 4.8: seviyeAtla butonunun ayar görüntüsü.....	8
Resim 4.9: Build Settings panelinin görüntüsü.....	8

## ŞEKİLLER LİSTESİ

## 1. GİRİŞ

1961 yılında yapılan ilk dijital oyun olan Space War'dan bu yana gitgide kullanıcı sayısını arttıran dijital oyunlar, günümüzde mobil cihazlarında yaygınlaşmasıyla pazar kapasitesini oldukça arttırmıştır.

Bilgi Teknolojileri ve İletişim Kurumu (BTK), salgın süreciyle büyüyen ve gelişen oyun sektörüne ilişkin, "Güvenli İnternet Merkezi" tarafından hazırlanan "2020 Dijital Oyunlar Raporu"nda; Söz konusu sektörün, 2020 yılında COVID-19 salgınının da etkisiyle küresel ölçekte **yüzde 9,3** artarak, *159 milyar dolara* ulaştığına işaret etmiş ve pazarın 2026 yılına kadar 295 milyar doları aşmasının tahmin edildiği belirtilmiştir. Dijital oyun endüstrisinin, salgın döneminde en büyük artışı mobil oyunlarla sağladığına da yer verilen bu raporda, bu oyunlarla 2020'de 77,2 milyar dolar gelir elde edildiği bildirilmiştir.

Yine aynı raporda, Türkiye'de dijital oyun sektörünün her yıl olduğu gibi 2020 yılında da büyümeye devam ettiği, salgının da etkisiyle 2020 yılında özellikle mobil oyun sektöründe hızlı bir ilerlemenin yaşandığı belirtilmiş, sektörün geçen yıl bir önceki yıla kıyasla **yüzde 29** büyümeyle rekor kırdığı kaydedilmiştir.

Bu veriler ışığında oyun sektörünün ne kadar büyüdüğü ve pazarın ne kadar hızlı geliştiği görülmektedir. Bu sebeple, bu sektör içinde yer almak, hem mikro hem de makro milli ekonomi açısından oldukça büyük bir önem arz etmektedir.

## 2. KULLANILAN TEKNOLOJİLER

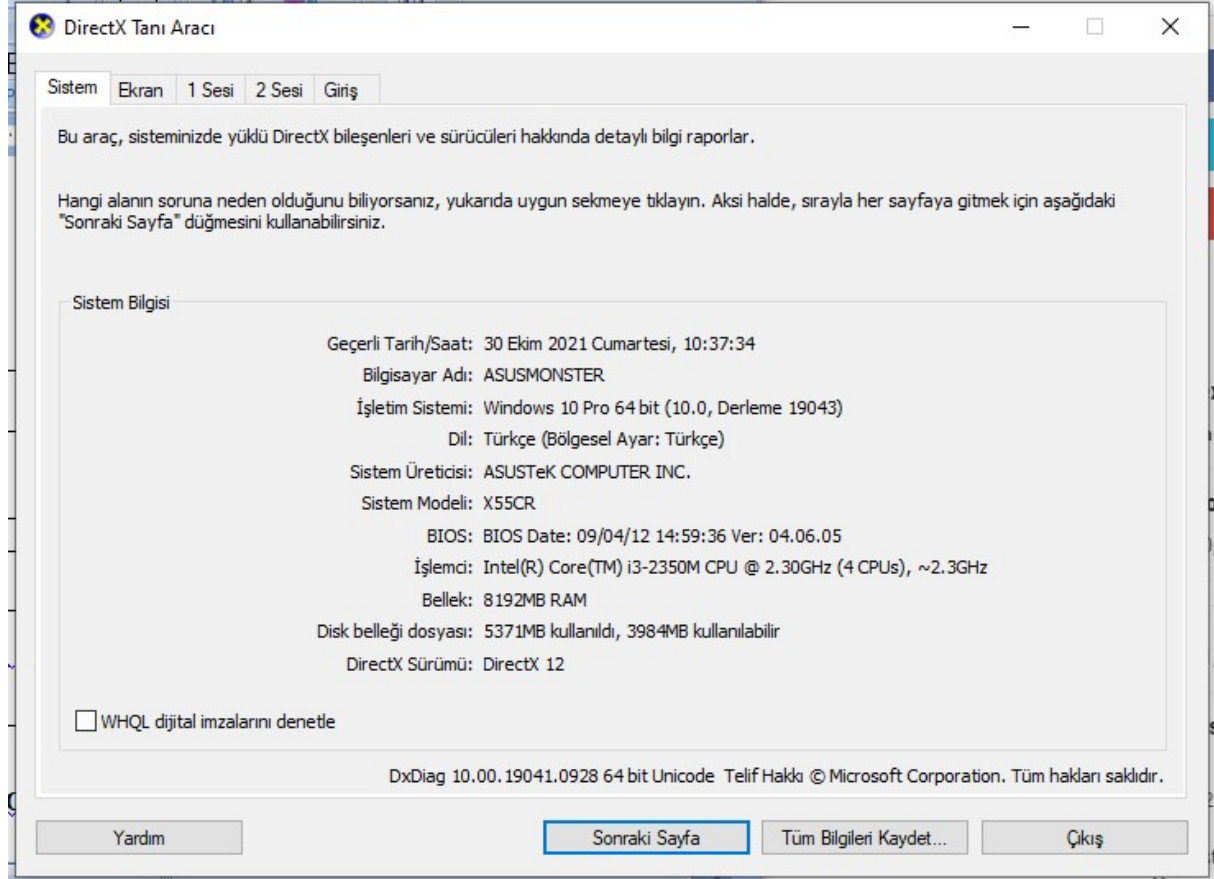
Oyun Motoru: Unity 2020.3.20f1 (64-bit)

Yazılım dili: C# (Visual Studio)

## 3. SİSTEM GEREKSİNİMLERİ

Minimum Gereksinimler	Windows	macOS	Linux
<b>İşletim Sistemi</b>	Windows 7 (SP1+) veya Windows 10 64-bit versiyonları (Sadece 64)	Sierra 10.13+ ve sonrası	Ubuntu 20.04, Ubuntu 18.04, ve CentOS 7
<b>CPU</b>	X64, SSE2	X64, SSE2	X64, SSE2
<b>Graphics API</b>	DX10, DX11, ve DX12(DirectX12)	Intel ve AMD	OpenGL 3.2+, Vulkan-capable, Nvidia veya AMD
<b>Ek gereksinimler</b>	Donanım satıcısı resmi olarak desteklenen sürücüler	Apple sürücülerini destekliyor.	X11 windowing system (Pencereleme sistemi) Nvidia resmi tescilli grafik sürücüsü veya AMD Mesa grafik sürücüsü üzerinde çalışan Gnome masaüstü ortamı.

## Oyunu Geliştirirken Kullanılan Sistem Bilgisi:

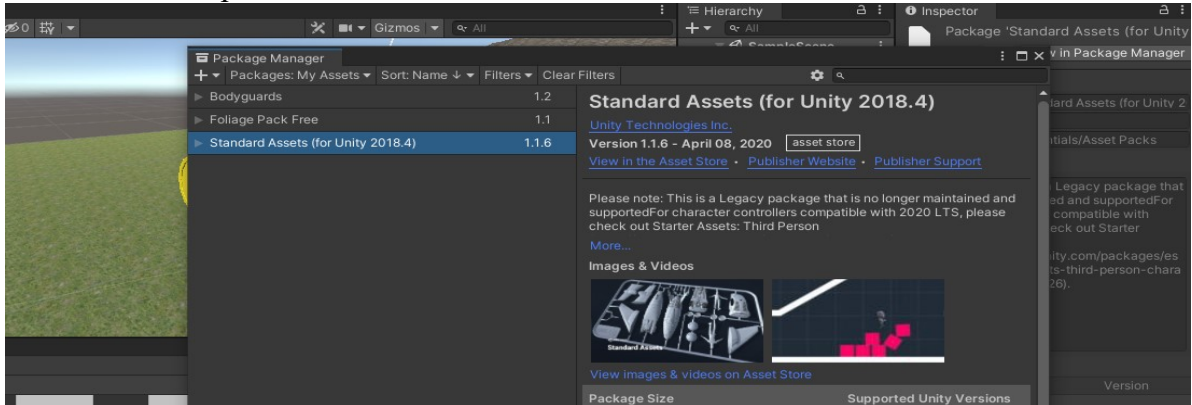


Resim 3.1: Oyun Programlama geliştirirken kullanılan bilgisayarın sistem bilgisi

## 4. KULLANIM SENARYOLARI, FONKSİYONLAR ve OYUN TASARIMI

Unity’de Seviye atlama mantığını göstermek için öncelikle basit bir **altın toplama** oyunu tasarladık. Kullanıcı kontrolünde olan Karakterimiz, platformdaki paralara çarptığı zaman, ekranın sol üstünde bulunan para miktarı artacak, OyunAyar.cs scriptimizde belirlediğimiz altın miktarına ulaşınca da *seviye atlama paneli* ekranda belirecek, Bu panel içinde bulunan *seviye atla* butonuna tıklanıldığında da yeni seviyeye geçilecektir. Bunları yaparken sırayla;

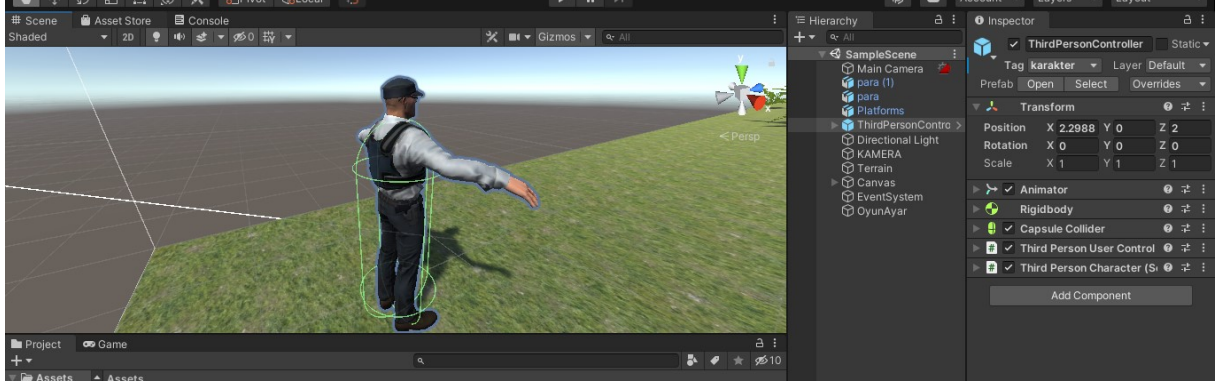
1- Window panelinde bulunan Packet manager’den Standart assets’i, Bodyguards’ı ve Cinemachine’i import ettik.



Resim 4.1: Package Manager’ın ekran görüntüsü

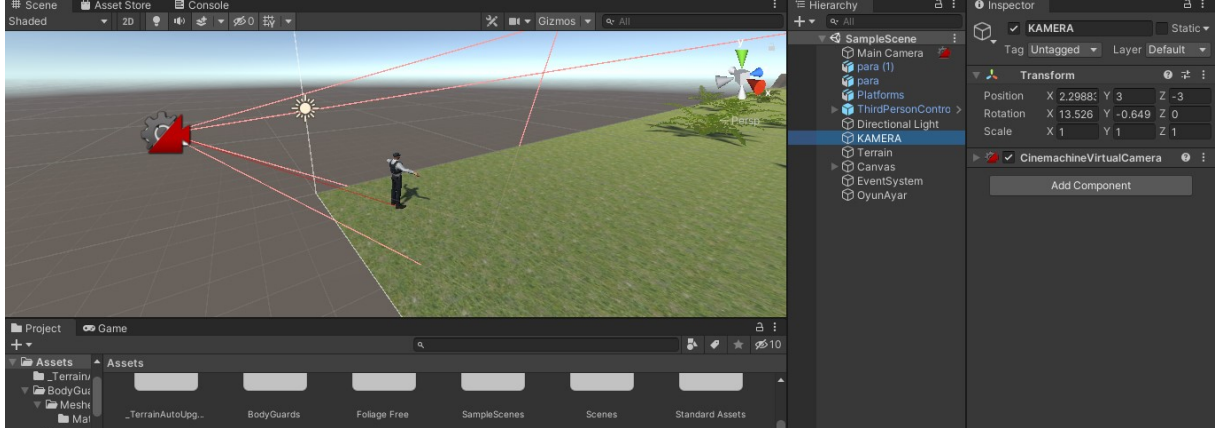
2- Standart assets klasörü içinde bulunan thirdpersoncontroller'in prefab'ını sahneye sürükleyerek default haldeki hazır karakteri ekledik. Bu sayede 3 boyutlu hareket yapacak bir karaktere sahip olmuş olduk.

3- Bodyguard assets'ın içinde bulunan Meshes klasöründen herhangi bir karakteri de sahneye sürükledik. Gerekli ayarları yaparak, default haldeki karakterin tüm özelliklerini bodyguard karakterimize ekledik ve default karakterin iskelet yapısını sildik. Bu sayede sahnede sadece bodyguard karakterimiz kaldı.



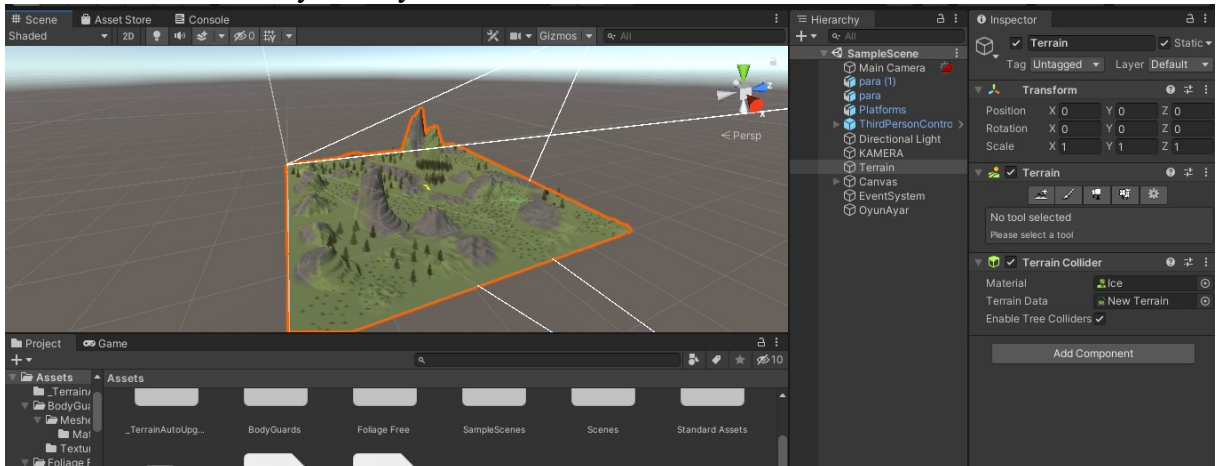
Resim 4.2: Bodyguard karakterinin ekran görüntüsü

4- Karakter işlemi sonlandıktan sonra, Cinemachine içinden virtual kamerayı ekleyerek gerekli ayarları yaptık. Bu sayede karakterimizi takip eden bir kamera yapmış olduk.



Resim 4.3: Virtual Kamera'nın ekran görüntüsü

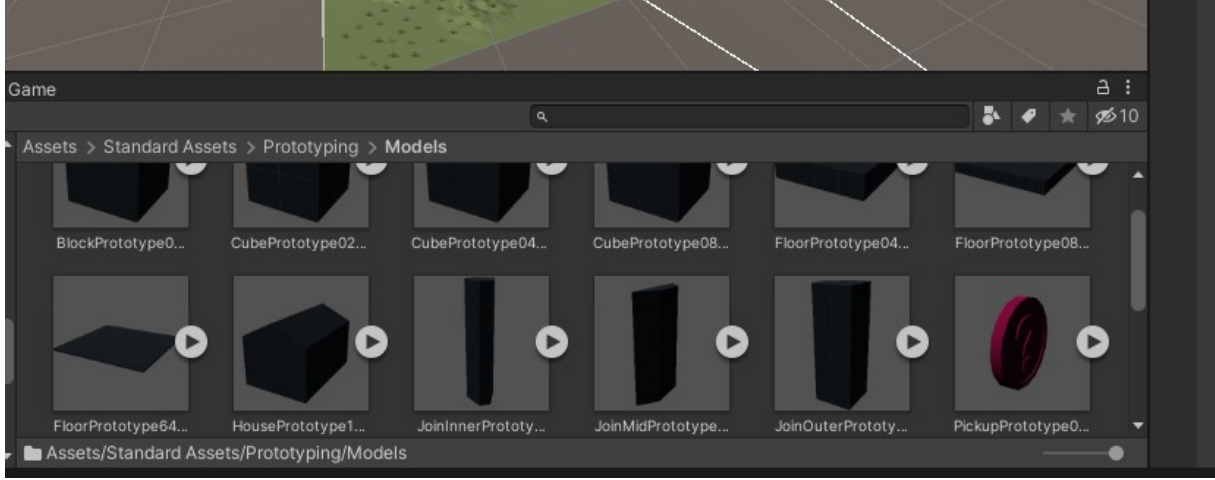
5- Terrain kullanarak oyun uzayı tasarladık.



Resim 4.4: Terrain ile hazırlanan oyun platformunun ekran görüntüsü

**Bu işlemlerin hepsi oyun tasarımı için gerekti. Şimdi gelelim Seviye atlama kısmına;**

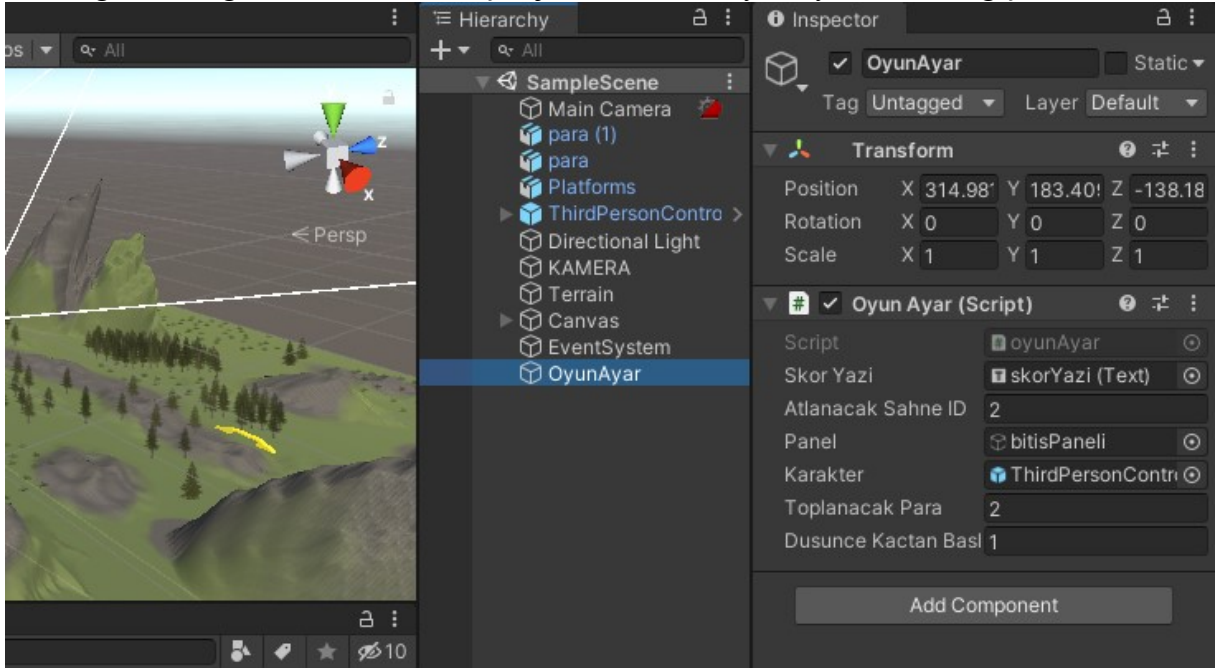
6- Standart Assets klasörü içinde bulunan Prototyping klaösrünü açıp, Model klasörü içindeki PickupPrototype01x01x01(para objesi) sahneye sürükledik.



Resim 4.5: Para modelinin bulunduğu yerin ekran görüntüsü

7- Karakterimiz bu paralara çarptıkça skorun artmasını, belirlediğimiz para miktarı bitince de seviye atlamasını sağlayacağımız kod bloğunu OyunAyar.cs içine yazdık.

8- Yazdığımız OyunAyar.cs scriptini, unity'de boş olarak oluşturduğumuz objenin içine atarak, gerekli değerleri atadık. Bu boş objenin adını da OyunAyar olarak değiştirdik.

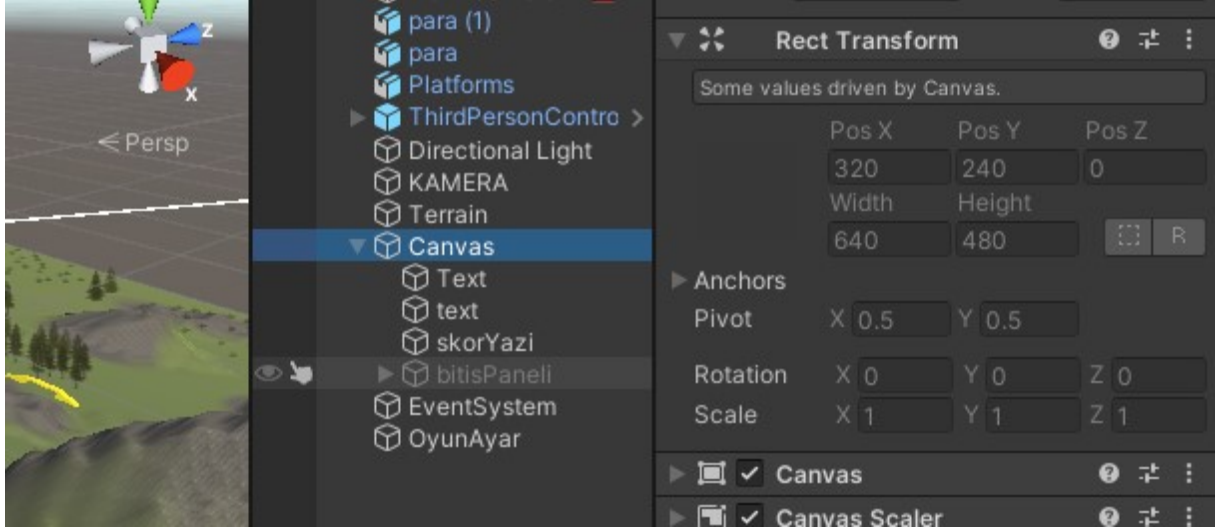


Resim 4.6: OyunAyar objesinin ayarları

9-paraKontrol.cs adında bir script daha oluşturarak, karakter herhangi bir para çarptığında, o paranın yok olmasını sağlayan kodumuzu yazdık. Bu scriptide, unity'de bulunan para objesinin içine attık. Çarpma fonksiyonu için OnCollisionEnter fonksiyonunu kullandık.

10- Karakter tüm paraları topladığında seviye atladığını anlaması ve yeni oyuna geçmesi için bitisPaneli adında bir canvas paneli oluşturduk. Bu panel sadece seviye atlandığı zaman gösterileceği için, GameObject-> Toogle Active State'e tıklayarak, Paneli Pasif hale getirdik.

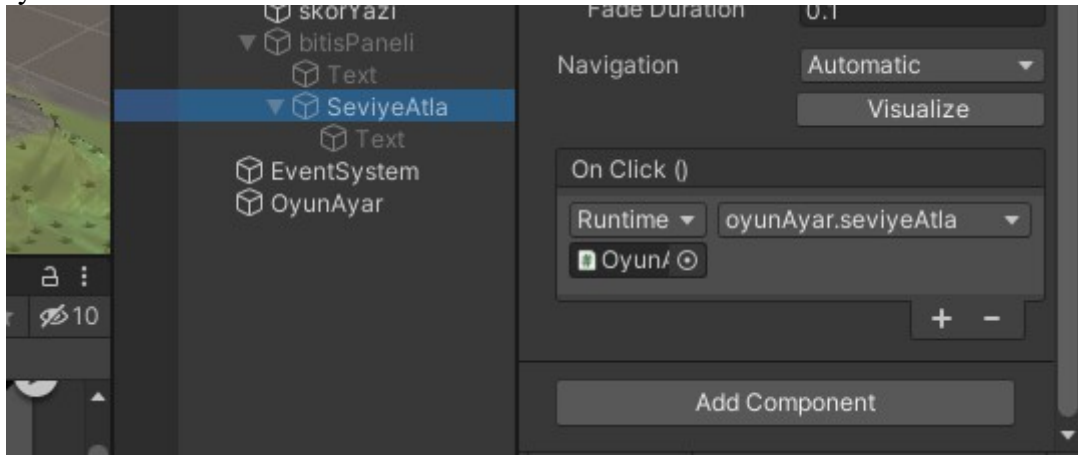




Resim 4.7: Pasif olan bitisPaneli görüntüsü

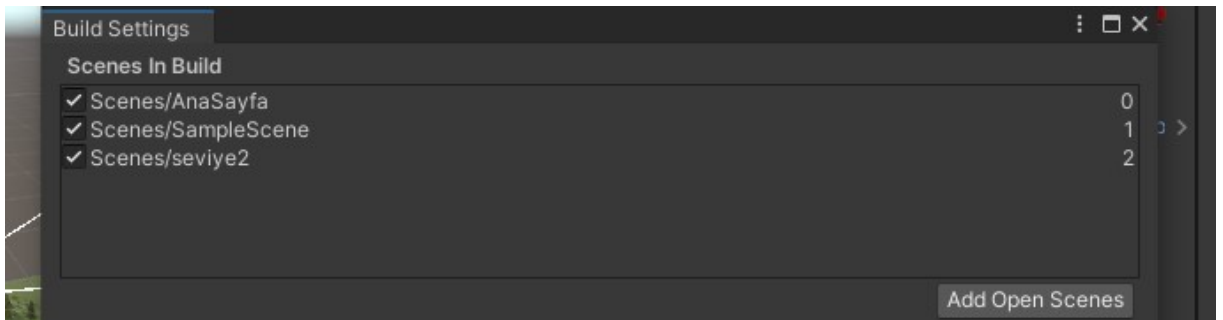
11- Karakterimiz tüm paraları toplayınca, pasif olarak ayarladığımız paneli OyunAyar.cs scriptindeki kod bloğu ile aktif hale getirdik ve bu panel içine seviyeAtla isimli bir buton koyduk.

12- Koyduğumuz bu butonun onclick'ine Oyunayar.cs içinde bulunan seviyeAtla() fonksiyonumuzu tanımladık. Bu sayede butona tıklayınca hangi Scene'ine gideceğimizi belirleyebildik.



Resim 4.8: seviyeAtla butonunun ayar görüntüsü

13- Seviye atlayabilmemiz için, yeni bir Scene oluşturduk. Oluşturduğumuz bu yeni Scene'ine bir Id atanabilmesi için onu, File-> BuildSettings'i açarak "Scene in Build" Paneline sürükledik. Hangi Id'de olmasını istiyorsak o id'de tanımladık.



Resim 4.9: Build Settings panelinin görüntüsü

## 5. KODLAR

### 5.1. OyunAyar.cs'nin Kodları

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; // oyundaki skor yazısını değiştirmek için ekledik....Text
sınıfı çalışsın diye...
using UnityEngine.SceneManagement; // sahne geçişleri için ekledik...

public class oyunAyar : MonoBehaviour
{
    //Skor arttırma-----

    public Text skorYazi; // unity'de OyunAyar'a tıkayınca skorYazi tanımsız
    görünecek.
    // 0 yüzden Hiyerarşi'deki skorYazi'yi, Inspector içindeki skorYazi içine
    sürükleyip atacağız...
    public static int skorSayi; // text formatına dönüştürmeden önce sayının kaç
    olduğunu bilmemiz lazım...

    public int atlanacakSahneID; // bunu public yaparak, OyunAyar kısmından her
    sahnede farklı sayı verebilme imkanı sunduk.
    // Örneğin, unity'de OyunAyar'a tıklayarak, Inspector da atlanacakSahneID kısmına
    Seviye1 de 2; Seviye2'de 3 yazacağız...

    public GameObject panel; // Seviye atlarken gerekli olan panelin aktif edilmesi
    için gerekli...
    //unity de oyunAyar'a tıkla. Inspector'daki panel içine bitisPanelini sürükle
    public GameObject karakter; // Seviye atlarken karakterin pasif edilmesi için
    gerekli...
    //unity de oyunAyar'a tıkla. Inspector'daki karakter içine tHİRDpPERSONCONTROLL'Ü
    SÜRÜKLE
    public int toplanacakPara; // level atlamak için gerekli olan yıldız sayısı...
    //Bunu public yaptık ki, her sahnede unity'de kendimiz değiştirebilelim.
    public int dusunceKactanBaslasin; // yere düşerse yine aynı leveldan başlaması için
    burada public tanımladık.
    // Değerleri unity de vereceğiz.
    //Örneğin Seviye1 de 1, seviye_2'de 2 değerini vereceğiz...

    void Update()
    {
        skorYazi.text = skorSayi.ToString(); // skoru güzel şekilde canvas içine
        yazsın

        //-----Bitiş Panelini aktif etme-----

        //Not: UI-> Panel sekmesinden eklenen paneli pasif yapmak için, GameObject-
        >Toggle Active State'i seçiyoruz.
        if (skorSayi == toplanacakPara) // Örneğin 1. levelda 5 para toplarsa oyun
        atlasın.. toplanacakPara değişkeni her sahnede unity de verilsin...
        {
            panel.SetActive(true); // panel görünsün
            karakter.SetActive(false); // karakter görünmesin
        }
        //-----Bitiş Panelini aktif etme BİTTİ-----
    }
}
```

```

//----- karakter düşünce oyunu yeniden başlatma-----
-----

    if (karakter.transform.position.y < -6.0f) // karakter oyun alanından aşağı
    düşerse
    {
        SceneManager.LoadScene(dusunceKactanBaslasin); // unity de OyurAyar(Bos)
        içindeki dusunceKactanBaslasin değerinde sahneyi yeniden başlatır...
        skorSayi = 0; // skoru sıfırla
    }

    //-----oyunu yeniden başlatma BİTTİ-----

}

public void seviyeAtla()
{
    SceneManager.LoadScene(atlanacakSahneID);

    //Öncelikle scene klasörü içine yeni bir scene oluşturuyoruz.
    //unity de file->build settings'e girip bu scene'i (add open scene) ekliyoruz.
    //Oradaki ID'ye göre (biz yeni scene'i 2 nolu ID'ye getirdik)atama yapıyoruz.
    //unity'de OyunAyar componentine bu scripti sürüklüyoruz. (Yoksa scripti unity
    görmüyor)
    // daha sonra unity de bitisButonuna tıklayıp, Inspector içinde onClick'te +
    işareti basıp OyunAyar component'ini içine sürüklüyoruz.
    //Son olarak da onclick için de function bölümünde bu seviyeAtla
    fonksiyonumuzu seçiyoruz ki, butona tıklandınca hangi fonksiyon devreye girecek buton
    bilsin...

    skorSayi = 0; // yeni level da paralar 0 dan başlasın...
}

}

```

## 5.2. paraKontrol.cs'nin Kodları

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class paraKontrol : MonoBehaviour
{

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag == "karakter") // karakter paraya çarptığında
        {
            Destroy(this.gameObject); // parayı götür. Çünkü bu script para elemanının
            içine atıldı...
        }
    }
}

```

```
        oyunAyar.skorSayi++; // oyunayar.cs içindeki puanı bir arttır.  
        //Destroy(this.gameObject.FindWithTag("yildiz")); //bu da tag'a göre yok  
etme    }  
        }  
  
}
```

## 6. KAYNAKLAR

- 1 - <https://www.btk.gov.tr/haberler/btk-2o2o-dijital-oyunlar-raporunu-yayinladi>
- 2- <https://www.youtube.com/watch?v=DC2B0Cjp1Pk>