



MESLEK YÜKSEKOKULU

BİTİRME PROJESİ İLERLEME RAPORU

ELEKTRONİK VE OTOMASYON BÖLÜMÜ

MOBİL TEKNOLOJİLERİ

BİTİRME PROJESİ

İKİ SAVAŞÇI

HAZIRLAYANIN;

ADI= EMİRCAN

SOYADI= ERGİN

OKUL NO= 18MY93019

DANIŞMAN=OĞR. GOR. NİLGÜN İNCEREİS

MAYIS 2020

İKİ SAVAŞÇI OYUN PROJESİ

BİTİRME PROJESİ

EMİRCAN ERGİN

Tarafından

İSTANBUL OKAN ÜNİVERSİTESİ

MOBİL TEKNOLOJİLERİ BÖLÜMÜNE

ÖNLİSANS

ONAYLAYAN

DANIŞMAN

ÖĞR. GÖR. NİLGÜN İNCEREİS

ÜYE

ÜYE

ÜYE

ÜYE

MAYIS 2020

PROGRAM: MOBİL TEKNOLOJİLERİ

İçindekiler

1. Karakter kodları	6
1.1. Karakter hareket Tanımlamaları	6
2. Karaktere Animasyon Ekleme	9
2.1. Animasyon Ekleme	9
2.2. Animasyon’a İsim Verme	9
2.3. Karakterin görsellerini seçiyoruz.....	10
2.4. Add keyframeye tıklayıp animasyonumuzu kaydediyoruz.	10
2.5. Animasyon Hareket Etme Diyagramı.....	11
3. Kartopu Dağılma Efektı	12
3.1. Dağılma Efektini Ekrana Getirme	12
3.2. Kartopu Değiştirme Efekt Özellikleri	12
3.3. Kartopu Dağılma Efekt Kodları ve Açılımları.....	13
3.4. Yaptığımız efekti script’e (koda) sürükleme	13
4. Kar Yağdırma Efektı	14
5. Giriş Ekranı.....	15
5.1. Giriş Ekranı Kodlaması	16
5.2. Giriş Ekranı OnClick.....	16
6. Karakterlerin Can Kısmı.....	17
6.1. Tanımlamalar	17
6.2. Görseldeki Canların yok olması	17
6.3. GameObject Bildirim Ekranı	18
7. Ses Efektı Nasıl Eklenir.....	18
8. Oyunu Baştan Başlatma	19
9. Giriş Ekranına “ESC” Tuşu ile Geri Dönmek.....	20
10. Yapılan Oyun EXE’ye döndürme	21
11. Oyun İçi Görsel	23
11.1. Tasarım Giriş Ekranı	23
11.2. Tasarım Level 1 Ekranı	23
11.3. Tasarım Level 2 Ekranı	24

12. Oyun Çalışırken	25
12.1. Giriş Ekranı	25
12.2. Level 1 Ekranı	25
12.3. Kazanan oyuncuyu Gösterme Ekranı.....	26
12.4. Level 2 Ekranı	26
12.5. Oyun Hakkında Ekranı	26
13. KAYNAKÇA.....	27

SENARYO

İki arkadaşın beraber eğlenceli vakit geçirecekleri bir masaüstü oyunu yapılması amaçlanmaktadır. İki düşman karakterin bir parkurda karşılıklı ateş ederek birbirlerini öldürmeleri hedeflemektedir. Her iki karakterin 5 canı bulunmaktadır.

NASIL OYNANIR

Birinci Oyuncu w-a-s-d ile karakteri hareket ettirebilecek. "F" tuşu ile karakter ateş edebilecek.

İkinci Oyuncu yön tuşları ile karakteri hareket ettirebilecek. "L" tuşu ile karakter ateş edebilecek.

"R" tuşu ile oynadığınız oyunu tekrar oynayabilirsiniz.

"ESC" tuşu ile giriş ekranına geri dönebilirsiniz.

1. Karakter kodları

1.1. Karakter hareket Tanımlamaları

Görsel 1.1’de görüldüğü gibi hareket ve atlama gücü tanımlandı

```
public float hareketHizi;  
public float atlamaGucu;
```

(GÖRSEL 1.1.)

Görsel 1.2’de görüldüğü gibi karakterin sağ, sol, zıplama ve ateş etme tanımlandı

```
public KeyCode sol;  
public KeyCode sag;  
public KeyCode ziplama;  
public KeyCode atisTopu;
```

(GÖRSEL 1.2.)

Görsel 1.3. görüldüğü gibi Rigidbody2D yani yer çekimi tanımlandı.

```
private Rigidbody2D RB;
```

(GÖRSEL 1.3.)

Görsel 1.4’ de görüldüğü gibi;

Rigidbody2D bileşenini kodun içerisine getirir.

Animator bileşenini kodun içerisine getirir.

```
void Start()  
{  
    //yerçekimi kodu  
    RB = GetComponent<Rigidbody2D>();  
    //animasyon kodu  
    anim = GetComponent<Animator>();  
}
```

(GÖRSEL 1.4.)

Görsel 1.5’de görüldüğü gibi;

- Karakterin zeminde kaymasını engeller.
- Karakter yatay düzlemde sağ ve sola ilerler.
- Karakterin Hızı “0” olduğunda durması

```
isGrounded = Physics2D.OverlapCircle(yerKontrolNoktasi.position, zemKontVaricap, zemin); //karakterin zeminde kaymasını engeller
if (Input.GetKey(sol)) //sol yönünde karakterin ilerlemesi
{
    RB.velocity = new Vector2(-hareketHizi, RB.velocity.y);
}
else if (Input.GetKey(sag)) //sağ yönünde karakterin ilerlemesi
{
    RB.velocity = new Vector2(hareketHizi, RB.velocity.y);
}
else
{
    RB.velocity = new Vector2(0, RB.velocity.y); //Karakterin durması
}
```

(GÖRSEL 1.5.)

- Görsel 1.7’de görüldüğü gibi GetKeyDown’a “W” seçildi .
- Görsel 1.6’da görüldüğü gibi “W” tuşuna bastıysak karakter zıplasin “W” tuşuna basmadıysak zıplamasın.
- Görsel 1.7’de görüldüğü gibi GetKeyDown’a “F” seçildi .
- Görsel 1.6’da görüldüğü gibi “F” tuşuna bastıysak karakter ateş etsin “F” tuşuna basmadıysak ateş etmesin

```
if (Input.GetKeyDown(ziplama) && isGrounded) //karakterin zıplaması
{
    RB.velocity = new Vector2(RB.velocity.x, atlamaGucu); //Karakterin zıplama gücü
}
if (Input.GetKeyDown(atisTopu)) //Karakterin ateş etmesi
{
}
```

(GÖRSEL 1.6.)

Sol	A
Sag	D
Ziplama	W
Atis Topu	F

(GÖRSEL 1.7)

Görsel 1.8’de görüldüğü gibi;

- Karakterin ateş etme ölçeğini, rotasyonunu ve pozisyonunu belirler.
- Karakterin yürümesi için yürüme animasyonun tetikler.

```
//Karakterin ateş etme pozisyon ve rotasyonu
GameObject ballClone = (GameObject)Instantiate(Kartopu, AtisNoktasi.position, AtisNoktasi.rotation);
ballClone.transform.localScale = transform.localScale;
anim.SetTrigger("Yurume");//animasyon yürümesini tetikler
```

(GÖRSEL 1.8.)

Görsel 1.9’ da görüldüğü gibi;

“X” 0’dan küçük ise vector3(-1, 1, 1) konumunun ölçeğini kullanır. Eğer “X” 0’dan büyük ise vector3(1, 1, 1) konumunun ölçeği kullanılır.

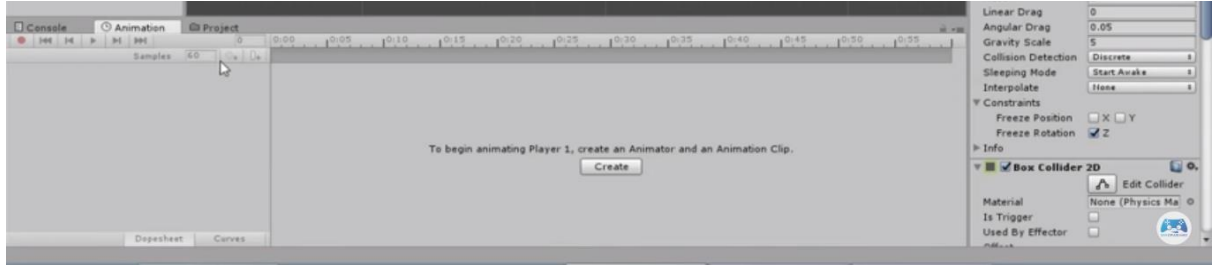
```
if (RB.velocity.x < 0)
{
    // "x" 0dan küçük ise
    transform.localScale = new Vector3(-1, 1, 1);
}else if(RB.velocity.x > 0)
{
    // "x" 0 dan büyük ise
    transform.localScale = new Vector3(1, 1, 1);
}
```

(GÖRSEL 1.9.)

2. Karaktere Animasyon Ekleme

2.1. Animasyon Ekleme

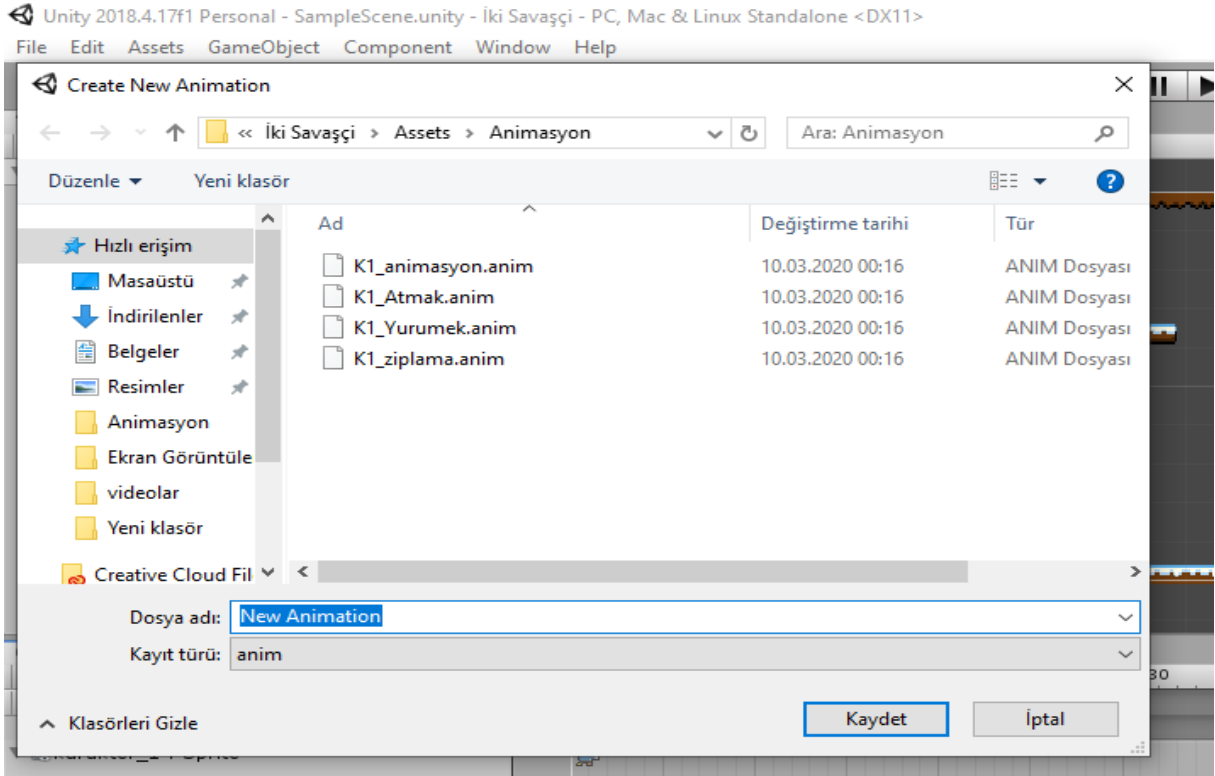
Görsel 2.1’ de görüldüğü gibi karakterin üzerine tıklayıp animation üzerine gelip “Create” diyoruz.



(Görsel2.1)

2.2. Animasyon’a İsim Verme

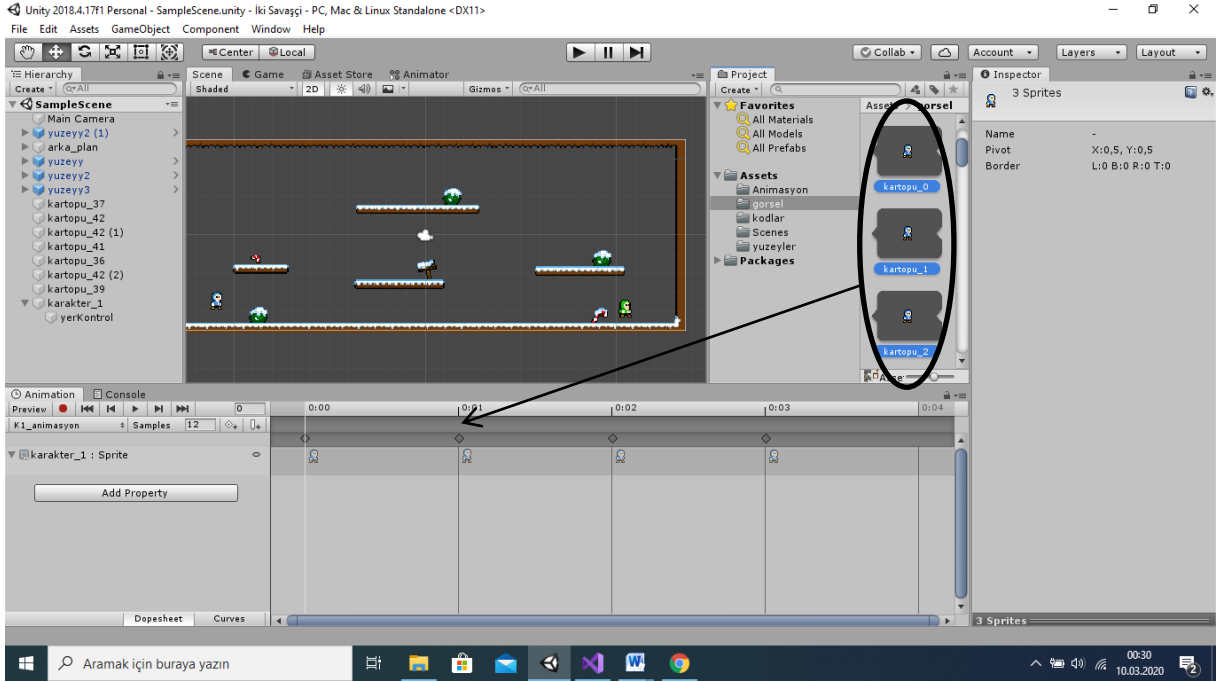
Görsel 2.2 de görüldüğü gibi Animasyonun ismini veriyoruz



(Görsel 2.2)

2.3.Karakterin görsellerini seçiyoruz.

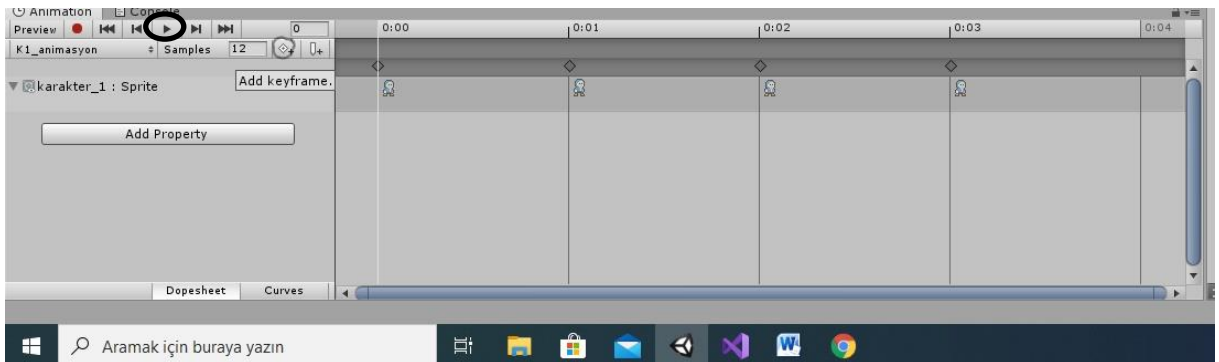
Görsel 2.3’ de görüldü gibi kullanacağımız görselleri seçip sürükleyip bırak yöntemiyle ok ile gösterilen yere bırakılmalıdır.



(Görsel 2.3)

2.4.Add keyframe'e tıklayıp animasyonumuzu kaydediyoruz.

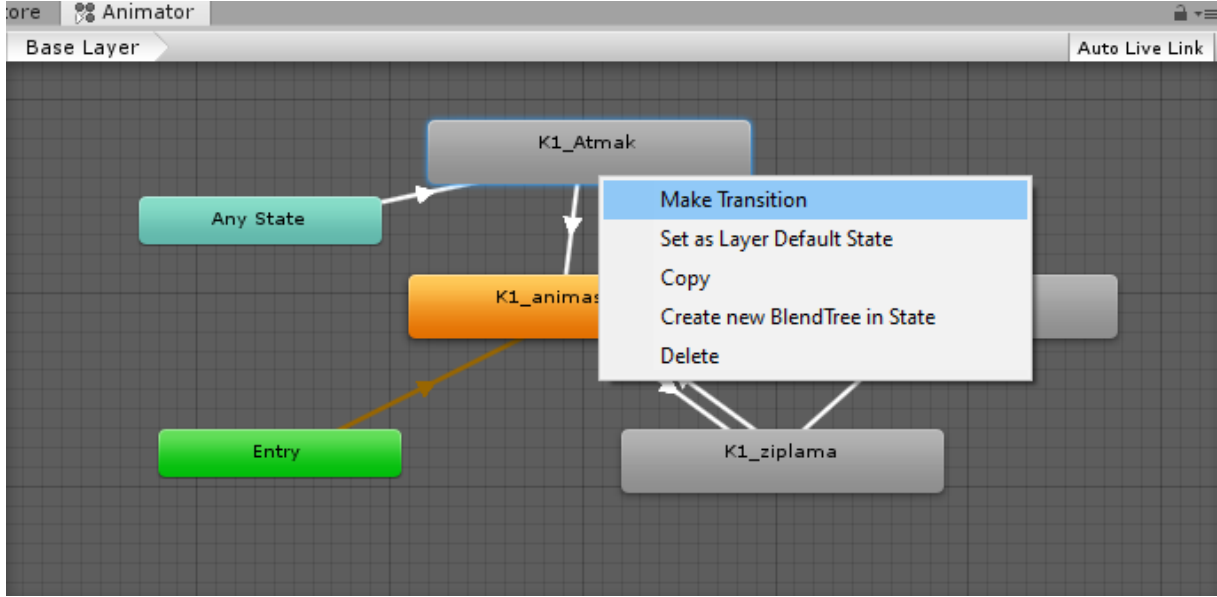
Görsel 2.4’ de görüldüğü gibi Animasyonu başlatıp Add keyframe’e tıklayıp animasyonu kaydediyoruz.



(Görsel 2.4)

2.5. Animasyon Hareket Etme Diyagramı

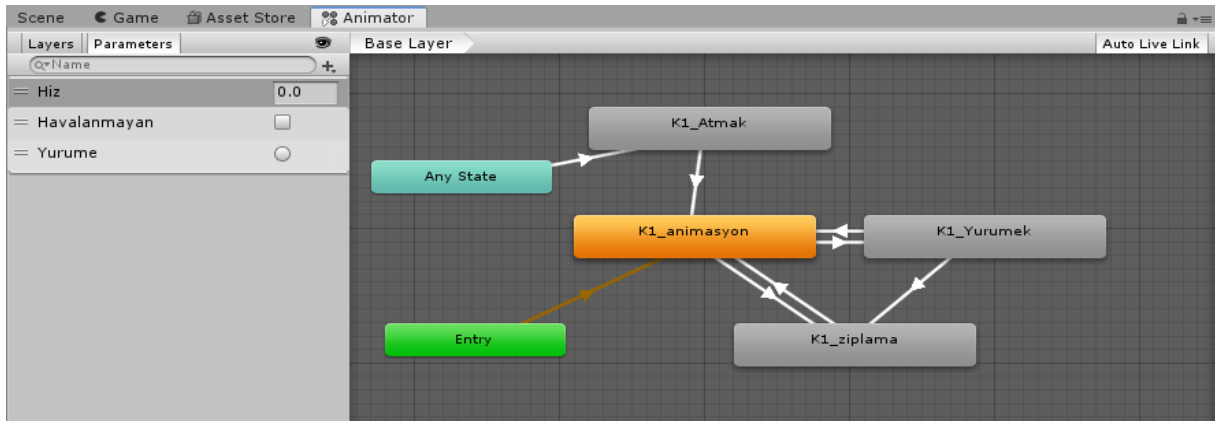
Görsel 2.5'te görüldüğü gibi Animasyonlar arası geçiş yapmak için Make Transition'a tıklayıp ne ile eşleştirilecekse onun üzerine tıklıyoruz.



(Görsel 2.5)

Görsel 2.5.1 görüldüğü gibi animasyon diyagram sıralamaları aşağıdaki gibidir;

- 1) Animasyon -> K1_Yurumek
- 2) K1_Yurumek -> Animasyon
- 3) Animasyon -> K1_ziplama
- 4) K1_Yurumek -> K1_ziplama
- 5) K1_ziplama -> Animasyon
- 6) Any State -> K1_Atmak
- 7) K1_Atmak -> Animasyon

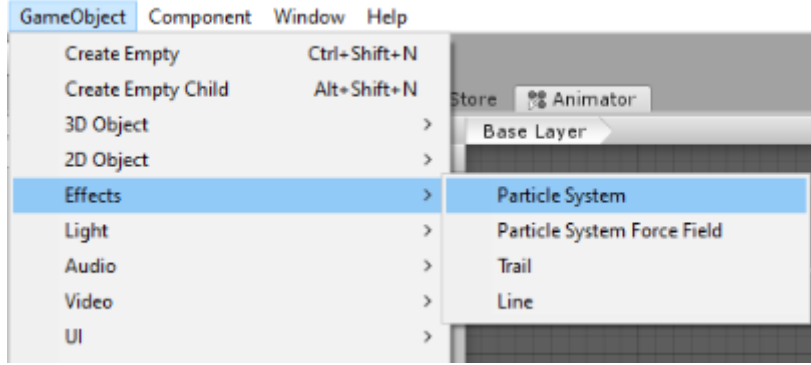


(Görsel 2.5.1)

3. Kartopu Dağılma Efektı

3.1. Dağılma Efektini Ekrana Getirme

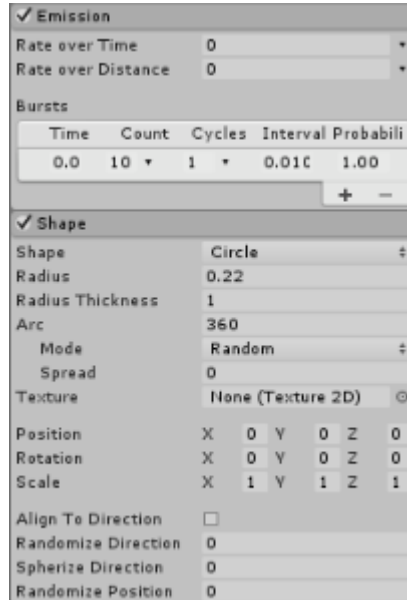
Görsel 3.1’de görüldüğü gibi GameObject -> effects -> Particle System



(Görsel 3.1)

3.2. Kartopu Değiştirme Efekt Özellikleri

Aşağıdaki görsellerde kartopunun karaktere veya duvara çarptığında verilen efektte değişen özellikler kutu içerisine alınmıştır.



3.3. Kartopu Dağılma Efekt Kodları ve Açılımları

```
public class Kartopu : MonoBehaviour{

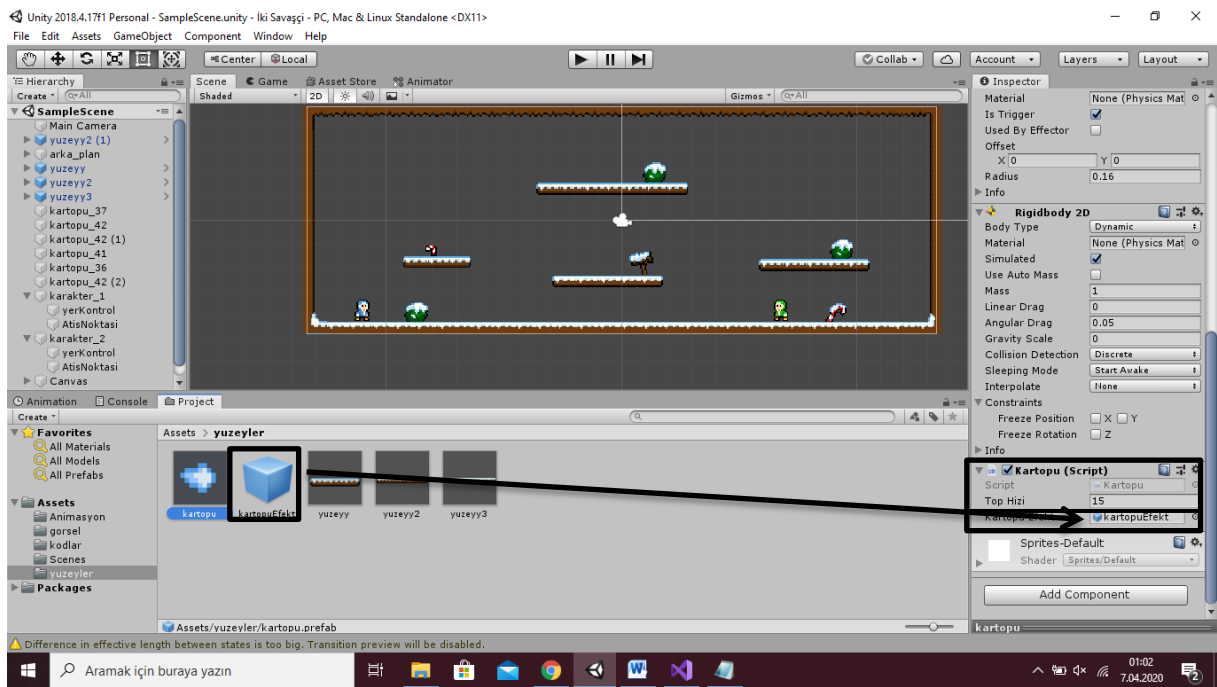
    public GameObject KartopuEfekt;//değişken tanımlama

    Instantiate(KartopuEfekt, transform.position, transform.rotation);//kartopu efekt

}
```

3.4. Yaptığımız efekti script'e (koda) sürüklemeye

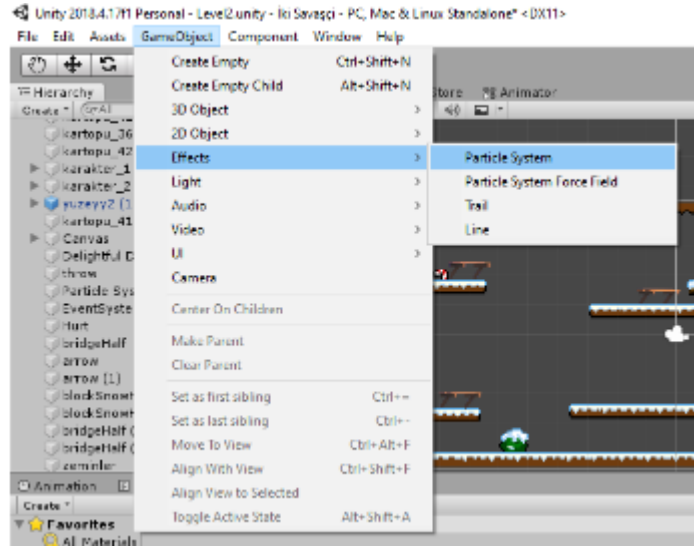
Görsel 3.4'de görüldüğü gibi kartopu gameobject'ini oluşturduğumuz kartopu efektine taşıyoruz.



(Görsel 3.4)

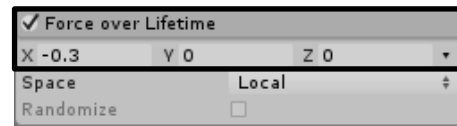
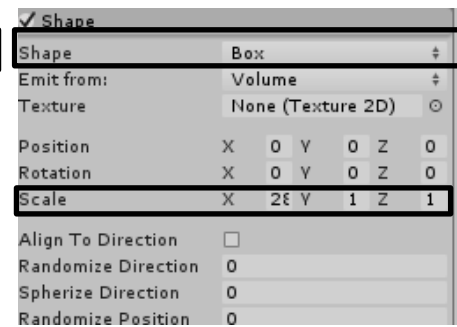
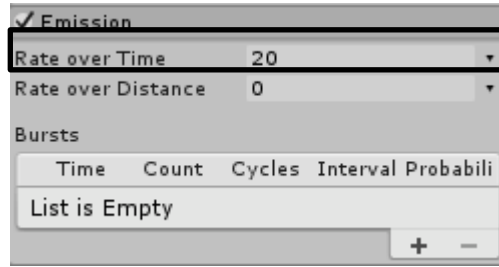
4. Kar Yağdırma Efektı

Görsel 4.1’ de görüldüğü gibi GameObject -> Effects -> Particle System



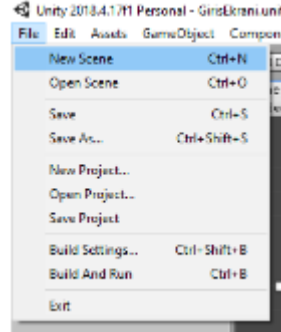
(Görsel 4.1)

Aşağıda kutu içerisine alınmış olanlarda değişiklik yapılmıştır.



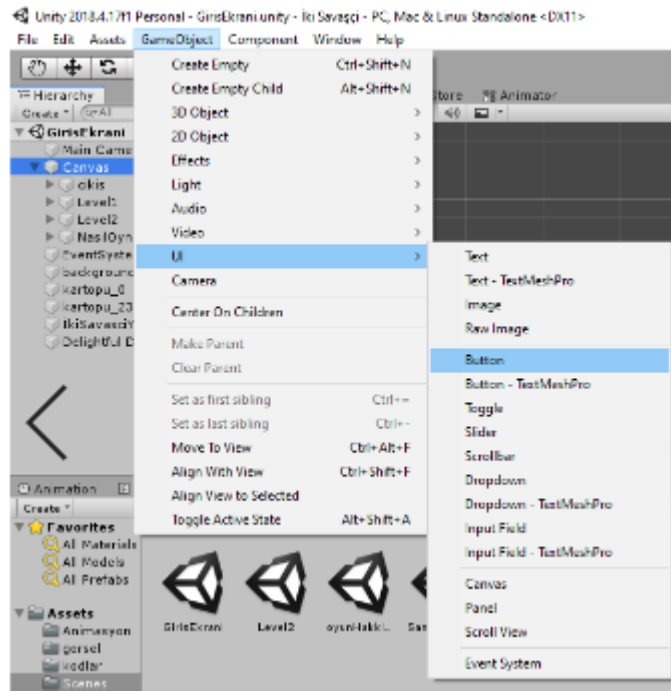
5. Giriş Ekranı

Görsel 5.1.1 'de görüldüğü gibi File -> New Scene



(Görsel 5.1.1)

Görsel 5.1.2' de görüldüğü gibi GameObject ->UI -> Button

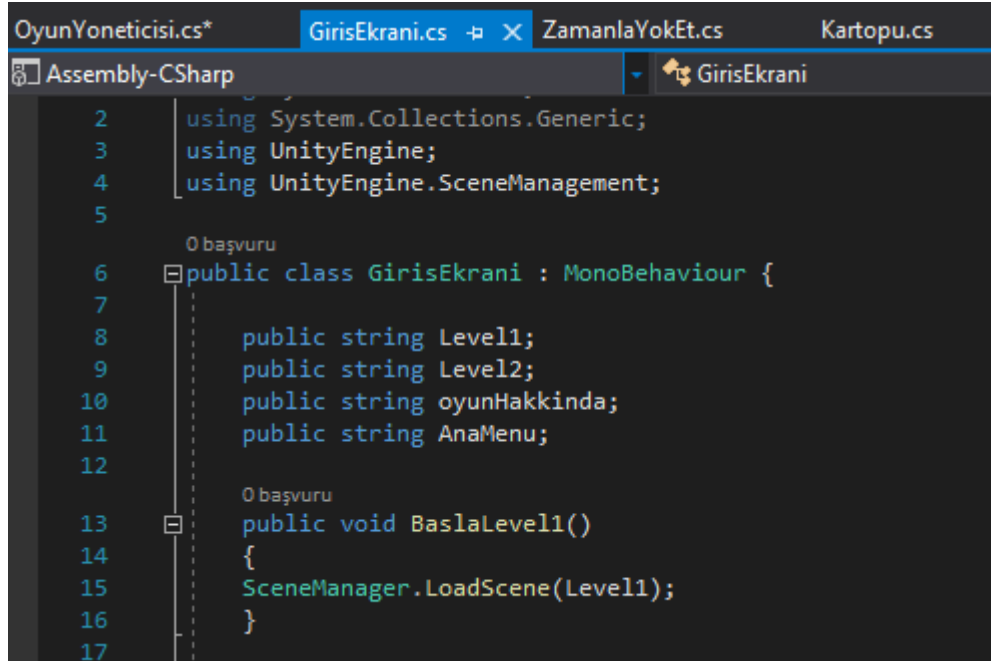


(GÖRSEL 5.1.2)

5.1. Giriş Ekranı Kodlaması

Görsel 5.3’de görüldüğü gibi giriş ekranından gidileceği ekrana ilk olarak Level1 adında bir string tanımlandı.

Gideceği ekranın adı yani string’de tanımladığı adı yazılmalıdır.



```
OyunYoneticisi.cs*  GirişEkranı.cs  ZamanlaYokEt.cs  Kartopu.cs
Assembly-CSharp  GirişEkranı
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  O başvuru
7  public class GirişEkranı : MonoBehaviour {
8      public string Level1;
9      public string Level2;
10     public string oyunHakkinda;
11     public string AnaMenu;
12
13     O başvuru
14     public void BaslaLevel1()
15     {
16         SceneManager.LoadScene(Level1);
17     }
18 }
```

(GÖRSEL 5.1.1.1)

5.2. Giriş Ekranı OnClick

GÖRSEL 5.2.1.1. ‘de görüldüğü gibi Canvas oluşturup GirişEkranı yazılan script(kodlar)’ı canvasın içerisine atılıp On Click + ya tıklanıp None kısmına canvas sürüklenecek. No Function kısmına yazdığımız BaslaLevel1() seçilerek bir sonraki ekrana geçilebilir.



(GÖRSEL 5.2.1.1)

6. Karakterlerin Can Kısımı

6.1. Tanımlamalar

Görsel 6.1.1’de görüldüğü gibi GameObject, int, string ve AudioSource(ses) tanımlandı.

```
2 başvuru
public class OyunYoneticisi : MonoBehaviour
{
    public GameObject oyuncu1;
    public GameObject oyuncu2;

    public int P1Life;
    public int P2Life;

    public GameObject P1Wins;
    public GameObject P2Wins;

    public GameObject[] p1Can;
    public GameObject[] p2Can;

    public AudioSource HurtSes;

    public string GirişEkranı;
```

(GÖRSEL 6.1.1)

6.2. Görseldeki Canların yok olması

Görsel 6.2.1 de görüldüğü gibi Karakterin canı 1’er 1’er azalmaktadır. 1. Oyuncunun canı i’den küçük ise sol üstte ve sağ üstteki can görsellerini eksilt.

```
1 başvuru
public void yaratamak1()
{
    P1Life -= 1; // oyuncu 1'in canı 1 er 1 er azalmaktadır

    for (int i = 0; i < p1Can.Length; i++)
    {
        if (P1Life > i) //1.oyuncunun canı i'den küçük ise
        {
            p1Can[i].SetActive(true); // can görsellerini eksilt
        }
        else
        {
            p1Can[i].SetActive(false); //değil ise can görselini eksiltme
        }
    }
}
```



(GÖRSEL 6.2.1)

6.3. GameObject Bildirim Ekranı

Görsel 6.3.1’de görüldüğü gibi Karakterin canı 0 dan küçük ise 2. Oyuncunun kazandığı gameObject ekranı çıkar.

```
void Update()
{
    if (P1Life <= 0) // 1. oyuncunun canı 0 dan küçük ise
    {
        oyuncu1.SetActive(false); //oyuncu1 objesi devre dışı bırakıldı
        P2Wins.SetActive(true); // 2.oyuncu kazandı objesi etkinleştirildi
    }
}
```

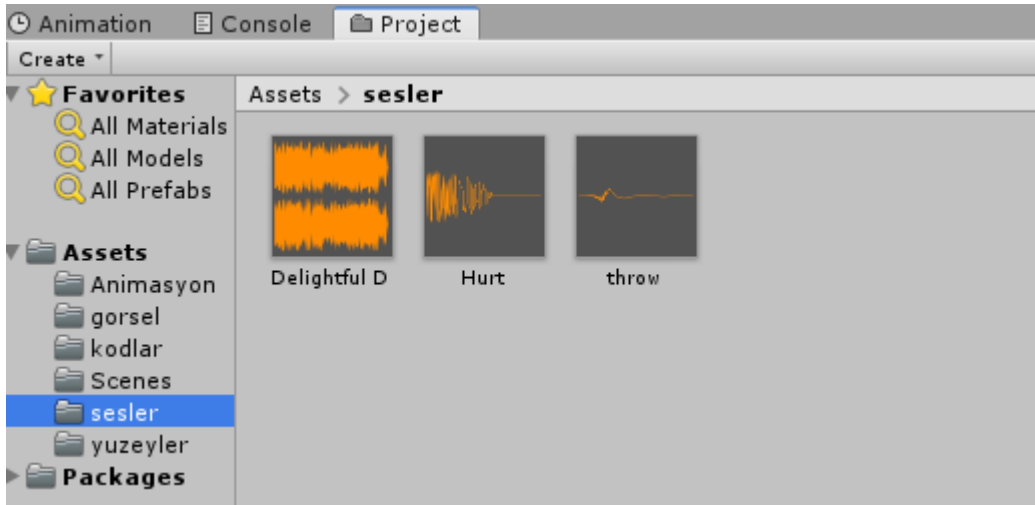
(GÖRSEL 6.3.1)



(GÖRSEL 6.3.2)

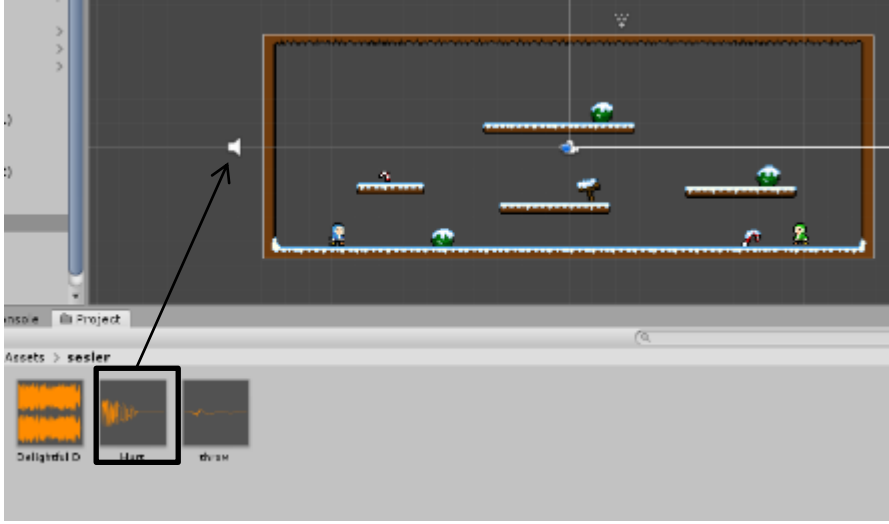
7. Ses Efektı Nasıl Eklenir

Görsel 7.1’de görüldüğü gibi sesler diye bir folder oluşturuldu. Folderin içerisine indirdiğimiz ses dosyaları atıldı.



(GÖRSEL 7.1)

Görsel 7.2’de görüldüğü gibi ses dosyasının taşıyıp projemizin içerisine bırakmalıyız.



(GÖRSEL 7.2)

Görsel 7.3’de görüldüğü gibi projeye atılan sesin çalması için (Görsel)’deki kod yazılmalıdır.

```
}  
HurtSes.Play(); //1.oyuncu için ses kodu  
}
```

(GÖRSEL 7.3)

8. Oyunu Baştan Başlatma

Görsel 8.1’de görüldüğü gibi oyuna baştan başlamak için “R” tuşuna basılır ve oynan ekran tekrar oynatılır.

```
//restart etme kodu  
if (Input.GetKeyDown(KeyCode.R)) //  
{  
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);  
}
```

(GÖRSEL 8.1)

9. Giriş Ekranına “ESC” Tuşu ile Geri Dönmek

Görsel 9.1’de görüldüğü gibi oyunda iken Giriş ekranına geri dönmek için aşağıdaki kod kullanılmaktadır.

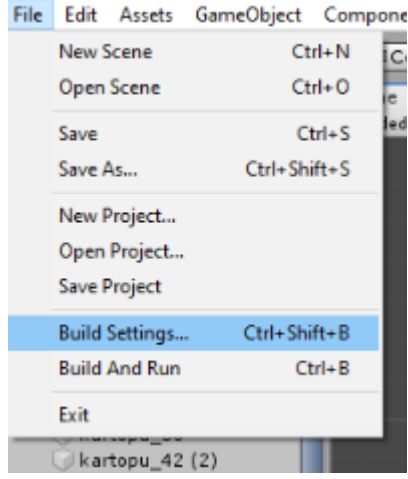
```
}  
// ESC tuşuna basarak giriş ekranına geri dönme kodu  
if (Input.GetKeyDown(KeyCode.Escape))  
{  
    SceneManager.LoadScene(GirisEkranı);  
}
```

(GÖRSEL 9.1)

10. Yapılan Oyun EXE'ye döndürme

Görsel 10.1'de görüldüğü gibi;

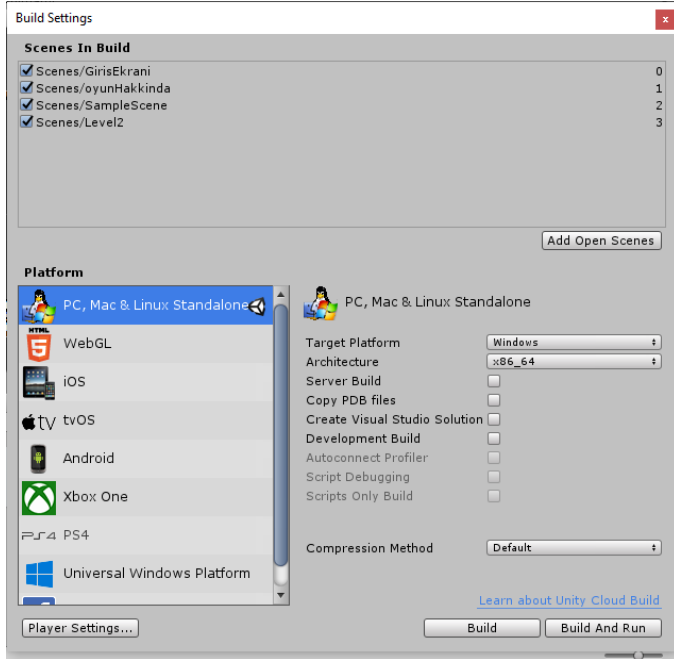
File -> Build Settings



(GÖRSEL 10.1.)

10.2'de görüldüğü gibi;

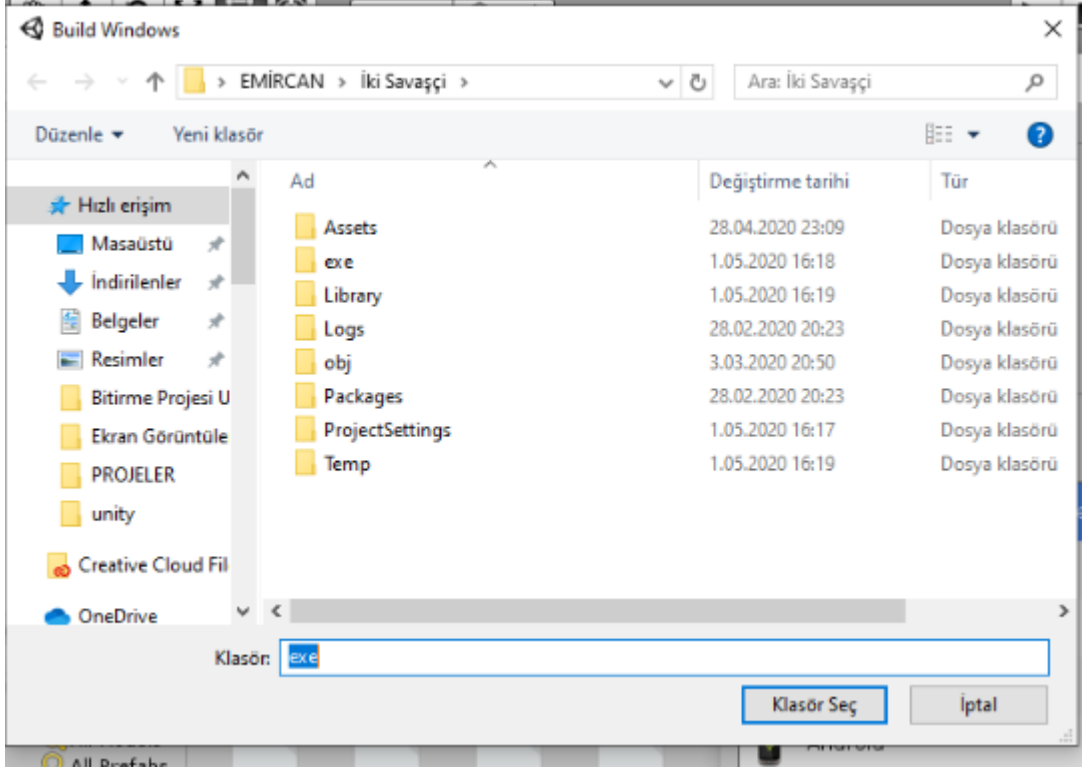
Kutucuk ile işaretli build'de tıklanır.



(GÖRSEL 10.2)

Görsel 10.3’de görüldüğü gibi

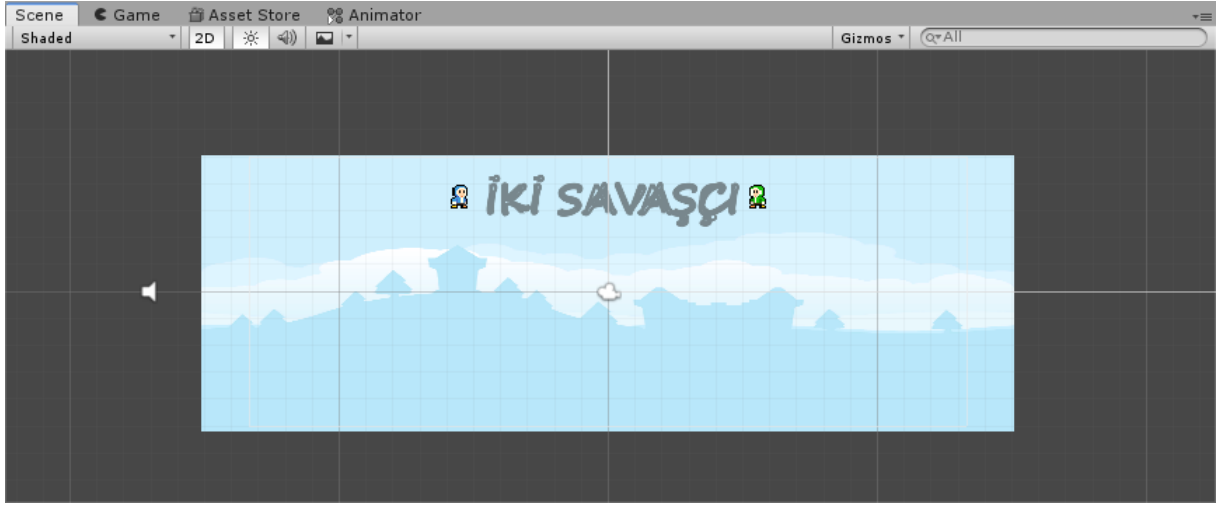
Klasör seçilir ve oyunumuzun exe’si oluşur.



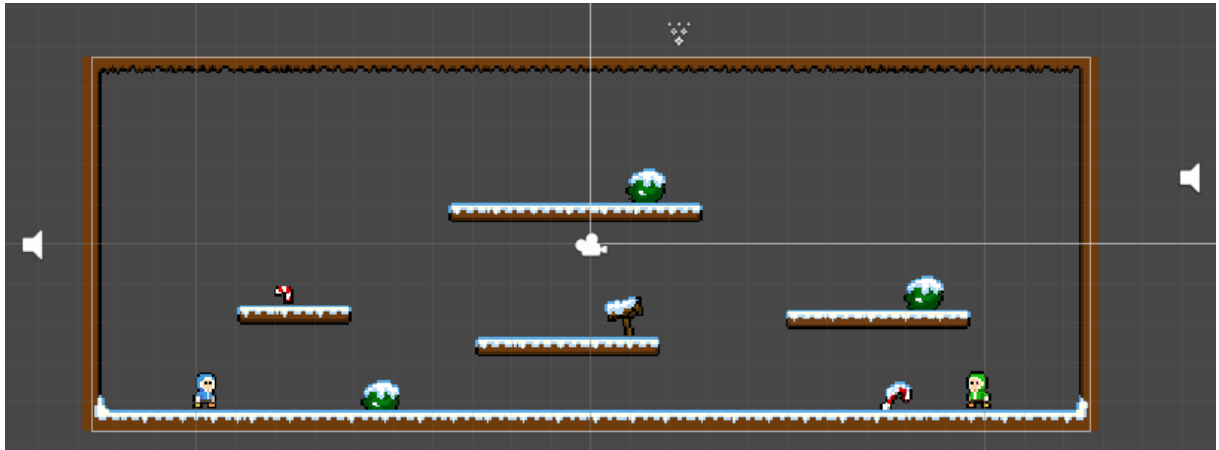
(GÖRSEL 10.3)

11. Oyun İi Grsel

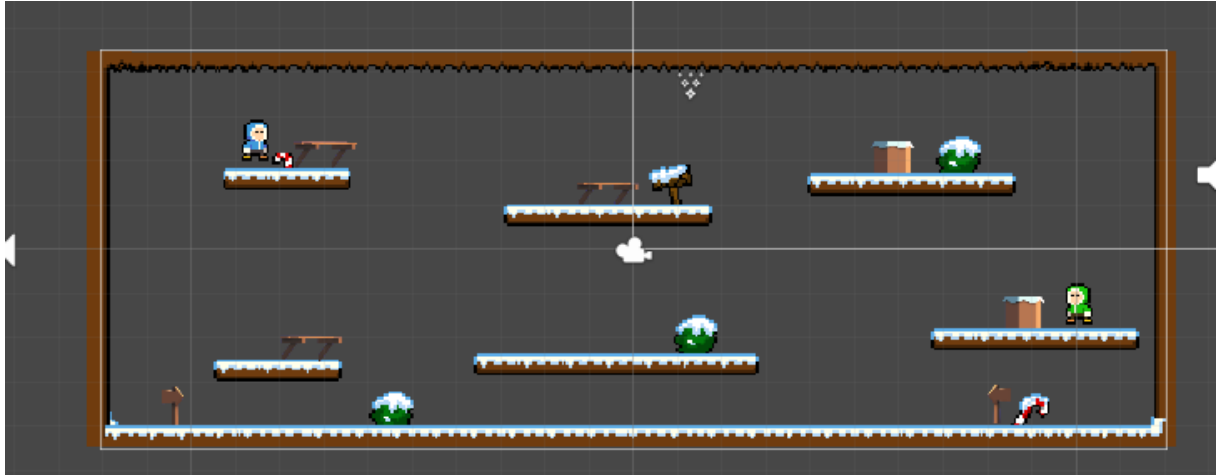
11.1. Tasarım Giriř Ekranı



11.2. Tasarım Level 1 Ekranı



11.3. Tasarım Level 2 Ekranı



12. Oyun Çalışırken

12.1. Giriş Ekranı



12.2. Level 1 Ekranı



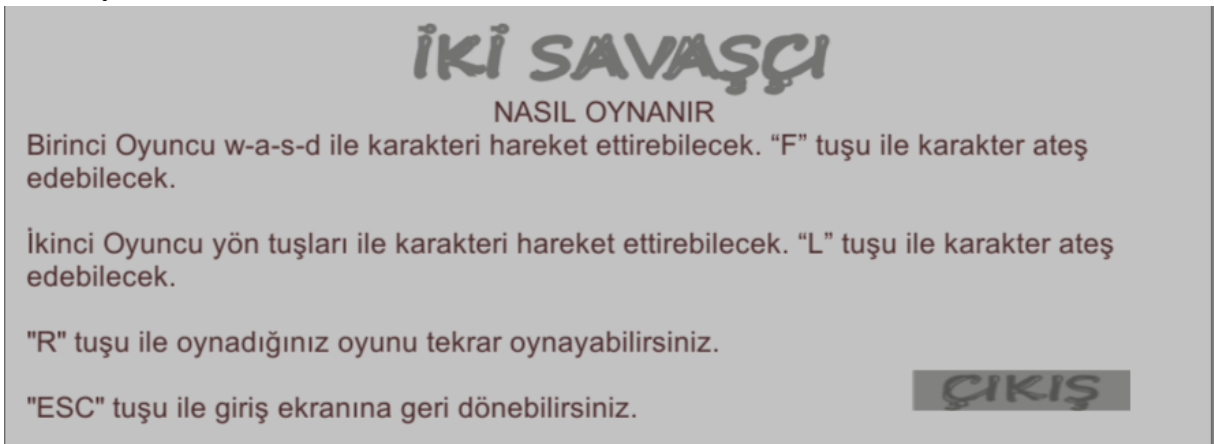
12.3. Kazanan oyuncuyu Gösterme Ekranı



12.4. Level 2 Ekranı



12.5. Oyun Hakkında Ekranı



13. KAYNAKÇA

- [1] <https://docs.unity3d.com/ScriptReference/Physics2D.html>
- [2] <https://docs.unity3d.com/ScriptReference/Physics2D.OverlapCircle.html>
- [3] <https://docs.unity3d.com/ScriptReference/Animator.SetTrigger.html>
- [4] <https://www.youtube.com/watch?v=7nxKAtxGSn8>
- [5] <https://www.youtube.com/watch?v=KQsWKHuKWLM&t=56s>
- [6] https://www.youtube.com/watch?v=RM_7LO03_Og
- [7] <https://docs.unity3d.com/ScriptReference/Transform-localScale.html>
- [8] <https://docs.unity3d.com/ScriptReference/Rigidbody-velocity.html>
- [9] <https://docs.unity3d.com/ScriptReference/GameObject.GetComponent.html>
- [10] <https://docs.unity3d.com/ScriptReference/Object.Instantiate.html>
- [11] <https://docs.unity3d.com/ScriptReference/Array-length.html>
- [12] https://en.wikipedia.org/wiki/Root_of_unity
- [13] <https://docs.unity3d.com/ScriptReference/Time-time.html>