



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
OYUN PROGRAMLAMAYA GİRİŞ ÖDEV RAPORU

İstenen Gereksinimlere Göre Hazırlanan
2D Oyun Tasarımı

G191210088

Bahadır Beşir Kestane

Projenin WebGL Linki

https://play.unity.com/mg/other/webgl_build-178

[Bağlantı](#)

Oyunun Oynarken Çekilmiş Ekran Kaydı Videosu

<https://drive.google.com/drive/folders/1pVTbGXST86ikw9doeAYRXTtQ6-edJhZf?usp=sharing>

[Bağlantı](#)

SAKARYA

Mayıs, 2023

İstenen Gereksinimlere Göre Hazırlanan 2D Oyun Tasarımı

Bahadır Beşir Kestane

G191210088

Özet

Geliştirilen oyun, oyuncu, düşman, çeşitli hareketli nesneler gibi oyun nesneleriyle tasarlanan iki boyutlu bir oyun. Ödevdeki gereksinimlere bağlı kalarak hareketler, animasyonlar vb. özellikler barındıran ve geliştirilen ortam yani Unity platformundaki sunulan çeşitli özellikleri kullanmadan, daha çok kod üzerinden geliştirilen bir oyundur.

© 2023 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Unity, Animasyon, 2D Oyun

1. OYUN

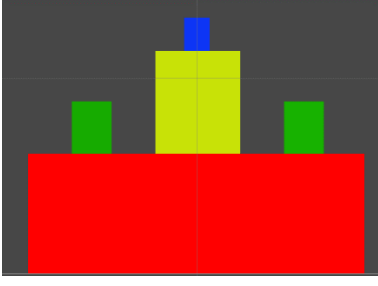
Ödevde istenen gereksinimler şu şekilde özetlenebilir:

- Rastgele, hareketli büyük yuvarlak nesneler üretilecek,
- Üretilen bu nesneler oyuncu tarafından vurulduğu takdirde iki parça halinde bölünüp ters istikamette yön alması ve ebeveyni gibi vurulabilir olması sağlanacak,
- Oyunun içinde bir düşman bulunacak ve bu düşman oyun sahnesini içinde belirli sürelerde sabit yatay ve sabit dikey şekilde hareket edecek ve düşman belirli aralıklarla mermi fırlatacak,
- Sahnenin içinde rastgele aralıklarla altıgen vurulabilir bir cisim belirecek ve vurulduğu takdirde oyuncu tek bir noktadan ateş etmek yerine iki noktadan birden ateş edebilecek,
- Oyuncu veya Düşman tarafından nesneler mermi ile vurulduğunda veya belirli çarpışmalarda, patlama efektine benzer bir animasyon yapılacaktır,
- Oyuncu vurulduğunda parçalara ayrılıp, ayrılan parçaların dönerek yok olması sağlanacak.

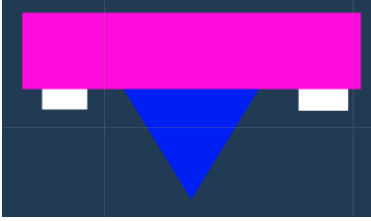
Ödevde istenmeden eklenen özellikler:

- Oyuncunun, vurduğu cisimle beraber iki noktadan ateş edebilme özelliği 5 saniye süreyle kısıtlandı, tekrar vurduğu takdirde özelliği tekrar 5 saniye sürecek ve bu süre ekranda gösterilecek.
- Üretilen yuvarlak nesneler 10 kez vurulmayıp sahneden çıktığında ve düşman ölmediyse düşmanın mermi atış sayısı artırılacak, düşman öldüyse yeni düşman üretilecek.
- Skor : Oyuncu sadece mermi ile vurduğu nesnelerden puan alacak. Çarpışma sırasında yok olan nesnelerden puan alamayacak. Düşman vurulursa 15 puan, büyük bos (sahnenin üstünden gelen büyük yuvarlak nesne) vurulursa 10 puan ve büyük bosun küçük parçaları vurulursa 5 puan alacak. Puan düşüşü olarak sadece düşmanın mermiyle oyuncuyu vurduğunda 10 puanlık bir azalma olacak.

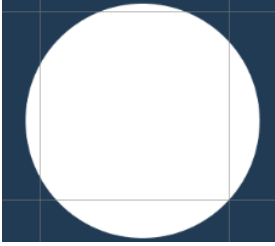
Oyunda Kullanılan Nesneler



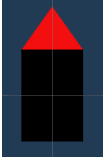
Oyuncu



Düşman



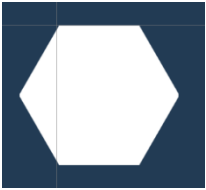
Büyük Bos ve Parçası



Oyuncu Mermisi



Düşman Mermisi



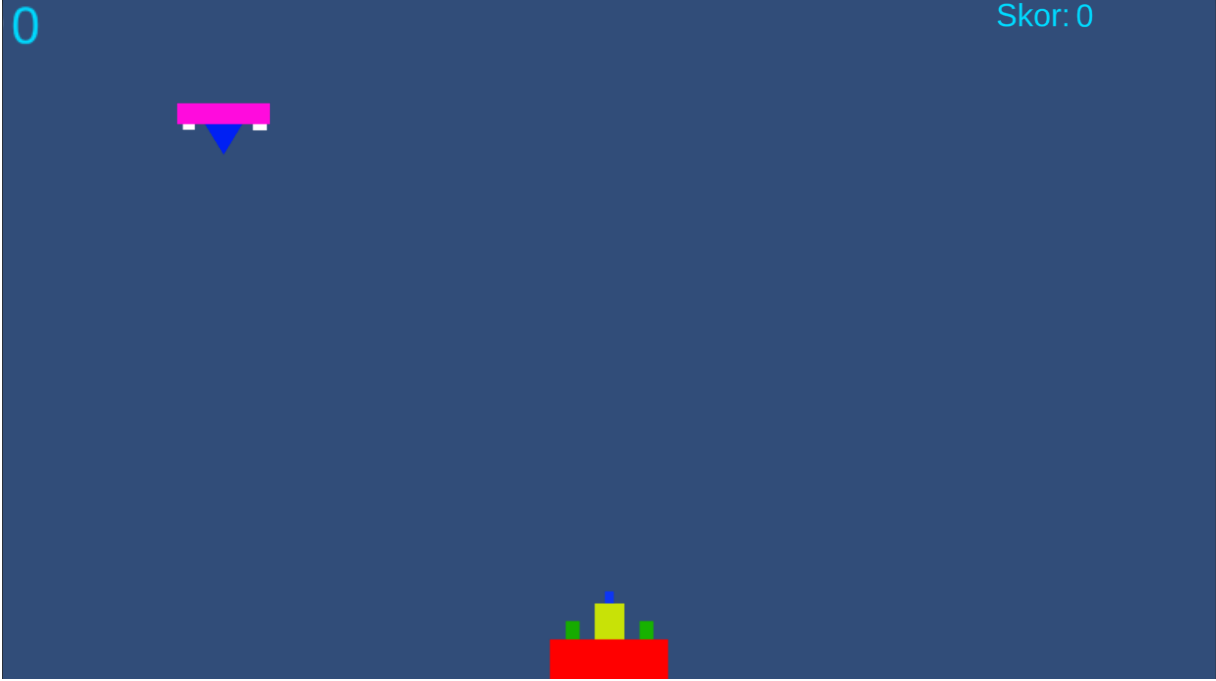
Yükseltme Cismi

Oyun Nasıl Oynanıyor

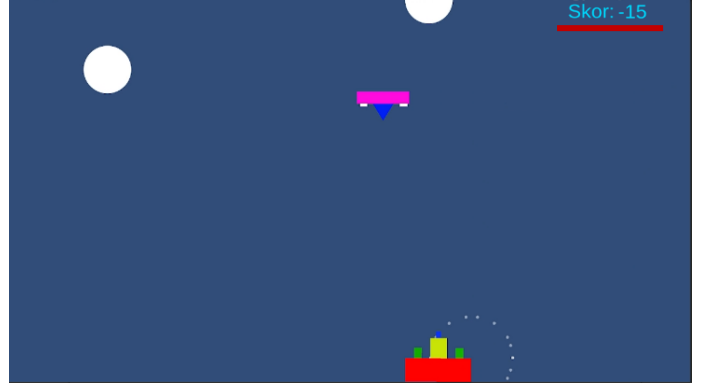
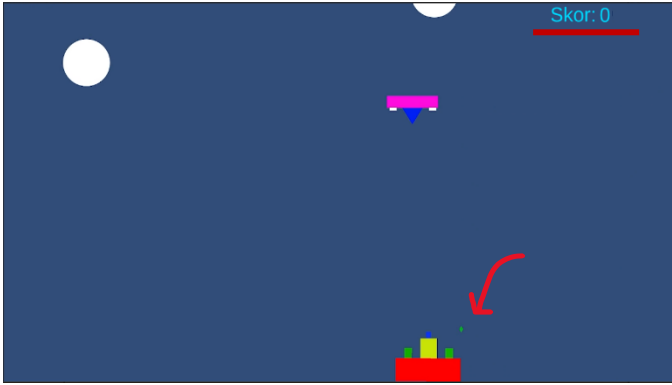
Oyuncu A ve D tuşlarıyla sola ve sağa hareket eder. Yukarı ve aşağı gitme özelliği yoktur. Space-boşluk tuşuna basıldığında mermi fırlatır.

2.ÇIKTILAR

2.1 Oyunun Açılış Ekranı

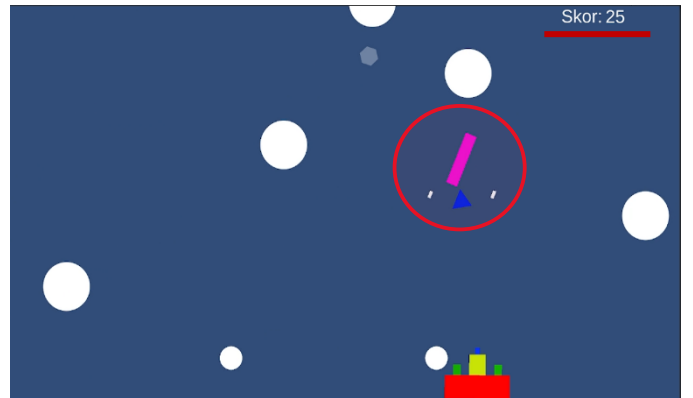
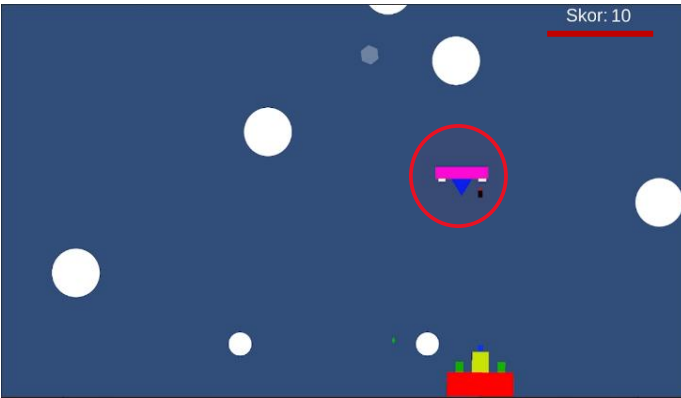


2.2 Düşmanın Saldırı Anı



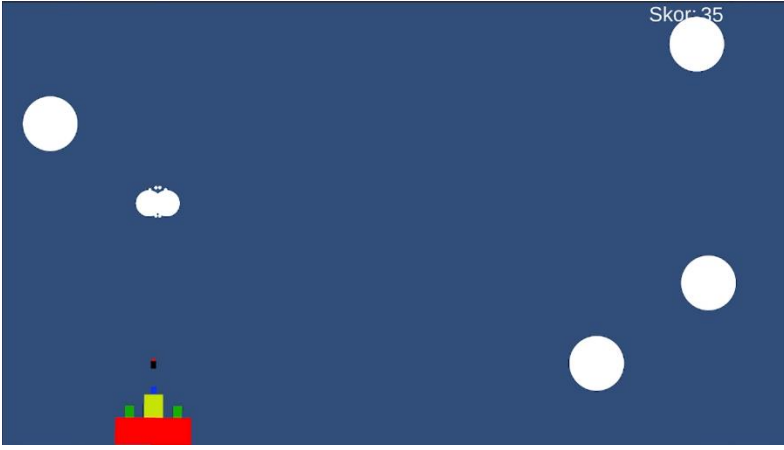
Düşman Oyuncuyu vurduğunda patlama “animasyonu” çıkıyor ve skor 15 puan azalıyor.

2.3 Oyuncunun Saldırı Anı



Oyuncu Düşmanı vurduğunda Düşman parçalara ayrılarak yok oluyor ve oyuncunun skoru 15 puan artıyor.

2.4



“B y k Bos’lar” vurulduėunda ikiye b l n yor ve Patlama animasyonu  ıkıyor.

2.5



“Upgrade” cismi vurulduėunda 5 saniye s reyle iki noktadan atıř imkanı saėlanıyor.

Oyunu Geliştirme Sırasındaki Önemli Noktalar

Patlama animasyonu

```
void patlamaAnimasyon(Vector3 pos)
{
    int parcacikSayisi = 5;
    float[] Xyonleri = new float[] { 1.0f, 0.95f, 0.87f, 0.55f, 0.14f };
    float[] Yyonleri = new float[] { 0.0f, 0.3f, 0.5f, 0.84f, 0.99f };
    float eksen = 1.0f;

    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < parcacikSayisi; j++)
        {
            var yeniPatlama = Instantiate(patlamaSablon);
            yeniPatlama.transform.position = pos;

            yeniPatlama.GetComponent<PatlamaKod>().yonAta(Xyonleri[j] * eksen, Yyonleri[j]);
        }

        eksen = -eksen;
    }

    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < parcacikSayisi; j++)
        {
            var yeniPatlama = Instantiate(patlamaSablon);
            yeniPatlama.transform.position = pos;

            yeniPatlama.GetComponent<PatlamaKod>().yonAta(Xyonleri[j] * eksen, -Yyonleri[j]);
        }

        eksen = -eksen;
    }
}
```

Patlamanın olduğu durumlarda çalışan kod bloğu.

Ayrıca saydamlığı yavaş yavaş artırılarak yok oluyor.

Düşmanın vurulma animasyonu

```
if (dusmanVuruldu == 0)
{
    gameObject.GetComponent<PolygonCollider2D>().enabled = false;

    rb.velocity=new Vector3(0.0f, 0.0f, 0.0f);
    if (saydamlik<=-1.0f)
    {
        Destroy(gameObject);
    }

    //transform.position += hizVektoru * Time.deltaTime;
    dusmanGovde.transform.position += new Vector3(0.0f, 1.2f, 0.0f) * Time.deltaTime;
    dusmanBurun.transform.position += new Vector3(0.0f, -1.2f, 0.0f) * Time.deltaTime;
    silah1.transform.position += new Vector3(-1.0f, -1.0f, 0.0f) * Time.deltaTime;
    silah2.transform.position += new Vector3(1.0f, -1.0f, 0.0f) * Time.deltaTime;

    float donus = 2.5f;
    dusmanGovde.transform.Rotate(0.0f, 0.0f, donus);
    dusmanBurun.transform.Rotate(0.0f, 0.0f, donus);
    silah1.transform.Rotate(0.0f, 0.0f, donus);
    silah2.transform.Rotate(0.0f, 0.0f, donus);

    if (azalmaSiniri >= azalmaSuresi)
    {
        dusmanGovde.GetComponent<SpriteRenderer>().color += saydamlikAzalt;
        dusmanBurun.GetComponent<SpriteRenderer>().color += saydamlikAzalt;
        silah1.GetComponent<SpriteRenderer>().color += saydamlikAzalt;
        silah2.GetComponent<SpriteRenderer>().color += saydamlikAzalt;

        saydamlik -= 0.04f;

        azalmaSiniri = 0.0f;
    }

    azalmaSiniri += Time.deltaTime;
}
```

Düşman parçalara ayrılır ve döner. Saydamlığı sıfırlanınca yok olur.

Kaynakça

- [1] Ders videoları
- [2] [Unity Learn](#)
- [3] [Ogreniyoruz.net](#)