

**HACETTEPE ÜNİVERSİTESİ**

**BİLİŞİM ENSTİTÜSÜ**

**BİLGİSAYAR GRAFİĞİ ANABİLİM DALI**

**BİLGİSAYAR ANİMASYONU ve OYUN BÖLÜMÜ**

**BCO650 Dönem Projesi**

Ragged Road Racing

Oyun Tasarım Belgesi

Prof. Dr. Hakan TÜZÜN

Proje Danışmanı

Murat Yılmaz

2020

# İÇİNDEKİLER

[1. GİRİŞ](#_Toc396861082) 3

[1.1. Oyunun Kısa Özeti ve Konsepti](#_Toc396861082) 3

[1.2. Benzer Oyun Türleri](#_Toc396861082) 3

[2. Oynanış ve Mekanikler 4](#_Toc396861086)

[2.1. Oynanış 4](#_Toc396861087)

[2.2. Mekanikler 4](#_Toc396861090)

[2.2.1. Fizik 4](#_Toc396861091)

[2.3. Oyun Nesneleri](#_Toc396861094) 5

[2.3.1. Benzinlikler](#_Toc396861111) 5

[2.3.2. Kamera Ayarları](#_Toc396861112) 5

[2.4. Taşıtlar](#_Toc396861116) 7

[2.4.1. Kamyonet](#_Toc396861116) 8

[2.4.2. Taksi](#_Toc396861116) 8

[2.5. Yakıt Seviyesi Çubuğu 1](#_Toc396861116)0

[2.6. Tuzaklar 1](#_Toc396861116)1

[2.6.1. Deniz-Göl](#_Toc396861116) 11

3. Materyaller ve Yöntem 20

3.1. Modelleme ………………………………………………………………………………………………………………………………..20

3.2.Işık ve Kamera……………………………………………………………………………………………………………………………..20

3.3.Görsel Efektler…………………………………………………………………………………………………………………………….21

3.4. Ses…………………………………………………………………………………………………………………………………………… 22

3.5. Kodlama…………………………………………………………………………………………………………………………………… 23

[4. İhtiyaç Duyulan Araçlar](#_Toc396861129) 23

1. GİRİŞ

## 1.1. Oyunun Kısa Özeti ve Konsepti

## Ragged Road Racing; Mobil Platform için hedeflenen, izometrik görüş açısına sahip bir yarış oyunudur. Oyun, karşılaşılan yol bozuklukları, eksiklikleri ve yakıt durumuyla mücadele vererek en uzun süre parkurda kalma amacı üzerine kurgulanmıştır. Kalabalık ve yorucu bir görünüm olmaması adına ve mobil cihazların kaynaklarını fazla tüketmemek(dosya boyutu-pil ömrü) için düşük poligonlu tasarıma sahiptir ve animasyon filmlerini anımsatan karikatüristik tarzda modeller kullanılmıştır.

## Tür:Yarış-Aksiyon

## Hedef Kitle: 13+

## 1.2. Benzer Oyun Türleri

2012 Yılında mobil platforma yönelik şekilde piyasaya sürülmüş olan Hill Climb Racing(2D) ile 2014 Yılında yine aynı platformda sunulmuş olan ve aslında C64 Frogger oyununun modern bir türevi olan Crossy Road oyununun birlikte harmanlanmış bir hali olarak da düşünülebilir.

## 1.3. Proje Kapsamı

Projenin ilk adımı olarak oyun fikri ve konsepti üzerinde araştırma yapılmıştır. Fikrin uygulanabilir olup olmadığını tespit etmek için benzer çeşitli oyun türleri incelenmiştir ve eksikliği görülmüştür. Oyun tasarımında kullanılacak modeller, sesler için *Unity Asset Store* ve *SoundCloud üzerinde telifsiz içerik üreten kişilerin* kütüphanelerinin incelemesi yapılmıştır.

Oyun fikri netleştirildikten sonra fokus dokümanı hazırlanmıştır. Bu dokümanda oyununun hikayesi ve oynanışı üzerinde durulmuştur. Fokus dökümanı üzerinde 2’den fazla yorum gelmeyince fikrin güzel olduğu ve radikal değişikliklere gidilmeye gereksinim duyulmadan oyun üzerinde ilk çalışmalar yapılmaya başlanabileceği anlaşılmıştır. Ardından fokus dökümanında düşünülen ögelerle örtüşecek şekilde modeller ve oyun sahnesinde yer alacak objeler Unity Asset Store üzerinden temin edilmiştir. Bu sayede tasarım ile fazla vakit kaybedilmeden oyun prototiplenmeye başlamıştır. Oyun geliştirme aracı olarak Unity Motoru tercih edilmesindeki en önemli etken mobil kullanıcılara yönelik düşük boyutlu export işlemi yapılabiliyor olması ve geniş bir low-polyasset kütüphanesine sahip oluşudur.

# 2. Oynanış ve Mekanikler

## 2.1. Oynanış

### Oyuncu sahne içerisinde yolun farklı yükseklikleri ve meydana gelen boşlukları karşısında aracına yeterli miktarda yakıt gönderecek şekilde hızlanmasını sağlayarak ilerlemeyi hedefler. Yakıt, oyuncunun yapacağı gazlama oranının agresifliğiyle doğru orantılı şekilde azalacağı içinsahne içerisinde belli bölgelerde yakıt istasyonları bulunacaktır. Fakat oyuncunun bu istasyonlarda bir süre beklemesi ve yakıtını o şekilde doldurması beklenmektedir. C64 RiverRaid oyunundaki yakıt ikmal mekaniği gibi düşünülebilir. Dilerse o istasyonda beklemeyebilir ya da deposunu doldurana kadar istasyon içerisinde duruş yapabilir. Bu tarzda bir dolum şekli düşünülmesindeki temel amaç oyuncunun pas geçtiği istasyona gerçek hayatta olduğu gibi yolun yönüne zıt şekilde ilerleyip aracını geri götüremeyeceği için yakıt konusunda kendine has bir strateji belirlemesine olanak sağlamaktır. Herhangi bir hikaye-senaryo barındırmamakta ve önceden tasarlanmış endless run tarzı sahnelerde aracın yakıtı bir şekilde bitinceye kadar bölümler sürmektedir. Oyuncu yolda topladığı altınlarla bir sonraki bölümün kilidini açmış olacaktır. Bölümlerin atmosferi, renkleri tematik olarak oyunun zorluğuna göre değişecektir. Bölümler içerisine ekstra zorluk katabilmesi açısından yola dökülmüş yağ birikintisi vb. zorluk ögeleri eklenebilir.

## 2.2. Mekanikler

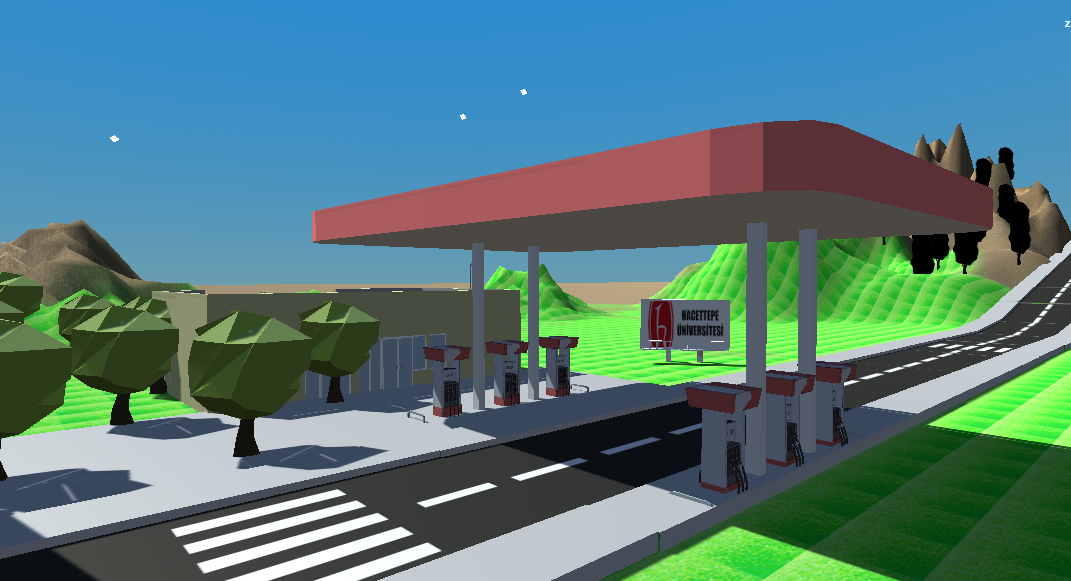
### 2.2.1. Fizik

Objelerin birbiri içerisinden geçmemeleri için Unity oyun motorunda yer alan çarpışma (collision) özellikleri incelendi. Bu doğrultuda sahne içinde kullanılacak hareketsiz nesnelerin kafeslerinin (static mesh) sınırlarına karar verildi. Hareketli bir öge olan kamyonet ile yol, binalar ve gem’lerin çarpışma-etkileşime girme sınırlarını belirleyen kapsüllerin boyutlandırılması da nesnelerle temasları halinde iç içe geçmeyecek şekilde ayarlandı. Günlük yaşantıdan esintiler barındıran, patinaj ve duman efekti eklendi. Çarpışma durumlarında sürekli aynı seslerin çıkmaması için gerekli varyasyonda ses efektleri edinildi.

## 2.3. Oyun Nesneleri

### 2.3.1 Benzinlikler

Oyunda kullandığımız kamyonetin yakıtı zaman içerisinde azalmaktadır. Oyuncunun kamyonetin yakıtı boş seviyeye düşmeden bir akaryakıt istasyonundan geçirmesi gerekmektedir. Yakıt istasyonlarını çerçeveleyen trigger alanları sayesinde bu sağlanmaktadır. Kamyonet box içerisinde kaldığı süre zarfında dolum gerçekleşmektedir.



**2.3.2 Kamera Ayarları**

Oyuncunun oyunu nasıl bir açıdan oynayacağına karar verebildiği bir alandır. Klavyeden C tuşuna basarak interiorcam’e geçiş yapabilir ya da eğer mobildeyse ekran üzerindeki kamera simgesine dokunarak kamera açısını değiştirebilir.

Oyunda otomobilin hareket alanlarının belirlenebilmesi için binaların ve yol kenarı kaldırımların da rigid body’e dönüştürülmesi gerekir.

Oyunda oyuncunun kamyoneti yönlendirebilmesi için (input) tuşlar belirlenmelidir. Burada kamyonetin hareketini sağlayan sağ, sol, ileri yön tuşları tanımlanmalıdır. Ayrıca oyunun temelini oluşturan benzin istasyonlarının da benzin miktarını manipüle etmesine izin verilmelidir.

### 2.4. Taşıtlar

Oyuna kamyonet ile başlanır fakat ilerleyen bölümlerde yeterli gem sayısına ulaşılarak ya da puan satın alınarak yeni modeller de aktif hale getirilebilir.

### 2.4.1 Kamyonet

Yüksek düşüşlerde ve engebeli yollarda rahat hareket eder ve kolay hasar görmez. Fakat bu niteliklerinden dolayı da hızlı ilerleyememektedir.



### 2.4.2 Taksi

Kalkıştan itibaren çok hızlı hareket edebilir, dikkatli kullanılması gerekir. Hızlı olması, beraberinde hassasiyeti de getirmektedir. Yakıt tüketimi de fazladır.



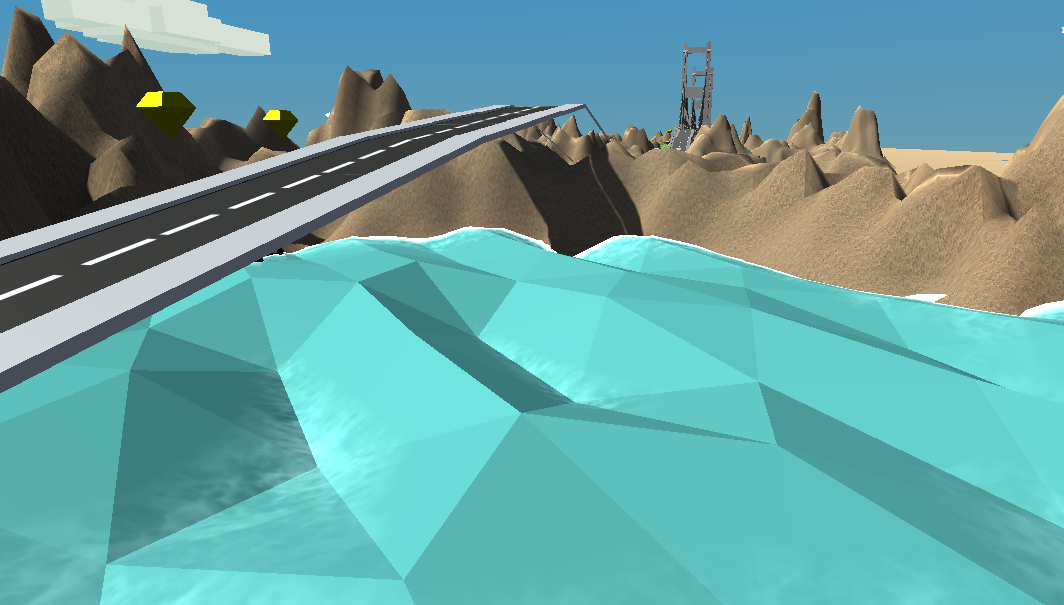
### 2.5. Yakıt Göstergesi

Kamyonette kalan yakıt bilgisini oyuncunun görmesi gerekir. Bunun için ekranın sol üst kısmında görülecek bir durum çubuğu kullanılır. Burada çubuk ekran üzerinde sabit olabileceği gibi, kullanılan taşıtın üzerinde hareketli halde de bulunabilir. Ben ekranın sol üst köşesinde diğer bilgi satırları ile sabit olmasının görsel olarak daha az yorucu olduğunu düşündüğüm için bu şekilde kullanmaya karar verdim. Widget oluşturulup içerisine kanvas eklenir. Hiyerarşide bu kanvasın bir alt seviyesinde olacak şekilde bir fuelbar (yakıt çubuğu) oluşturulur. Yakıt çubuğunun özelliklerini oluşturan veriler girilir. Burada en mühim nokta fonksiyon olarak tanımlanmış olan yüzde değerinin ilerleme çubuğunda seçilmesi ve belli bir değerin altına düştüğünde oyunun durdurulabilmesidir.

### 2.6. Tuzaklar

Oyunda oyuncunun yönettiği taşıtın hareketlerini zorlayan ya da yavaşlatan çeşitli unsurlar bulunur. Bunlara köprüler, tepe geçişleri ve yol boşlukları örnek olarak verilebilir. Oyuncunun hızlı olması gerektiği gibi zaman zaman kontrollü geçişler yaparak bölümü dikkatli bir şekilde tamamlaması da beklenmektedir.

### 2.6.1. Deniz-Göl



Eğer bir şekilde taşıt denize düşerse oyunun bittiğine dair bilgi veren canvas tetiklenir ve bir menü çıkar. Oyuncunun bölüme baştan başlaması gerekir.

# 

# 3. Materyaller ve Yöntem

## 3.1. Modelleme

Oyundaki kamyonet ve taksi için Unity Assetstore üzerinden modeller alındı ve oyunun stiline uygun olacak şekilde detaylandırıldı.

## 3.2. Işık ve Kamera

Unity oyun motorunda bulunan SkyLight ve DirectionalLight nesneleri kullanılarak sahne ışığı ayarlanır. Sahneye ışık kaynakları ile birlikte yansıtıcı objeler de eklenebilir. Kameranın kamyoneti izometrik olarak gösterebilmesi için uygun şekilde konumlandırılması gerekir.

## 3.3. Görsel Efekt



Oyun içerisinde Unity oyun motoru içinde bulunan görsel efektler kullanılmıştır. Bunlar; çeşitli patlama ve duman efektlerinden oluşmaktadır. Kullanılan araç eğer çok yüksekten düşerse ya da şiddetli bir şekilde bir çarpışma yaşarsa camerashake efekti devreye girecek ve oyuncuya çarpışma şiddetini daha bariz bir şekilde yansıtacaktır.

## 3.4. Ses

Sesler için öncelikle bir arşiv oluşturuldu, Müzik, çarpışma, patinaj, lastik kayma sesleri gibi. Oyun içerisinde hangi noktalarda kullanılmasının uygun olacağına karar verildikten sonra arşivlenen ses dosyaları seçilerek uygun uzunluklara getirildi. Bunlar yine tetikleyicilerle ilişkilendirilerek sahne içerisine yerleştirildi.

## 3.5. Kodlama

C# dili kullanılarak otomobilin fiziksel hareketleri, süspansiyonu, lastik dönüşleri ve yönelimleri tasarlanan şekilde kodlandı. UI içerisindeki bilgi ekranlarında da C#’dan faydalanıldı.

Yarış oyunları mobil cihazlarda daima cihaz kaynaklarını batarya ve cihaz kapasitesi yönüyle fazlasıyla tüketen bir tür olmuştur. Ragged Road Racing oyununda ise hedef kullanıcıya sabit izometrik açı ve bununla beraber lowpoly görünümle renkli, eğlenceli bir deneyim sunmaktır. Bu sayede sistem kaynakları da önemli ölçüde az tüketilecektir.

# 

# 4. İhtiyaç Duyulan Araçlar

|  |  |  |
| --- | --- | --- |
| **ARAÇ** | **Kullanım Alanı** | **Yapılan Çalışma** |
| Unity Engine | Oyun Motoru | Level tasarımı, fizik ve oyun mekanikleri oluşturuldu. |
| Unity Asset Store | Modeller | 3 boyutlu modeller edinildi ve oyun temasına göre değişiklikler yapıldı. |
| Microsoft Visual Studio | Kodlama aracı | Oyun kodlama işlemi C# ile gerçekleştirildi. |
| Microsoft Office | Döküman Hazırlanması | Fokus ve oyun tasarım dokümanları yazıldı. |
| Adobe Premiere | Tanıtım Videosu - Kurgu | Oyunun kısa teaser’ı hazırlandı. |