

**T.C.  
SAKARYA ÜNİVERSİTESİ  
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 498 BİTİRME ÇALIŞMASI**

**UNITY İLE 2D OYUN TASARIMI**

**B201210308 – Fırat TURAN**

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ  
Tez Danışmanı : Prof. Dr. CEMİL ÖZ**

**2023-2024 Bahar Dönemi**

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**UNITY İLE 2D OYUN TASARIMI**

**BSM 498 - BİTİRME ÇALIŞMASI**

**Fırat TURAN**

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**

**Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.**

.....  
**Jüri Başkanı**

.....  
**Üye**

.....  
**Üye**

## ÖNSÖZ

Bu tez, bitirme projem olarak geliřtirdiđim "Aksiyon Roguelike Hayatta Kalma Oyunu'nun tasarım ve geliřtirme sürecini detaylandırmaktadır. Bu projeyi gerekleřtirme sürecinde, oyun geliřtirme alanında edindiđim teorik bilgileri pratiđe dökme fırsatı buldum ve bu süreçte karřılařtıđım eřitli zorlukları ařarak deđerli deneyimler kazandım.

Projemi geliřtirirken, Unity oyun motorunu ve eřitli grafik, ses ve kodlama araçlarını kullanarak oyunun tüm bileřenlerini oluřturup entegre ettim. Bu süreçte, oyun tasarımı, seviye tasarımı, karakter tasarımı ve oyun mekaniklerinin uygulanması gibi birçok farklı alanı deneyimleme fırsatım oldu. Ayrıca, oyun test süreçleri ve performans optimizasyonu konularında önemli bilgiler edindim.

Bu tez, oyun geliřtirme sürecini ve projeden elde edilen sonuçları kapsamlı bir řekilde sunarak, gelecekte benzer projeler gerekleřtirecek öđrencilere ve arařtırmacılara yol gösterici olmayı amaçlamaktadır. Bu alıřmanın, oyun geliřtirme alanına olan ilgimi daha da pekiřtirdiđini ve kariyerimde bu alanda ilerlemeye devam etme isteđimi artırdıđını belirtmek isterim.

## İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER .....	iv
ŞEKİLLER LİSTESİ .....	vi
ÖZET.....	vii
BÖLÜM 1. GİRİŞ.....	1
1.1. Proje Tanıtımı.....	1
1.1.1 Oyun türü ve konsepti.....	1
1.1.2. Hedef kitle.....	2
1.2. Projenin Amacı ve Kapsamı.....	3
1.2.1. Ana hedefler .....	3
1.2.2. Kapsam ve sınırlamalar.....	4
BÖLÜM 2. LİTARATÜR TARAMASI.....	6
2.1. Roguelike Oyunların Tarihçesi ve Tanıtımı .....	6
2.2. Aksiyon Roguelike Oyunların Tarihçesi ve Tanıtımı.....	6
2.3. Hayatta Kalma Oyunları .....	7
2.4. Benzer Oyunlar .....	7
BÖLÜM 3. TASARIM VE GELİŞTİRME SÜRECİ .....	9
3.1. Oyun Tasarımı.....	9
3.1.1. Karakter tasarımı.....	9
3.1.2. Düşman tasarımı .....	9
3.1.3. Yetenek tasarımı .....	10
3.1.4. Harita tasarımı.....	11
3.2. Oyun Mekanikleri .....	12
3.2.1. Hareket sistemi.....	12
3.2.2 Hasar sistemi .....	13
3.2.3. Can sistemi.....	13
3.2.4. Toplanabilir objeler .....	14
3.3. Teknik Tasarım.....	14
3.3.1. Kullanılan araçlar ve teknolojiler.....	14
3.3.2. Kodlama yaklaşımları .....	15
3.3.3. Kullanıcı arayüz(UI) tasarımı .....	16

BÖLÜM 4. UYGULAMA VE ENTEGRASYON .....	18
4.1. Oyun Motoru Entegrasyonu.....	18
4.2. Grafik ve Ses Entegrasyonu.....	18
4.3. Kullanıcı Arayüzü (UI) Entegrasyonu .....	19
BÖLÜM 5. TEST VE OPTİMİZASYON.....	20
5.1. Oyun Test Süreçleri.....	20
5.2. Hata Ayıklama ve Düzeltme.....	20
5.3. Performans Optimizasyonu.....	21
5.3.1 FPS artırma teknikleri .....	21
5.3.2 Bellek yönetimi.....	21
BÖLÜM 6. SONUÇLAR VE ÖNERİLER.....	22
6.1. Sonuçlar .....	22
6.1.1. Başarıyla tamamlanan özellikler.....	22
6.1.2. Teknik zorluklar .....	22
6.1.3. Zaman yönetimi .....	22
6.1.4. Öğrenilen dersler.....	22
ÖZ GEÇMİŞ .....	23

## ŞEKİLLER LİSTESİ

Şekil 2.1 Vampire Survivors. ....	7
Şekil 2.2 Brotato.....	8
Şekil 3.2 Düşman Görselleri .....	9
Şekil 3.3 Yetenek Görselleri .....	10
Şekil 3.4 Zemin Görselleri .....	11
Şekil 3.5 Hareketli Zemin Görselleri .....	11
Şekil 3.6 Harita Tasarımı .....	12
Şekil 3.7 Yetenek Görselleri .....	13
Şekil 3.8 Can Görselleri .....	13
Şekil 3.9 Toplanabilir Obje .....	14
Şekil 3.10 Hasar Geliştirme Ekranı .....	16
Şekil 3.11 Yetenek Geliştirme Ekranı .....	16
Şekil 3.12 HUD .....	17
Şekil 3.13 Tur Göstergesi .....	17
Şekil 3.14 Akış Diagramı .....	17

## ÖZET

Anahtar Kelimeler: Unity, 2D oyun geliştirme, aksiyon roguelike, oyun mekanikleri

Bu tez, aksiyon roguelike hayatta kalma türünde bir 2D oyunun geliştirilmesini ele almaktadır. Projenin amacı, Unity oyun motoru kullanarak oyunculara sürükleyici ve zorlu bir oyun deneyimi sunmaktır. Oyunun tasarımı ve geliştirme süreci, karakter ve seviye tasarımı, oyun mekanikleri, performans optimizasyonu ve test aşamalarını kapsamaktadır.

Oyun tasarımı sürecinde, karakterlerin yetenek ve özellikleri belirlenmiştir. Seviye sistemi, zorluk seviyelerinin dengelenmesi ve hasar sisteminin dengelenmesi konularını içermektedir.

Teknik tasarım aşamasında, GIMP kullanılarak grafikler düzenlenmiş, GitHub ile proje yönetimi ve versiyon kontrolü sağlanmıştır. Kodlama yaklaşımları olarak modüler ve nesne yönelimli programlama prensipleri benimsenmiştir. Test ve optimizasyon süreçleri, performans sorunlarının çözülmesi ve bellek yönetiminin iyileştirilmesini içermektedir. Test senaryoları oluşturulmuş ve analiz edilmiştir.

# BÖLÜM 1. GİRİŞ

Bu projede Unity ile 2D bir oyun yapılacaktır. Projenin asıl amacı oyun mekaniklerinin nasıl yapıldığını öğrenmek ve bunu tecrübe etmektir.

## 1.1. Proje Tanıtımı

Oyun belirli bir alanda rastgele oluşan düşmanlarla savaştığımız bir aksiyon roguelike hayatta kalma oyunudur. Oyunda üzerime gelen düşmanları öldürüp deneyim puanı ile karakterimizin yetenekleri geliştirdiğimiz bir yapıya sahiptir.

### 1.1.1 Oyun türü ve konsepti

Oyun aksiyon roguelike hayatta kalma türlerini barındıran bir oyundur. Bizi öldürmeye çalışan düşmanları öldürüp yeteneklerimizi geliştirdiğimiz bir yapıya sahiptir.

Oyunda 3 farklı Fire, Nature ve Water olmak üzere üç adet saldırı türü bulunmaktadır. Her bölümün sonunda Fire, Nature ve Water saldırıları türlerinin hasarını geliştirmekteyiz. Seviye atladığımızda Fire, Nature ve Water yeteneklerimizin farklı özelliklerini(hasar alanı, dönüş hızı, vb.) geliştirmekteyiz. Düşmanlar üç farklı saldırıya karşı FireResist, NatureResist, WaterResist olmak üzere üç farklı savunma mekaniği bulunmaktadır. Oyun turlardan oluşmaktadır. Her tur düşmanlar rasgele yerde oluşmakta ve oyuncuya doğru ilerlemektedir. Düşmanların Resistler'i rasgele bir şekilde oluşmaktadır. Fire, Nature ve Water saldırısına dayanıklı düşmanlar oluşturulmaktadır. Bölüm sonunda hayatta kalan düşmanların bir sonraki nesile genlerini aktarabilmektedir. Örneğin Water gücünü geliştirdik ve WaterResist'i yüksek olan düşmanlar bölüm içinde hayatta kalma şansı daha çok olacaktır, tur sonunda hayatta kalırlarsa bir sonraki neslin WaterResist değerinin yüksek gelme ihtimali artacaktır. Renklerine göre düşmanların hangi Resist'e bağışıklığı olduğu anlaşılacaktır.



Projenin geliştirilmesindeki adımlar şöyle belirlenmiştir; Oyun mekaniklerinin belirlenmesi, düşmanların ve düşman yapay zekalarının tasarlanması, kullanıcı arayüzü tasarımının yapılması, düşmanlar için animasyonların eklenmesi, optimizasyon yapılması ve oyun testlerinin yapılması ile birlikte projeyi son haline getirmek.

### 1.1.2. Hedef kitle

Bu aksiyon roguelike hayatta kalma oyunu, geniş bir hedef kitleye hitap etmeyi amaçlamaktadır. Hedef kitle, aşağıdaki demografik ve psikografik özelliklere sahiptir:

Demografik Özellikler:

- Yaş Aralığı: 14-35 yaş arası
- Cinsiyet: Erkek ve kadın oyuncular
- Eğitim Düzeyi: Lise ve üzeri eğitim seviyesine sahip kişiler
- Gelir Düzeyi: Düşük ve orta gelir grubuna ait bireyler

Psikografik Özellikler:

- Oyun Tercihleri: Aksiyon, macera ve strateji oyunlarına ilgi duyan oyuncular. Özellikle roguelike ve hayatta kalma oyunlarını tercih edenler.
- Oyun Tarzı: Zorlu oyunları seven ve meydan okumaları kabul eden oyuncular. Yüksek tekrar oynanabilirlik oranı olan oyunlara ilgi duyanlar.
- Zaman Kullanımı: Boş zamanlarında oyun oynamayı seven, özellikle hafta sonları ve akşam saatlerinde aktif olan oyuncular.

İlgi Alanları ve Davranışlar:

- İlgi Alanları: Aksiyon ve macera filmleri, strateji ve zeka oyunları, e-spor ve oyun toplulukları.
- Davranışlar: Yeni çıkan oyunları deneme isteği, oyun içi başarımlar ve ödüller için uğraşma, çevrimiçi oyun topluluklarına katılma ve oyun hakkında geri bildirimde bulunma.

Bu hedef kitle, oyun deneyiminden yüksek beklentilere sahiptir ve oyun dünyasında yeni ve zorlu deneyimler aramaktadır. Oyun, bu beklentileri karşılamak amacıyla yüksek kaliteli grafikler, akıcı ve dinamik bir oyun mekaniği, derinlemesine stratejik unsurlar sunmayı hedeflemektedir. Böylece, hedef kitleye hitap eden etkileyici bir oyun deneyimi sağlanacaktır.

## **1.2. Projenin Amacı ve Kapsamı**

Bu çalışmanın amacı, 2D oyun geliştirme sürecini incelemek ve başarılı bir oyun geliştirmek için gereken adımları detaylandırmaktır. Projenin kapsamı, oyun tasarımından yazılım geliştirmeye, grafik ve ses entegrasyonundan oyun testine kadar geniş bir yelpazeyi kapsar. Oyun performansı ve kullanıcı deneyimi ön planda tutulmuştur. Projenin sınırlamaları arasında belirli bütçe ve zaman kısıtlamaları yer almaktadır.

### **1.2.1. Ana hedefler**

Bu aksiyon roguelike hayatta kalma oyununun geliştirilmesindeki ana hedefler, oyunun hem teknik hem de tasarım açısından belirli kriterlere ulaşmasını sağlamaktır. Bu hedefler, oyunun başarısını ve oyuncu memnuniyetini artırmaya yöneliktir. Ana hedefler aşağıdaki gibi belirlenmiştir:

Yüksek Kaliteli Oyun Deneyimi Sunmak:

- Oynanabilirlik: Oyunculara akıcı, dinamik ve keyifli bir oyun deneyimi sunmak. Oyun mekaniğinin sorunsuz ve kesintisiz çalışmasını sağlamak.
- Grafikler: Kaliteli grafikler ile oyuncunun oyuna tam anlamıyla dalmasını sağlamak. Görsel unsurların atmosferi desteklemesi.

Stratejik ve Derinlemesine Oyun Mekaniği Geliştirmek:

- Karakter Gelişimi: Oyuncuların karakterlerini farklı yetenekler geliştirmelerine olanak tanımak. Oyunculara kendi oyun stillerini oluşturma özgürlüğü sağlamak.

- **Kaynak Yönetimi:** Oyuncuların kaynaklarını dikkatli bir şekilde yönetmeleri ve stratejik kararlar almaları gereken bir oyun sistemi kurmak.

**Geniş Bir Hedef Kitleye Hitap Etmek:**

- **Erişilebilirlik:** Oyunun geniş bir oyuncu kitlesi tarafından oynanabilir olmasını sağlamak için kolay anlaşılır bir kullanıcı arayüzü ve kontrollere sahip olmak.

### **1.2.2. Kapsam ve sınırlamalar**

Bu aksiyon roguelike hayatta kalma oyunu projesi, belirli bir kapsam dahilinde geliştirilmiştir ve bazı sınırlamalara tabidir. Projenin kapsamı ve sınırlamaları aşağıdaki gibi belirlenmiştir:

**Oyun Mekanikleri:**

- **Temel Oyun Döngüsü:** Oyunun temel oyun döngüsü, düşmanlarla savaşma, yetenek geliştirme ve hayatta kalma üzerine kuruludur.
- **Roguelike Elementler:** Kalıcı ölüm mekaniği ve her oynanışta farklı deneyimler sunan bir yapı.
- **Karakter Gelişimi:** Oyuncuların karakterlerini geliştirmelerine olanak tanıyan seviye atlama, yetenek ağacı ve çeşitli güçlendirmeler sistemi.

**Grafik ve Ses:**

- **2D Grafikler:** Yüksek kaliteli 2D sprite'lar ve animasyonlar.
- **Ses Tasarımı:** Atmosferik müzik ile oyun deneyimini zenginleştirme.

**Kullanıcı Arayüzü:**

- **Menü Sistemleri:** Hasar geliştirme, yetenek geliştirme menüsü ve oyun içi kullanıcı arayüzü.
- **HUD:** Oyuncunun sağlık durumu, seviye ve tur gibi bilgileri gösteren bir HUD.

**Platformlar:**

- **PC:** Oyun, öncelikle PC platformunda çalışacak şekilde geliştirilmiştir.

#### Geliştirme Süresi ve Kaynaklar:

- Zaman Kısıtlaması: Proje, belirli bir süre içinde tamamlanmak zorundadır, bu da bazı özelliklerin sınırlı kalmasına neden olabilir.
- Ekibin Boyutu: Proje ekibi sınırlı sayıda kişiden oluşmaktadır, bu da iş yükünün sınırlı kaynaklarla yönetilmesini gerektirir.

#### Teknik Sınırlamalar:

- Yapay Zeka: Düşman yapay zekası (AI) belirli bir karmaşıklık düzeyinde tutulmuştur. Daha karmaşık AI davranışları zaman ve kaynak kısıtlamaları nedeniyle sınırlandırılmıştır.

#### İçerik Kapsamı:

- Eşya ve Yetenek Çeşitliliği: Başlangıçta sınırlı sayıda eşya ve yetenek mevcut olacaktır.

#### Çok Oyunculu Mod:

- Tek Oyunculu Odak: İlk sürümde sadece tek oyunculu mod bulunacaktır. Çok oyunculu mod, gelecekteki güncellemeler için planlanmış ancak geliştirme süresi içinde uygulanmamıştır.

Bu kapsam ve sınırlamalar, projenin gerçekçi bir şekilde yönetilmesini sağlamak amacıyla belirlenmiştir. Belirlenen kapsam dahilinde en yüksek kalitede ve oynanabilirlikte bir oyun sunmayı hedeflemektedir. Sınırlamalar, gelecekteki geliştirme aşamalarında ele alınabilecek ve genişletilebilecek alanları belirtmektedir.

## BÖLÜM 2. LİTARATÜR TARAMASI

### 2.1. Roguelike Oyunların Tarihçesi ve Tanıtımı

Roguelike oyun türü, adını 1980 yılında çıkan Rogue adlı oyundan almaktadır. Bu tür, rastgele oluşturulan zindanlar, kalıcı ölüm mekaniği ve yüksek zorluk seviyesi gibi özellikleriyle tanınır. Roguelike oyunlar, genellikle sıra tabanlı oynanışa sahip olup, her oynanışta farklı bir deneyim sunmayı hedefler.

Örnek Çalışmalar:

- Rogue (1980) oyunu, türün temelini atan ilk oyun olarak kabul edilir.
- Nethack (1987) ve Angband (1990), roguelike türünün evriminde önemli adımlar atan oyunlardır.

### 2.2. Aksiyon Roguelike Oyunların Tarihçesi ve Tanıtımı

Aksiyon roguelike oyunlar, geleneksel roguelike unsurları aksiyon tabanlı oynanış ile birleştirir. Bu tür, hızlı tempolu oyun mekaniği ve gerçek zamanlı strateji gerektiren bir yapıya sahiptir.

Örnek Çalışmalar:

- The Binding of Isaac (2011), aksiyon ve roguelike unsurlarını başarılı bir şekilde birleştirerek türün popülerleşmesine katkıda bulunmuştur.
- Dead Cells (2018), roguelike mekaniklerini platform öğeleri ile birleştirerek yenilikçi bir oyun deneyimi sunmuştur.

### 2.3. Hayatta Kalma Oyunları

Hayatta kalma oyunları, oyuncunun zorlu bir çevrede hayatta kalmasını amaçlayan oyunlardır. Kaynak yönetimi, stratejik düşünme ve çevresel tehditlerle başa çıkma gibi unsurlar içerir.

Örnek Çalışmalar:

- Don't Starve (2013), hayatta kalma oyunlarının mekaniklerini estetik bir görsellik ile sunarak türün dikkat çeken örneklerinden biri olmuştur.
- Subnautica (2018), su altı keşfi ve hayatta kalma unsurlarını bir araya getirerek yenilikçi bir yaklaşım sunmuştur.

### 2.4. Benzer Oyunlar



Şekil 2.1 Vampire Survivors

Vampire Survivors, poncle olarak da bilinen Luca Galante tarafından geliştirilen ve yayınlanan, roguelike bir shoot 'em up rol yapma video oyunudur. 17 Aralık 2021'de erken erişimde yayınlandı. Oyuncu, sürekli canavar dalgalarına karşı savaşırken otomatik olarak saldıran bir karakteri kontrol etmektedir. Hedef, saldırıda mümkün olduğunca uzun süre hayatta kalmak ve ek karakterleri, silahları ve karakter yeteneklerini güçlendirmektir.



Şekil 2.2 Brotato

"Brotato" hızlı tempolu, aksiyon dolu bir roguelike hayatta kalma oyunudur. Oyuncular, patates şeklindeki karakterleri kontrol ederek dalgalar halinde gelen düşmanlara karşı hayatta kalmaya çalışır. Her turda, oyuncular çeşitli silahlar, ekipmanlar ve güçlendirmeler toplayarak karakterlerini güçlendirir. Oyun, rastgele oluşturulan seviyeler ve zengin oyun mekaniği ile her oynanışta farklı bir deneyim sunar. Eğlenceli ve bağımlılık yapıcı yapısıyla, oyunculara sürekli olarak kendilerini geliştirme ve yeni stratejiler deneme fırsatı verir.

## BÖLÜM 3. TASARIM VE GELİŞTİRME SÜRECİ

### 3.1. Oyun Tasarımı

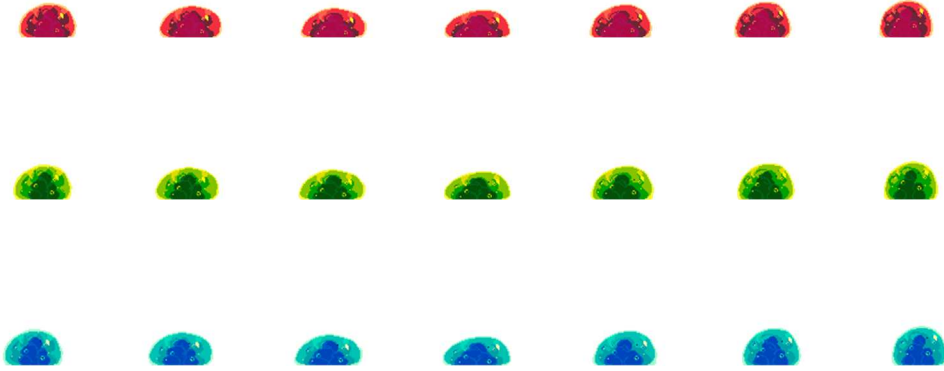
#### 3.1.1. Karakter tasarımı



Şekil 3.1 Karakter Görselleri

Oyunun ana karakteri için hazır assetler kullanılmıştır. Bu assetler, karakterlerin görünümünü ve animasyonları için kullanılmıştır.

#### 3.1.2. Düşman tasarımı

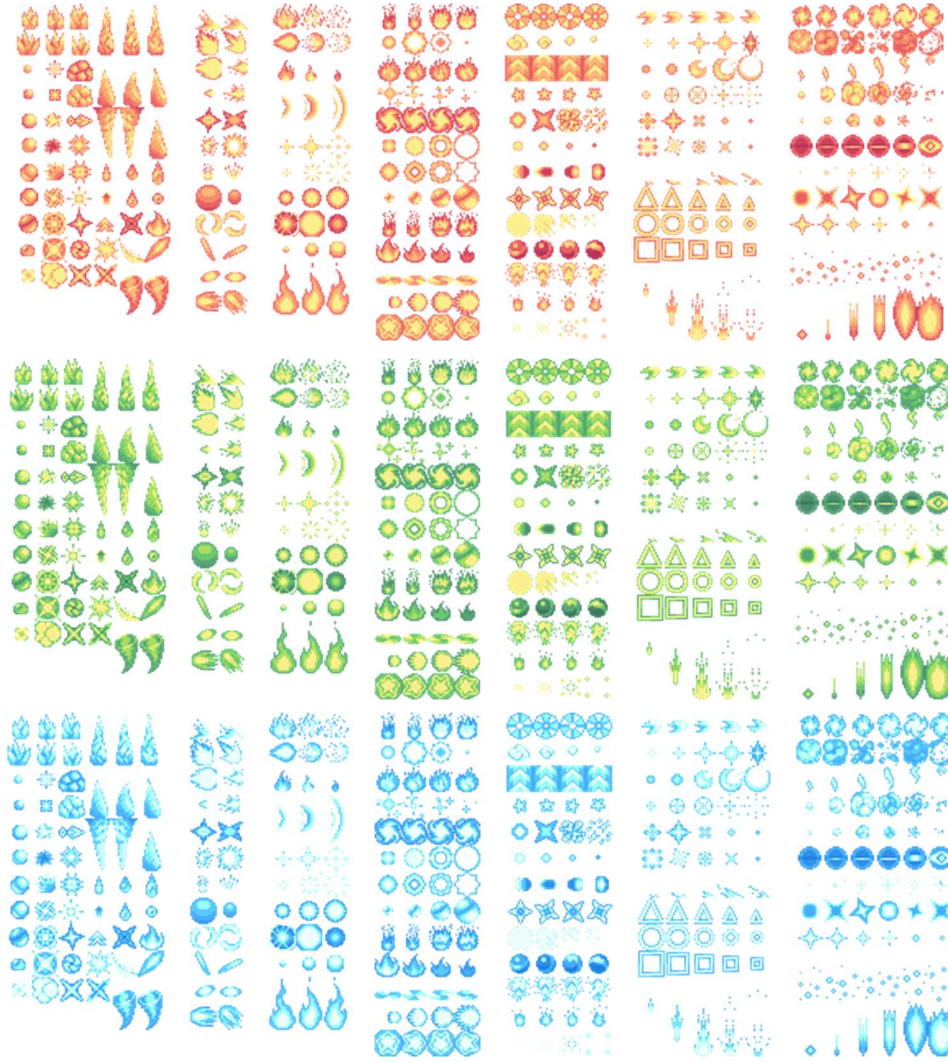


Şekil 3.2 Düşman Görselleri

Oyunun düşman karakteri için hazır assetler kullanılmıştır. Oyunda 3 farklı düşman türü bulunmaktadır. Her düşman türü için farklı assetler kullanılmıştır. Bu assetler, düşmanların görünümünü ve animasyonları için kullanılmıştır. Koşma, saldırma ve ölme animasyonları kullanılmıştır.



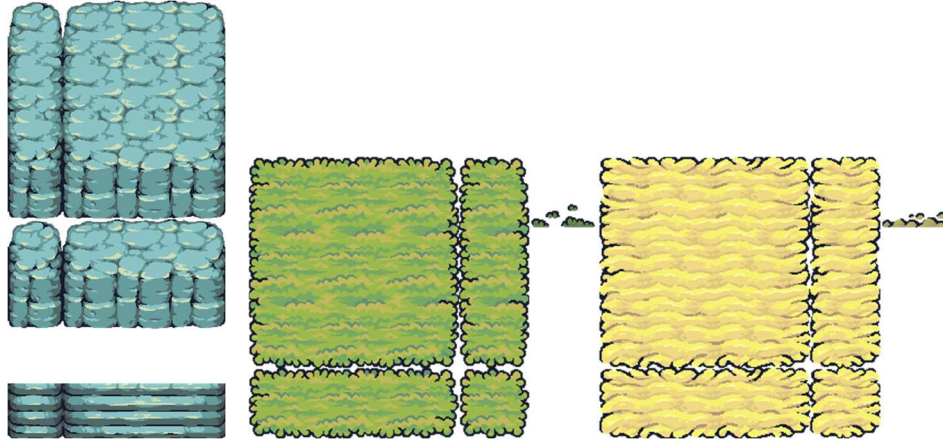
### 3.1.3. Yetenek tasarımı



Şekil 3.3 Yetenek Görselleri

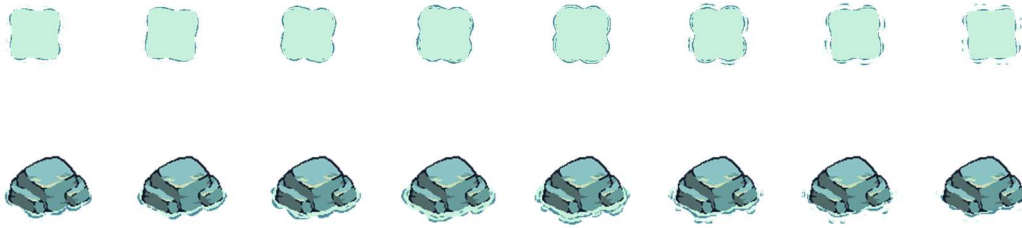
Oyunun karakterin yetenekleri için hazır assetler kullanılmıştır. Oyunda 3 farklı saldırı türü bulunmaktadır. Her saldırı türü için farklı assetler kullanılmıştır. Bu assetler, yeteneklerin görünümlerini ve animasyonları için kullanılmıştır. Assetten oyunumuza uygun olanlar seçilerek kullanılmıştır.

### 3.1.4. Harita tasarımı



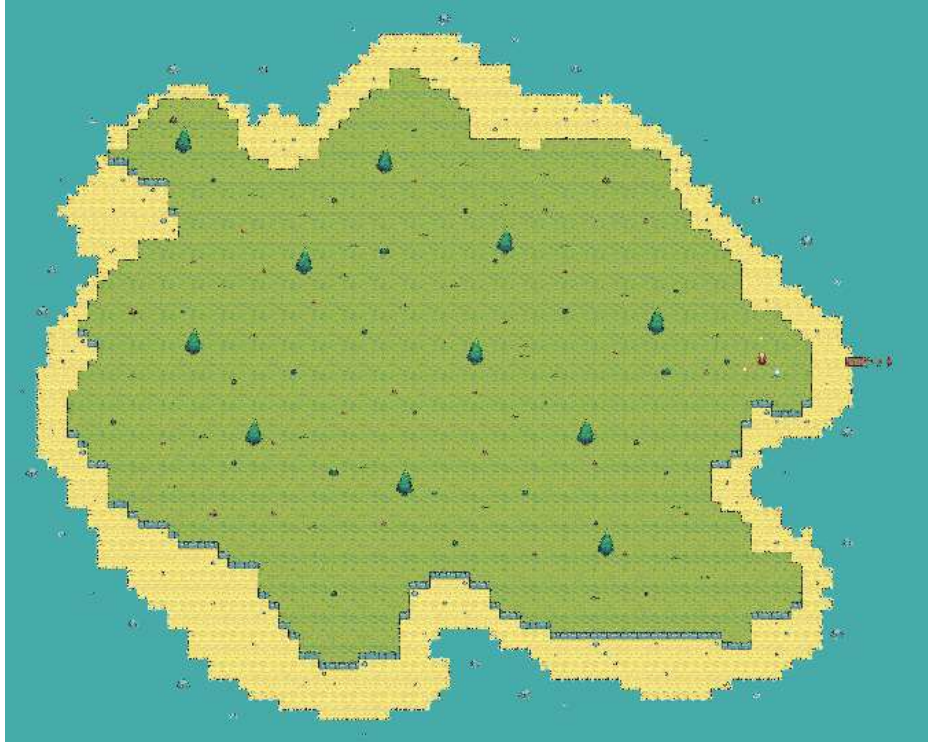
Şekil 3.4 Zemin Görselleri

Oyunun harita tasarımı için hazır assetler kullanılmıştır. Tilemap oluşturularak harita tasarımı yapılmıştır.



Şekil 3.5 Hareketli Zemin Görselleri

Daha iyi bir görsel deneyim sunmak için hareketli tilemapler oluşturulmuştur. Kıyıya vuran dalga ve kayalara vuran dalga animasyonları haritaya eklenerek daha canlı bir görünüm elde edilmiştir.



Şekil 3.6 Harita Tasarımı

Tilemap oluşturularak bir harita tasarımı yapılmıştır. Oyunun konseptine göre bir ada tasarımı yapılmıştır. Colliderlar eklenerek çarpışma kontrolleri sağlanmıştır.

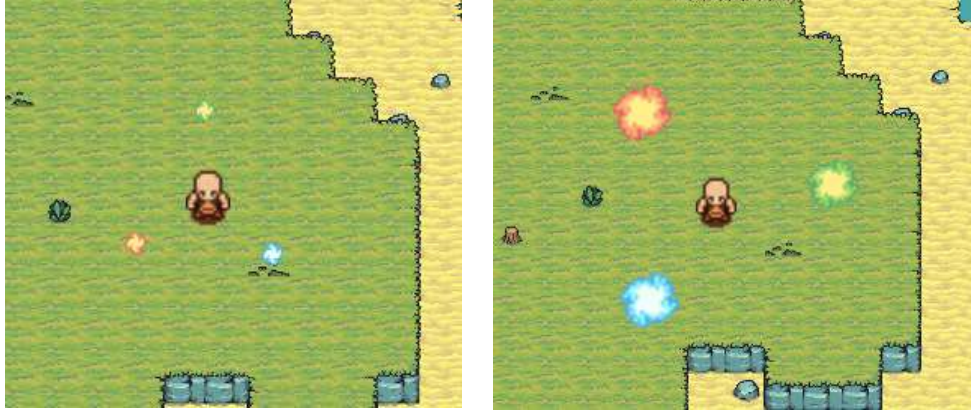
### 3.2. Oyun Mekanikleri

Oyun genel olarak bize saldıran düşmanlardan kaçınıp onları öldürerek seviye atlamak ve kendimizi geliştirmek üzerine kurulu bir sisteme sahip.

#### 3.2.1. Hareket sistemi

Oyun W,A,S ve D veya yön tuşları ile yukarı, aşağı, sağa ve sola hareket edebildiğimiz 2 boyutlu bir yapıya sahip.

### 3.2.2 Hasar sistemi



Şekil 3.7 Yetenek Görselleri

Oyunda Fire,Nature ve Water olmak üzere üç farklı saldırı türü bulunmaktadır. Ve bu saldırılara karşı dirençli yaratık türleri bulunmaktadır. Fire saldırı türüne dirençli bir yaratık fire saldırısından daha az hasar alacaktır. Bu üç farklı yetenek türünün hasarlarını tur sonlarında arttırabilmekteyiz. Ayrıca her seviye atladığımızda saldırı türlerinin büyüklüğü, dönüşü hızı ve hasar alanı gibi özelliklerini geliştirebilmekteyiz.

### 3.2.3. Can sistemi



Şekil 3.8 Can Görselleri

Oyunda 6 adet canımız bulunmaktadır. Bu canların hepsi tükendiği zaman oyun tamamen bitmektedir. Her tur başında can değerimiz tekrardan sıfırlanmaktadır. Ayrıca yaratıklardan düşük bir oranla düşen eti toplayıp can değerimizi tam olarak doldurabilmekteyiz.

### 3.2.4. Toplanabilir objeler



Şekil 3.9 Toplanabilir Obje

Oyunda düşmanları öldürdüğümüzde belirli bir oranla içlerinden toplanabilir nesneler düşmektedir. Düşman düşen eti topladığımız zaman canımızın tamamını doldurmaktayız. Bu da oyuncuya canı az kaldığı zaman toparlama şansı tanımaktadır.

## 3.3. Teknik Tasarım

### 3.3.1. Kullanılan araçlar ve teknolojiler

GIMP (GNU Image Manipulation Program), oyunun 2D grafik tasarımı, sprite oluşturma işlemleri ve assetler üzerindeki işlemler için kullanılmıştır. GIMP'in sunduğu geniş araç seti, karakter ve seviye tasarımlarının yaratılmasında büyük kolaylık sağlamıştır.

GitHub, versiyon kontrolü ve işbirliği aracı olarak kullanılmıştır. Git ve GitHub, geliştirme sürecindeki kod değişikliklerinin izlenmesini, farklı sürümlerin yönetilmesini sağlamıştır. GitHub üzerindeki depolar, projenin güvenli bir şekilde saklanmasını ve gerektiğinde geri dönülebilir sürümler oluşturulmasını mümkün kılmıştır.



### 3.3.2. Kodlama yaklaşımları

Oyun geliştirme sürecinde, temiz, sürdürülebilir ve etkili kod yazımını sağlamak amacıyla belirli kodlama yaklaşımları benimsenmiştir.

Kod yapısı, modüler bir yaklaşımla tasarlanmıştır. Bu sayede, her bir oyun bileşeni bağımsız olarak geliştirilebilmiş ve test edilebilmiştir. Modülerlik, kodun okunabilirliğini ve bakımını kolaylaştırmış, aynı zamanda yeniden kullanılabilirliği artırmıştır.

Nesne yönelimli programlama prensipleri, karakterler, düşmanlar ve oyun mekaniği gibi unsurların sınıflar ve nesneler aracılığıyla modellenmesini sağlamıştır. Bu yaklaşım, kodun daha yapılandırılmış ve anlaşılır olmasına yardımcı olmuştur.

Ekip içi kod incelemeleri, yazılan kodun kalitesini artırmak ve olası hataları erken aşamada tespit etmek için düzenli olarak yapılmıştır. Bu süreç, kodun gözden geçirilmesini ve en stabil şekilde çalışmasını sağlamıştır.

### 3.3.3. Kullanıcı arayüz(UI) tasarımı



Şekil 3.10 Hasar Geliştirme Ekranı

Bu ekranda karakterin yetenekleri ile vereceği hasar değerlerini arttırabiliyoruz. Her tur sonunda oyuncu bu ekran ile istediği hasar türünü geliştirebilmektedir. Her tur sade on yetenek puanı kullanabilmektedir.



Şekil 3.11 Yetenek Geliştirme Ekranı

Bu ekranda kullanıcı yetenekleri ile ilgili çeşitli geliştirmeler yapabilmektedir. İlk olarak Fire, Nature ve Water yeteneklerinin büyüklüğünü arttırabilmektedirler. Yetenek son seviyeye ulaştığı zaman Evolve ile yetenekleri daha güçlü bir seviyeye getirebilmektedirler. Onun haricinde tüm yetenekleri etkileyen dönüş hızı ve hasar alanını genişletebilme seçenekleride mevcuttur.



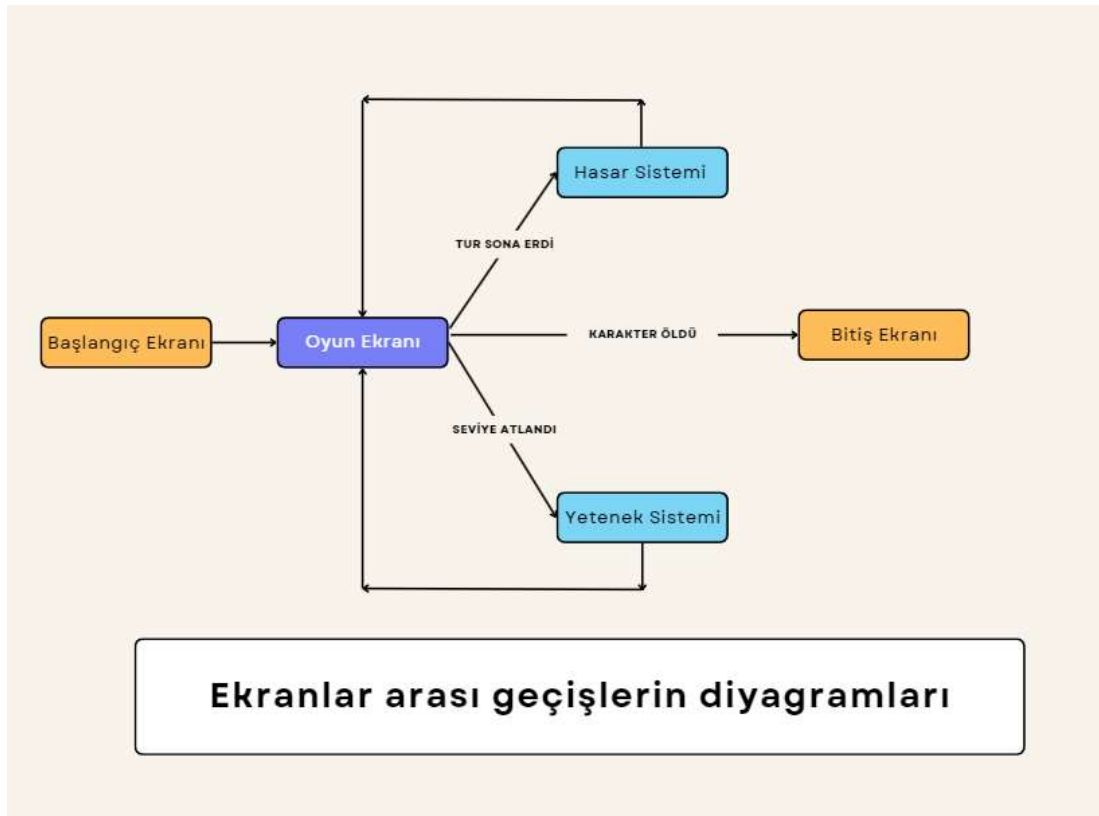
Şekil 3.12 HUD

Karakterin genel özelliklerini barındıran ve oyun esnasında ekranın sol üst köşesinde bulunan tasarım içinde can, seviye ve deneyim miktarını göstermektedir.



Şekil 3.13 Tur Göstergesi

Oyun esnasında kaçınıcı turda olduğumuzu takip edebildiğimiz bir arayüz. Ekranın orta üst tarafında bulunmaktadır.



Şekil 3.14 Akış Diagramı



## **BÖLÜM 4. UYGULAMA VE ENTEGRASYON**

### **4.1. Oyun Motoru Entegrasyonu**

Unity Proje Yapısı:

Unity'de projenin temel yapısı oluşturulmuş, gerekli sahneler, ve script dosyaları organize edilmiştir. Bu yapı, oyunun kolay yönetilmesini ve geliştirilmesini sağlamıştır.

Oyun Bileşenlerinin Birleştirilmesi:

Karakterler, düşmanlar, seviye haritaları ve diğer oyun bileşenleri Unity sahnelerine entegre edilmiştir. Bu süreçte, animasyonlar, ses efektleri ve oyun mekanikleri de motorun ilgili sistemlerine eklenmiştir.

### **4.2. Grafik ve Ses Entegrasyonu**

Grafiklerin Entegrasyonu:

GIMP ile oluşturulan 2D sprite'lar ve diğer görsel öğeler, Unity'ye import edilerek oyun sahnelerine yerleştirilmiştir. Sprite atlasları ve animasyonlar, Unity'nin Animator aracı kullanılarak entegre edilmiştir.

Seslerin Entegrasyonu:

Oyun için hazırlanan müzik ve ses efektleri, Unity'nin AudioSource ve AudioManager bileşenleri kullanılarak oyuna entegre edilmiştir. Her bir ses efekti ve müzik parçası, oyun olaylarına ve ortama uygun olarak yapılandırılmıştır.

### 4.3. Kullanıcı Arayüzü (UI) Entegrasyonu

Kullanıcı arayüzü bileşenlerinin oyuna nasıl entegre edildiği ve kullanıcı deneyimini nasıl iyileştirdiği bu alt başlıkta ele alınır.

#### UI Tasarımı ve Geliştirilmesi:

Oyunun kullanıcı arayüzü, Unity'nin UI araçları kullanılarak tasarlanmış ve geliştirilmiştir. Ana menü, oyun içi HUD (Heads-Up Display), envanter sistemi ve diğer UI elemanları, kullanıcı dostu bir deneyim sunmak amacıyla dikkatlice tasarlanmıştır.

#### UI İşlevselliğinin Eklenmesi:

UI elemanlarının işlevselliği, C# script'leri ile sağlanmıştır. Butonlar, slider'lar ve diğer etkileşimli UI bileşenleri, oyuncunun girişlerine ve oyun içi olaylara tepki verecek şekilde programlanmıştır.

## **BÖLÜM 5. TEST VE OPTİMİZASYON**

### **5.1. Oyun Test Süreçleri**

Oyun geliştirme sürecinde, çeşitli test yöntemleri kullanılarak oyunun kalitesi ve oynanabilirliği değerlendirilmiştir.

Test süreci, belirli bir plan ve metodolojiye göre yürütülmüştür. Bu kapsamda, farklı test türleri uygulanmıştır. Oyunun farklı senaryolarında karşılaşılabilecek olası durumlar ve bunların nasıl test edileceği belirlenmiştir. Kodun küçük parçaları (fonksiyonlar veya sınıflar) bağımsız olarak test edilmiştir. Birim testler, her bileşenin doğru çalıştığını doğrulamak için kullanılmıştır. Farklı bileşenlerin birlikte çalışmasını sağlamak amacıyla yapılan testlerdir. Bu testler, oyunun genel işlevselliğini kontrol etmek için kullanılmıştır.

### **5.2. Hata Ayıklama ve Düzeltme**

Geliştirme sürecinde karşılaşılan hataların tespiti ve düzeltilmesi için çeşitli araçlar ve teknikler kullanılmıştır.

Oyunun çökmesine neden olan hatalar tespit edilmiş ve düzeltilmiştir. Grafiklerde oluşan hatalar ve eksiklikler tespit edilerek düzeltilmiştir. Oyun içi mekaniklerin beklenmeyen şekilde çalışması durumunda, bu hatalar tespit edilerek gerekli düzeltmeler yapılmıştır.

Unity'de, hataların tespiti ve takibi için Debug.Log komutu kullanılmıştır. Bu komut, konsola hata mesajları ve diğer önemli bilgileri yazdırarak hata ayıklamayı kolaylaştırır. Unity Profiler aracı, oyunun performansını izlemek ve optimize etmek için kullanılmıştır. Bu araç, CPU, GPU ve bellek kullanımı hakkında detaylı bilgiler sunar.

### **5.3. Performans Optimizasyonu**

Oyunun en iyi performansı sağlaması için çeşitli optimizasyon teknikleri uygulanmıştır.

#### **5.3.1 FPS artırma teknikleri**

Yüksek çözünürlüklü sprite'lar yerine, düşük çözünürlüklü ve daha optimize edilmiş sprite'lar kullanılarak grafik performansı artırılmıştır. Görsel efektler ve shader'lar optimize edilerek, oyunun görsel kalitesinden ödün vermeden performans iyileştirmeleri yapılmıştır.

Kod yapısında gereksiz ve yoğun hesaplamalar tespit edilerek azaltılmış veya tamamen kaldırılmıştır. Daha verimli algoritmalar kullanılarak, kod performansı artırılmış ve işlem süresi kısaltılmıştır.

#### **5.3.2 Bellek yönetimi**

Bellek sızıntılarını önlemek için dinamik bellek yönetimi ve kaynakların doğru şekilde serbest bırakılması sağlanmıştır. Oyun içinde kullanılmayan nesneler tespit edilerek, bellekten silinmiştir.

## **BÖLÜM 6. SONUÇLAR VE ÖNERİLER**

### **6.1. Sonuçlar**

#### **6.1.1. Başarıyla tamamlanan özellikler**

Oyun içi temel mekanikler (karakter hareketleri, savaş sistemi, seviye geçişleri) başarılı bir şekilde uygulanmıştır. Grafik ve Ses Entegrasyonu: GIMP ile tasarlanan grafikler ve oyun için oluşturulan ses efektleri Unity'e başarılı bir şekilde entegre edilmiştir. Oyunun baştan sona oynanabilirliği sağlanmış, tüm mekanikler planlandığı şekilde uygulanmıştır.

#### **6.1.2. Teknik zorluklar**

Kısıtlı bütçe olduğu için ücretsiz assetler kullanılmıştır. Projeye uygun asset bulunması süreci kısıtlı imkanlardan dolayı uzun sürmüştür.

#### **6.1.3. Zaman yönetimi**

Projenin zaman çizelgesine uygun olarak tamamlanması için sıkı bir planlama ve takip süreci yürütülmüştür. Belirli aşamalarda yaşanan gecikmeler, etkili zaman yönetimi ile telafi edilmiştir.

#### **6.1.4. Öğrenilen dersler**

Geliştirme sürecinde Unity ile oyun geliştirme hakkında büyük bir ilerleme kat edilmiştir. Unity ile proje geliştirirken dikkat edilmesi gerekenler anlaşılmıştır. Bu, gelecekteki projelerde daha iyi bir yaklaşıma sahip olmamı sağlayacaktır.

## ÖZ GEÇMİŞ

Fırat Turan, 22.04.1996 da İstanbul'da doğdu. İlk, orta ve lise eğitimini Kartal'da tamamladı. 2014 yılında Şehit Öğretmen Hüseyin Ağırman Mesleki ve Teknik Anadolu lisesi, Bilgisayar Bölümü'nden mezun oldu. 2016 yılında Kocaeli Üniversitesi Bilgisayar Programcılığı Bölümü'nden mezun oldu. 2020 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü kazandı. 2013'te Biofarma İlaç Sanayi ve Ticaret A.Ş de lise stajını tamamladı. 2015 yılında Starpet Garzan Akaryakıt Pazarlama A.Ş de üniversite stajını tamamladı. 2017 yılında KM Kümsan Vinç Sistemleri Şirketinde bilgi işlem departmanında 2 yıl 3 ay çalıştı. Öğrenimi 4.sınıf olarak devam etmektedir.

## BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : UNITY İLE 2D OYUN TASARIMI

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): FIRAT TURAN

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
<b>Yazılı Çalışma</b>			
<b>Çalışma klavuza uygun olarak hazırlanmış mı?</b>	x	0-5	
<b>Teknik Yönden</b>			
<b>Problemin tanımı yapılmış mı?</b>	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
<b>Yapılan işlerin zorluk derecesi?</b>	x	0-25	
<b>Sözlü Sınav</b>			
<b>Yapılan sunum başarılı mı?</b>	x	0-5	
<b>Soruları yanıtlama yetkinliği?</b>	x	0-20	
<b>Devam Durumu</b>			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
<b>Diğer Maddeler</b>			
<b>Toplam</b>			

DANIŞMAN (JÜRİ ADINA):

DANIŞMAN İMZASI: