

**T.C MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ BÖLÜMÜ**

**LİSANS MÜHENDİSLİK PROJESİ**

**UNITY İLE MACERA TÜRÜNDE MASAÜSTÜ BİLGİSAYAR OYUNU GELİŞTİRİLMESİ**

**Hazırlayan**

**Furkan GÖLEMEZ**

**Danışman**

**Prof. Dr. İlhan TARIMER**

**15 Ocak 2025**

**MUĞLA**

# KABUL VE ONAY

Furkan Gölemez tarafından hazırlanan Unity ile Macera Türünde Masaüstü Bilgisayar Oyunu Geliştirilmesi adlı bu çalışmanın, Lisans Mühendislik Projesi olarak uygun olduğunu onaylarım.

Prof. Dr. İlhan TARIMER

Bu çalışma, jürimiz tarafından, Bilişim Sistemleri Mühendisliği Bölümü’nde Lisans mühendislik projesi olarak kabul edilmiştir.

Başkan: Prof. Dr. İlhan TARIMER İmza:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Üye: Doç. Dr. Sedat ATMACA İmza:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Üye: Araş. Gör. Hilal KOÇAK İmza:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# MÜHENDİSLİK PROJESİ BİLDİRİMİ

Bu çalışmadaki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca yazım kurallarına uygun olarak hazırlanan bu çalışmada yararlanılan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Furkan GÖLEMEZ

UNITY İLE MACERA TÜRÜNDE MASAÜSTÜ BİLGİSAYAR OYUNU GELİŞTİRİLMESİ

Furkan GÖLEMEZ

T.C

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ BÖLÜMÜ

15 Ocak 2025

# ÖZET

Bu proje, kullanıcılara gerçekçi bir hayatta kalma deneyimi sunan bir video oyunu tasarlamaya ve geliştirmeye yönelik çalışmalardan oluşmaktadır. Oyunun dünyası, 8 milyon metrekarelik devasa bir haritada, hayvanlar ve tehlikeli iskelet düşmanlarla dolu bir ortam yaratır. Kullanıcılar, açlık, susuzluk gibi mekaniklere dikkat ederek hayatta kalmaya çalışır ve çevreden topladıkları kaynaklarla üretim yaparak eşyalar üretir. Unity3D motoruyla geliştirilen oyunda interaktif öğeler, yapay zeka tabanlı yön bulma teknolojileri ve mesh manipülasyonları gibi modern yaklaşımlar içererek yenilikçi bir oyun deneyimi sunmayı hedeflemektedir.

Anahtar Kelimeler : Hayatta Kalma Oyunu, Unity3D, Yapay Zeka

Sayfa Adedi : 34

Bitirme Çalışma Yöneticisi : Prof. Dr. İlhan TARIMER

DEVELOPMENT OF AN ADVENTURE TYPE DESKTOP COMPUTER GAME WITH UNITY

Furkan GÖLEMEZ

T.C.

MUGLA SITKI KOÇMAN UNIVERSITY TECHNOLOGY FACULTY

INFORMATION SYSTEMS ENGINEERING DEPARTMENT

15 January 2025

# ABSTRACT

This project consists of work to design and develop a video game that offers users a realistic survival experience. The world of the game creates an environment full of animals and dangerous skeletal enemies on a huge 8 million square meter map. Users try to survive by paying attention to mechanics such as hunger and thirst, and produce items by crafting with the resources they collect from the environment. Developed with the Unity3D engine, the game aims to offer an innovative gaming experience by incorporating modern approaches such as interactive elements, AI-based navigation technologies and mesh manipulations.

Keywords : Survival Game, Unity3D, Artificial intelligence

Page Number :34

Advisor : Prof. Dr. İlhan TARIMER

# TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla, projeyi tamamlama sürecimde bana rehberlik ve kılavuzluk eden danışmanım Prof. Dr. İlhan TARIMER'e teşekkür ederim. Ayrıca, projenin çeşitli aşamalarında yardımlarıyla katkı sağlayan aileme ve arkadaşlarıma sonsuz minnettarım.

Furkan GÖLEMEZ

İÇİNDEKİLER

[KABUL VE ONAY i](#_Toc188007108)

[MÜHENDİSLİK PROJESİ BİLDİRİMİ ii](#_Toc188007109)

[ÖZET iii](#_Toc188007110)

[ABSTRACT iv](#_Toc188007111)

[TEŞEKKÜR v](#_Toc188007112)

[ŞEKİLLERİN LİSTESİ viii](#_Toc188007113)

[1.GİRİŞ 1](#_Toc188007114)

[2. UNITY 3D OYUN MOTORU 3](#_Toc188007115)

[2.1 Unity 3D Oyun Motorunun Tarihsel Gelişimi 3](#_Toc188007116)

[2.2 Unity 3D Oyun Motorunun NavMesh Teknolojisi 3](#_Toc188007117)

[2.2.1 NavMesh'in Temel Prensipleri 4](#_Toc188007118)

[2.2.2 NavMesh'in Avantajları 4](#_Toc188007119)

[2.2.3 NavMesh’in Kullanım Alanları 5](#_Toc188007120)

[2.2.4 NavMesh Özellikleri 5](#_Toc188007121)

[2.2.5 NavMesh'in Zorlukları ve İleri Düzey Kullanımları 7](#_Toc188007122)

[3. UNITY ASSET STORE 8](#_Toc188007123)

[4. C# PROGRAMLAMA DİLİ 10](#_Toc188007124)

[4.1 C# ve Unity 11](#_Toc188007125)

[5. VISUAL STUDIO CODE 12](#_Toc188007126)

[5.1 Visual Studio Code ve C# 12](#_Toc188007127)

[5.2 Visual Studio Code ve Unity 12](#_Toc188007128)

[6. HOLYPOLY OYUNUNDAKİ SİSTEMLER VE ÖZELLİKLER 14](#_Toc188007129)

[6.1 Kaynak Toplama Sistemi 14](#_Toc188007130)

[6.2 Envanter Sistemi 15](#_Toc188007131)

[6.3 Craft (Zanaat) Sistemi 16](#_Toc188007132)

[6.4 Ağaç Kesme Sistemi 17](#_Toc188007133)

[6.5 Dayanıklılık Sistemi 17](#_Toc188007134)

[6.6 Düşman Karakterler ve Avlanma Sistemi 18](#_Toc188007135)

[6.7 Yemek Pişirme Sistemi 19](#_Toc188007136)

[6.8 Gece Gündüz Sistemi 20](#_Toc188007137)

[6.9 Kayıt Alma Sistemi 21](#_Toc188007138)

[7.SONUÇ VE ÖNERİLER 22](#_Toc188007139)

[KAYNAKLAR 23](#_Toc188007140)

[EKLER 25](#_Toc188007141)

[ÖZGEÇMİŞ 26](#_Toc188007142)

# ŞEKİLLERİN LİSTESİ

[**Şekil 1.** Unity NavMesh Bake ekranı 6](#_Toc187821751)

[**Şekil 2.** NavMesh Agent ekranı 6](#_Toc187821752)

[**Şekil 3.** Navmesh Obstacle Ekranı 7](#_Toc187821753)

[**Şekil 4.** Örnek bir kod kesiti 11](#_Toc187821754)

[**Şekil 5.** Visual Studio Code’un Unity’e özel yapısı 13](#_Toc187821755)

[**Şekil 6.** Kodda yazılan kısmın Unity’de görünümü 13](#_Toc187821756)

[**Şekil 7.** HolyPoly’nin envanter sistemi 15](#_Toc187821757)

[**Şekil 8.** HolyPoly’deki craft sisteminin Survival kategorisi 16](#_Toc187821759)

[**Şekil 9.** HolyPoly’deki craft sisteminin Tools kategorisi 16](#_Toc187821760)

[**Şekil 10**. HolyPoly’deki dayanıklılık çubukları 18](#_Toc187821761)

[**Şekil 11.** HolyPoly’deki düşman örneği 19](#_Toc187821762)

[**Şekil 12.** HolyPoly’deki kamp ateşinin pişirme öncesi ekranı 20](#_Toc187821763)

[**Şekil 13.** HolyPoly’deki kamp ateşinin pişirme işlemi bittikten sonra ekranı 20](#_Toc187821764)

# 1.GİRİŞ

Bitirme çalışmasının amacı, araştırmacının oyun programlamaya yönelik kendini

geliştirmesi ve Unity 3D programını kullanarak haritayı ve modelleri tasarlayıp iyi bir görsellik ve sadelik yakalayarak kullanıcıya sunmayı amaçlamasıdır.

Proje, 8 000 000 metrekare gerçek hayata oranla, büyük bir adada geçmektedir. Oyuncular bu adada açlık, susuzluk, avlanma ve kaynak toplama gibi zorluklarla karşılaşacaklardır. Adanın gündüzleri barışçıl bir ortamı vardır; geceleri ise iskelet benzeri düşmanlarla dolu tehlikeli bir çevreye dönüşür.

Araştırmacı proje yapımına 2024 yılının bahar döneminde başlamış olup, 2025 yılının güz döneminde projeyi sonlandırmış ve gerekli olan kaynakları toplamıştır.

Bitirme çalışması olan HolyPoly oyununda yapay zeka sistemleri bulunmaktadır. Gerekli olan materyaller Asset Store üzerinden indirilmiştir. Bu projede Unity 2022.3.21f1 sürümü kullanılmıştır. Unity’nin kendi belgelerinden yararlanarak kodlaması ve tasarımı yapılmıştır [1].

Demirci ve Palancı 2019 yılında yaptıkları çalışmada oyun teorisinin ekonomi, işletme ve finans gibi alanlardaki uygulamalarını incelemişlerdir. Bu çalışma, oyun teorisinin farklı disiplinlerde nasıl kullanıldığını anlamak için kapsamlı bir kaynak sunmaktadır [(Demirci ve Palancı, 2019)][2].

Türkiye'de Oyun Geliştirme ve Grafik Tasarımı: Bu çalışma, Türkiye'deki oyun geliştirme sektöründe grafik tasarımının rolünü ve önemini analiz etmektedir. Oyun sektöründeki istihdam alanları ve yaratıcılığın önemi üzerine yapılan literatür taraması, sektördeki durumu anlamak için değerlidir [(İstanbul Gelişim Üniversitesi, 2022)] [3].

Gökçek ve Akbulut, 2022 yılında yaptıkları araştırmada, bağımsız oyun geliştiricilerinin oyun tasarımı sürecinde karşılaştıkları ihtiyaçları, problemleri ve farklılıkları keşfetmeyi amaçlamaktadır. Oyun geliştirme ve tasarım süreci üzerine yapılan literatür çalışması, bağımsız oyun geliştiricileri için önemli bulgular sunmaktadır [(Kavak Gökçek ve Akbulut, 2022)] [4].

Kızıltaş (2020), Muğla Sıtkı Koçman Üniversitesi Teknoloji Fakültesi’nde yaptığı çalışmada oyunlara yapay zeka karakterlerin nasıl ekleneceğini araştırmış, NavMesh teknolojisi kullanılması sonucuna varmıştır [5].

Geliştirilen oyunun içindeki varlıklar aşağıdaki paketlerden alınmıştır [6-10].

Stylized 3D Tools, Oyunun içindeki balta ve kazma 3d modelleri bu paketten alınmıştır.

Fantasy Wooden GUI, Oyunun içindeki kullanıcı arayüzü üretmek için bu paketten yararlanılmıştır.

Stylized Bear - RPG Forest Animal, Oyundaki ayı 3d modeli ve animasyonları için bu paketten yararlanılmıştır.

Realistic Terrain Textures, Oyundaki kaplamalar için bu paketten yararlanılmıştır.

AllSky, Oyundaki gökyüzü materyalleri için bu paket kullanılmıştır.

Bu mühendislik projesi raporunun 2nci Bölümü’nde “Unity 3D Oyun Motoru” açıklanmıştır. Çalışmanın 3üncü Bölümü’nde, “Unity Asset Store”, 4üncü Bölümü’nde, “C# Programlama Dili”, 5inci Bölümü’nde, “Visual Studio Code”, 6ncı Bölümü’nde, “HolyPoly Oyunundaki Sistemler ve Özellikler”, 7nci Bölümü’nde, “Sonuç ve Öneriler”, 8nci Bölümü’nde “Kaynaklar”, 9uncu Bölümü’nde ise “Ekler” verilmiştir. En son sayfada ise, hazırlayanın “Özgeçmiş” verilmektedir.

# 2. UNITY 3D OYUN MOTORU

Unity 3D, 2005 yılında Unity Technologies tarafından piyasaya sürülmüş popüler bir oyun motorudur. Kaliforniya merkezli şirket, bu motoru ilk olarak macOS platformunda oyun geliştirme kolaylığı sağlamak için yaratmıştır. Ancak zamanla Unity, Windows ve Linux gibi platformları da destekleyerek kapsama alanını genişletmiştir. İlk çıktığında genellikle başlangıç seviyesindeki geliştiricilere hitap ederken, sağladığı çok yönlülük ve düşük maliyetli lisanslama modeli ile sektörde hızla yayılmıştır.

Unity, çok platformlu geliştirme kapasitesi sayesinde geliştiricilere Android, iOS, Windows, macOS, oyun konsolları ve hatta sanal gerçeklik cihazları için uygulamalar oluşturma olanağı tanır.

Araştırmacı projenin temellerine bakarak uygulamasının tamamında Unity programını kullanmıştır. Projenin haritasını oyun motorunun kendisinde tasarlamış olup oyun içerisinde mevcut olan dağ, ağaç gibi materyalleri motorun vermiş olduğu imkanlar doğrultusunda oluşturmuştur.

## 2.1 Unity 3D Oyun Motorunun Tarihsel Gelişimi

2005 yılında Unity’nin ilk sürümü piyasaya sürüldü.2010 yılında Unity Free modelinin tanıtılması, motorun popülaritesini arttırdı.2015 yılında Unity 5 ile yeni grafik teknolojileri ve daha önceki versiyonlara kıyasla büyük çapta iyileştirmeler sağlandı.2021 ve sonrası için Unity Engine’in HDRP ve URP ile yeni nesil çözünürlük ve performans alanında daha fazla olanak sunduğu güncel sürümleri geliştirildi.

## 2.2 Unity 3D Oyun Motorunun NavMesh Teknolojisi

NavMesh (Navigation Mesh), oyun geliştirme dünyasında özellikle AI (Yapay Zekâ) karakterlerinin dünya üzerinde doğru ve verimli bir şekilde hareket edebilmesi için kullanılan bir navigasyon çözümüdür [11]. NavMesh, oyun içindeki bir harita ya da çevre üzerinde yapay zekanın izleyebileceği yolları tanımlayan bir yapı oluşturur. Karakterlerin engelleri aşmadan ve optimum yolları takip ederek belirlenen hedeflere ulaşmalarını sağlar.

## 2.2.1 NavMesh'in Temel Prensipleri

NavMesh, temelde çevredeki navigasyon açısından geçilebilir alanları birleştirerek bir mesh (ağ) oluşturur. Bu ağ üzerinde yapay zeka karakterleri, koşulabileceği, yürüyebileceği, zıplayabileceği ya da ilerleyebileceği yerleri bulur. Geliştirici, harita üzerinde geçilebilir alanları tanımladıktan sonra, NavMesh bu alanları dinamik bir şekilde optimize ederek hedefe ulaşan bir yol sunar.

Bir NavMesh ağı, belirli kurallar ve algoritmalar kullanılarak oyun dünyasında oluşturulur. Herhangi bir geometriye karşılık gelen geçilebilir bölgeler, poligonlar ya da triangular mesh (üçgen ağı) gibi yapılarla harita üzerinde temsil edilir.

Poligon: Her bir geçiş alanı (flat surface) genellikle bir poligon ile temsil edilir. Bu poligonlar, navigasyon ağının "köyleri" olur.

Zorluk: Düz ya da eğimli alanlar genellikle farklı seviyelerde olabilen mesafeyi tanımlar.

Engeller: NavMesh algoritması, haritada bulunan statik ve dinamik engelleri hesaba katarak yeni yol hesaplamaları yapar. Yani duvarlar, nesneler ya da düşmanlar birer engel olabilir.

## 2.2.2 NavMesh'in Avantajları

Zaman ve Performans Verimliliği: NavMesh, yapay zeka karakterlerinin doğru yollarda hareket etmesini sağlarken daha az hesaplama gücü gerektirir. Yapay zekanın yol almasını sağlamak için dinamik path finding (yol bulma) algoritmaları yerine sadece harita üzerindeki geçilebilir alanların hesaplanması yeterlidir.

Gerçekçilik: Yapay zeka karakterlerinin dünyada doğal ve gerçekçi bir şekilde gezinmesini sağlar. Karmaşık arazilerde, engelleri aşarken oldukça akıcı bir hareket deneyimi yaratılabilir.

Düşük Donanım Maliyeti: Performans açısından da düşük maliyetlidir çünkü NavMesh bir dizi karmaşık hesaplama işlemini önceden yaparak haritadaki hareketi çok daha hızlı simüle eder.

Unity, NavMesh sistemini kendi içinde entegre bir araç olarak sağlar. NavMesh, yerleşik araçlar ve API (Uygulama programlama arabirimi) yardımıyla oyun geliştirenlere hızla entegre etme imkânı sunar.

## 2.2.3 NavMesh’in Kullanım Alanları

Yapay Zeka (AI) Yürüyüşü: NavMesh, NPC'lerin (Oyuncu tarafından oynanamayan karakter) ya da düşmanların harita üzerindeki yolları ve yolların engellerden nasıl kaçındığını hesaplamak için kullanılır. Bu tür yönlendirmelerle çok daha gerçekçi bir yapay zeka hareketi sağlanabilir.

Oyun Dünyası Haritaları: NavMesh oyun haritasında yapay zeka için geçilebilir alanlar yaratmak amacıyla kullanabilir. Bu, yapay zekanın yalnızca mantıklı yollarda hareket etmesini sağlayacak şekilde yapılandırılabilir.

Araç ve Zorluk Seviye Yönetimi: Zorlu seviyelerde, oyuncu ve yapay zeka arasında geçiş yollarını optimize ederek, oyuncuya ya da karakterlere zorluklar eklenebilir.

## 2.2.4 NavMesh Özellikleri

NavMesh Bake (Pişirme): Navigasyon ağı pişirme aşamasında, Unity belirli bir seviyedeki geçilebilir alanları ve engelleri algılar ve bir NavMesh'i hesaplar. Bu süreçte harita üzerinde geçiş yapılacak alanlar doğru şekilde tespit edilir. NavMesh bake ekranı (Şekil 1) bu şekilde gözükür.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil . Unity NavMesh Bake ekranı

Şekil 1’de NavMesh Bake ekranının Unity üzerinde nasıl gözüktüğü gösterilmektedir. Step Height, Radius, Max Slope ve Bake tuşlarının açıklamaları yazılmıştır [11].

metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturulduNavMesh Agents: NavMesh Agent, Unity’nin yapay zeka (AI) karakterlerine yönlendirme işlevini sağlayan bir bileşendir. Her yapay zeka objesine eklenen bu bileşen, harita üzerinde bir hedef belirleyip en hızlı yoldan o hedefe ulaşmasını sağlar. NavMesh agent ekranı (Şekil 2) aşağıdaki şekilde gözükür.

Şekil . NavMesh Agent ekranı

NavMesh Obstacles: Bu özellik, oyun dünyasında dinamik engellerin NavMesh üzerinde etkili olmasını sağlar. Yani, karakterlere ve hareketli nesnelere göre yeniden hesaplamalar yapılabilir. NavMesh obstacle ekranı Şekil 3’teki gibi görünür.

metin, sayı, numara, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil . Navmesh Obstacle Ekranı

## 2.2.5 NavMesh'in Zorlukları ve İleri Düzey Kullanımları

Dinamik Engeller: NavMesh genelde statik haritalarda kullanılır, fakat Holypoly oyununda dinamik engeller ve hedefler gereklidir. Bunun için NavMesh Obstacle özellikleri kullanılır, fakat bu yeniden hesaplamayı zorlaştırır ve performans sorunlarına yol açar.

Kompleks Çevreler: Holypoly gibi haritası çok büyük ve dinamik yapıda oyunlarda NavMesh entegrasyonu karmaşıklaşır. Büyük haritalar için çoklu NavMesh seviyeleri oluşturmak gerekir. Örneğin, bir düz alanda yolculuk yapmakla, zorlu kayalık bir dağ yolunda yürümek farklı beceriler gerektirir.

# 3. UNITY ASSET STORE

Unity Asset Store, Unity oyun motoru ile geliştirilen projeler için tasarlanmış kapsamlı bir dijital pazar yeridir. Bu platform, hem profesyonel geliştiriciler hem de amatör kullanıcılar için geliştirme sürecini kolaylaştırmayı ve hızlandırmayı amaçlar. Unity Asset Store, bir oyunun veya uygulamanın gereksinim duyabileceği çok çeşitli içerik ve araçlar sunar. Bu içerikler arasında 3D modeller, ses efektleri, müzik dosyaları, animasyonlar, araç kitleri ve daha fazlası bulunur. Böylece geliştiriciler, kendi ihtiyaçlarına uygun içerikleri kolayca bulabilir ve projelerine entegre edebilir. [12].

Platformun bir diğer önemli avantajı, zaman ve maliyet açısından sağladığı tasarruftur. Sıfırdan bir varlık üretmek veya bir sistemi geliştirmek yerine, halihazırda optimize edilmiş ve yüksek kaliteli varlıklar edinmek mümkün olur. Bu durum, özellikle sınırlı bütçeye sahip bağımsız oyun geliştiricileri ve küçük ekipler için büyük bir avantajdır.

Unity Asset Store’un geniş bir kullanıcı topluluğuna hitap etmesi, bu platformun dinamik yapısını ve sürekli olarak yenilenen içerik havuzunu da beraberinde getirir. Dünyanın dört bir yanından bağımsız içerik oluşturucular, kendi varlıklarını bu mağazada satışa sunarak gelir elde edebilir. Böylece Unity Asset Store, yalnızca içerik tüketicilerine değil, içerik üreticilerine de önemli fırsatlar sunar.

Bu mağaza, kullanıcı dostu bir arayüz ve gelişmiş arama filtreleri ile donatılmıştır. Geliştiriciler, kategoriler, fiyat aralığı, popülerlik veya kullanıcı değerlendirmelerine göre filtreleme yaparak ihtiyaç duydukları varlıkları kolayca bulabilir. Ayrıca, her varlık sayfasında, kullanıcıların varlık hakkında geri bildirim bırakabileceği yorumlar ve puanlamalar bulunur. Bu özellikler, bir varlık satın almadan önce kalite ve uyumluluk konusunda güvenilir bir değerlendirme yapılmasına olanak tanır.

Son olarak, Unity Asset Store, farklı lisanslama seçenekleri ile projelerde kullanılacak varlıkların telif hakları konusunda şeffaflık sağlar. Lisanslama seçenekleri, projelerin ticari veya ticari olmayan kullanımına uygun şekilde düzenlenmiştir. Bu sayede, geliştiriciler projelerinin yasal yükümlülüklerini kolayca karşılayabilir.

Bağımsız geliştiriciler veya küçük ekipler genellikle kısıtlı bir bütçeyle çalışır. Asset Store sayesinde pahalı yazılımlar veya uzman ekiplere ihtiyaç duymadan profesyonel varlıklar edinebilirsiniz. Çok daha düşük maliyetle kaliteli içerik üretmeniz mümkün hale gelir. Örneğin, bir şehir ortamını sıfırdan modellemek yerine uygun fiyatlı bir paketle oyununuzun görsel tasarımını tamamlayabilirsiniz.

Unity Asset Store’da birçok içerik bağımsız geliştiriciler veya profesyonel ekipler tarafından sunulur. Bu geliştiriciler genellikle ürünlerine güncellemeler getirir ve kullanıcıların geri bildirimlerine cevap verir. Bu da Asset Store’un sürekli gelişen bir ekosisteme sahip olmasını sağlar.

# 4. C# PROGRAMLAMA DİLİ

C#, 2000 yılında Microsoft tarafından .NET Framework'ün bir parçası olarak geliştirilmiş bir programlama dilidir. Anders Hejlsberg tarafından tasarlanan bu dil, Java ve C++ gibi dillerden etkilenmiş, nesne yönelimli programlama prensiplerini güçlü bir şekilde destekleyen bir yapıya sahiptir.

C#, sınıflar ve nesneler etrafında bir yapı sunarak kodun yeniden kullanılabilirliğini artırır ve karmaşık sistemleri düzenli hale getirir. C#, statik tip güvenliği sağlayan, nesne yönelimli bir programlama dilidir. Bu özellik, yazılım geliştirme sürecinde hataların erken tespit edilmesine yardımcı olur ve yazılımın daha güvenilir hale gelmesini sağlar. Nesne yönelimli programlama (OOP) prensiplerini benimseyen C#, sınıflar, nesneler, kalıtım, çok biçimlilik (polimorfizm) ve kapsülleme gibi özelliklerin kullanılmasına olanak tanır. Otomatik bellek yönetimi sunan dil, çöp toplama (garbage collection) mekanizması sayesinde bellek sızıntılarının önüne geçer. Geliştirici dostu bir yaklaşımla tasarlanan C#, kolay okunabilir, anlaşılır ve modern bir sözdizimine sahiptir. .NET Core’un gelişimi ile birlikte platform bağımsızlık özelliği kazanan C#, artık Windows dışında macOS ve Linux gibi farklı platformlarda da çalışabilmektedir. Güvenlik açısından ise tip güvenliği sağlayarak veri doğruluğunu artırır ve potansiyel güvenlik açıklarını en aza indirir. Ayrıca, .NET Framework ile çalıştığı için sağlam bir kütüphane ve runtime (çalışma zamanı) desteği sunar [13].

C#, hem akademik çalışmalar hem de endüstriyel projeler için önemli bir dil olarak kabul edilmektedir. Akademik dünyada, algoritma geliştirme, yazılım tasarımı ve nesne yönelimli programlama gibi konularda eğitim verirken, endüstride özellikle kurumsal uygulamalar ve oyun geliştirme alanlarında geniş bir kullanım alanı bulmaktadır. Ayrıca, açık kaynak topluluğunun katkılarıyla sürekli olarak gelişmekte ve güncellenmektedir. Bu genişletilmiş uyumluluk, Unity oyun motoru gibi çok platformlu araçlarla uyumlu bir çalışma alanı oluşturur.

## 4.1 C# ve Unity

Unity oyun motorunda, C# ana programlama dili olarak kullanılır. Karakter Kontrolleri, oyuncu hareketlerini ve animasyonları düzenlemek için, fizik sistemleri, çarpışma, kuvvet, kütle gibi mekanik sistemleri kodlamak için, yapay zeka, düşman davranışları ve dinamik olayları kontrol etmek için, entegrasyon, veri işleme, dış kaynaklarla bağlantılar gibi oyun dışı işlemleri yönetmek için kullanılır. C# Programlama dili kullanılarak yazılan bir kod dosyasının örneği Şekil 4’te gösterilmiştir.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil . Örnek bir kod kesiti

Şekil 4’te projede kullanılan kayıt alma yöneticisi (Save Manager) scriptinin ilk satırlarındaki kodlar gösterilmektedir. Kodun ileri kısımlarında kullanılmak üzere değişkenlerin tanımlanması gösterilmiştir.

# 5. VISUAL STUDIO CODE

Visual Studio Code (VS Code), Microsoft tarafından geliştirilmiş, modern bir kod düzenleme ve geliştirme ortamıdır. Açık kaynaklı yapısı ve geniş eklenti desteği sayesinde, dünyanın en popüler IDE'lerinden (Integrated Development Environment) biri haline gelmiştir. Araştırmacı projede esnek yapısından dolayı Visual Studio Code tercih etmiştir.

VS Code, düşük sistem kaynakları kullanarak hızlı bir çalışma ortamı sunar. Büyük projeler üzerinde bile hızlı kod işleme yetenekleri gösterir. Windows, Linux ve macOS platformlarında sorunsuzca çalışabilir. VS Code, kullanıcıların ihtiyaçlarına göre özelleştirilebilen bir yapıya sahiptir. C# geliştiricileri için .NET eklentisi kritik önem taşır.

## 5.1 Visual Studio Code ve C#

Visual Studio Code, Unity projelerinde C# kullanımı için en ideal araçlardan biri olarak öne çıkar ve geliştirme sürecini kolaylaştıran birçok önemli özelliğe sahiptir. Geliştiricilerin yazdığı kodu anlamlı önerilerle otomatik olarak tamamlayan IntelliSense, kod yazımını hızlandırırken hataları en aza indirir. Bunun yanı sıra, kodda kırılma noktaları (breakpoints) ekleyerek hataların hızlıca tespit edilmesini sağlar, böylece hata ayıklama süreci daha etkili bir hale gelir. Ayrıca, kodun çalıştırılması veya test edilmesi gibi süreçler doğrudan VS Code'un içinden gerçekleştirilebilir, bu da geliştiricilere hem zaman kazandırır hem de projelerini daha verimli bir şekilde yönetmelerine olanak tanır [14].

## 5.2 Visual Studio Code ve Unity

Unity projelerinde VS Code kullanımı, geliştiricilere oldukça esnek bir geliştirme ortamı sunar. Unity Projesinde, “Preferences > External Tools” menüsünden Visual Studio Code seçilerek bağlanabilir. Visual Studio Code (Şekil 5) kullanarak Unity inspector (Şekil 6) ekranına erişen kod satırı örnek olarak verilmiştir.

Özel Unity eklentisi, oyuncu hareketlerinden yapay zeka yazılımına kadar her işlemi C# ile entegre eder.

**metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Şekil . Visual Studio Code’un Unity’e özel yapısımetin, ekran görüntüsü, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil . Kodda yazılan kısmın Unity’de görünümü

Şekil 5-6 ‘ da hayvanları kontrol ettiğimiz “Animal” scriptinin değişkenleri ve o değişkenlerin unity inspector’da nasıl görüldüğü gösterilmiştir.

# 6. HOLYPOLY OYUNUNDAKİ SİSTEMLER VE ÖZELLİKLER

HolyPoly’de kullanılan sistemler oyuncunun çevresiyle etkileşim kurmasını, kaynak toplamasını, tehlikelerden korunmasını ve ihtiyaçlarını yönetmesini sağlayacak şekilde tasarlanmıştır. Bu tür oyunlarda en temel sistemlerden biri olan envanter sistemi (Sırt çantası sistemi), oyuncunun topladığı kaynakları düzenlemesine olanak tanır. Crafting (Zanaat) sistemi, oyuncunun topladığı kaynakları kullanarak yeni eşyalar ve araçlar üretmesi sağlanır. Sağlık ve dayanıklılık sistemleri, oyuncunun hayatta kalmak için yiyecek, su ve sağlık ihtiyaçlarını karşılamasını zorunlu kılar. Ayrıca, gün döngüsü gibi çevresel sistemler, oyuncuyu değişen koşullara uyum sağlamaya iter. Bunun yanı sıra, yapay zeka tabanlı düşmanlar ve vahşi yaşam gibi unsurlar, oyun dünyasını daha tehlikeli ve heyecan verici hale getirir. Bu sistemler bir arada çalışarak, oyuncuya sürekli bir meydan okuma ve derin bir oyun deneyimi sunar.

## 6.1 Kaynak Toplama Sistemi

Kaynak toplama sistemi, HolyPoly’nin temel unsurlarından biri olarak, oyuncunun çevreden farklı türde kaynakları toplayarak bunları çeşitli amaçlarla kullanmasını sağlar. Bu sistem, oyun dünyasında yer alan ağaç, taş ve su gibi doğal öğeleri içerir ve oyuncunun bu kaynakları toplamak için belirli araçlara veya yöntemlere ihtiyaç duymasını gerekir (Odun toplamak için balta, su depolamak için matara üretmek vb.). Kaynak toplama süreci genellikle oyuncunun keşif yapmasını teşvik ederken, topladığı materyalleri envanter sistemine eklemesine olanak tanır. HolyPoly’de kaynaklar, çevrede sınırlı miktarda bulunarak stratejik düşünmeyi gerektirir. Ayrıca, kaynak toplama sistemi crafting (Zanaat) ve yapı inşa etme gibi diğer sistemlerle entegre edilerek oyuncunun oyun içi ilerlemesini ve hayatta kalma şansını artırır. Bu sistem, aynı zamanda risk ve ödül dengesi yaratarak, oyuncuyu tehlikeli bölgelerde daha değerli kaynaklar aramaya teşvik eder.

## 6.2 Envanter Sistemi

Envanter sistemi, oyuncunun topladığı tüm kaynakları, eşyaları, araçları ve diğer öğeleri saklamasını, düzenlemesini ve kullanmasını sağlayan bir mekanizmadır. Bu sistem, oyuncunun hayatta kalma ve ilerleme stratejisini şekillendiren önemli bir sistemdir. Bir arayüz üzerinden sunulan envanter sistemi, oyuncunun sahip olduğu öğelerin görsel olarak sergilendiği bir tablo yapısına sahiptir (Şekil 7).

Envanter sistemi, HolyPoly’de sınırlı kapasiteye sahip olacak şekilde tasarlanmıştır. Sınırlı kapasiteye sahip envanterler, oyuncunun hangi eşyaları taşıyacağına karar verirken stratejik düşünmesini gerektirir.

Envanter sistemi diğer oyun mekanikleriyle entegre çalışır. Örneğin, crafting (Zanaat) sistemi için gerekli malzemeler, envanterden seçilerek kullanılır. Ayrıca, araçlar veya yiyecekler gibi öğeler envanterden seçilip doğrudan kullanılabilir. Envanter sistemi, oyuncunun oyundaki ilerlemesini kolaylaştıran, karar alma süreçlerini etkileyen ve kaynak yönetimi becerilerini sınayan bir yapıya sahiptir. Bu nedenle, HolyPoly oyununda kilit bir rol oynar.

ekran görüntüsü, sanat içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil . HolyPoly’nin envanter sistemi

Şekil 7’de verilen oyun içi resimde, envanter sisteminin oyun içinde görünümü verilmiştir. Kullanıcının etkileşime geçebileceği eşyaları görebilmesini sağlar.

## 6.3 Craft (Zanaat) Sistemi

Craft sistemi, oyuncuların topladıkları malzemeleri kullanarak yeni eşyalar, araçlar veya yapılar oluşturmasına olanak tanıyan bir oyun mekanizmasıdır. Bu sistem, oyuncunun yaratıcılığını ve stratejik düşünme becerisini ödüllendirirken, oyun içi ilerlemenin ve hayatta kalmanın temel bir unsuru olarak öne çıkar.

Craft sistemi bir blueprint (tarif) tabanlı yapıya sahiptir. Oyuncular, belirli bir eşya veya aracın üretimi için gerekli olan malzemeleri öğrenmek ve bunları toplamak zorundadır. HolyPoly’de bu tarifler başlangıçta oyuncuya açık bir şekilde sunulur (Şekil 10). Örneğin, bir balta üretmek için çubuk ve taş, bir kamp ateşi yapmak için taş ve odun gibi malzemeler gerekir. Bu sistem, oyunculara esneklik sunarak oyun tarzlarını kişiselleştirmelerine olanak tanır. Oyuncular, acil durumlar için hayatta kalma ekipmanları yapabilir, daha güçlü araçlar üreterek düşmanlarla mücadele edebilir veya çevresel tehlikelere karşı dayanıklı barınaklar inşa edebilir.

**metin, dikdörtgen, resim çerçevesi, el yazısı içeren bir resim

Açıklama otomatik olarak oluşturuldu**Craft sistemi envanter, kaynak toplama ve ilerleme sistemleriyle entegre bir şekilde çalışır. Bu da oyuncunun oyun dünyasıyla daha derin bir bağ kurmasını sağlar ve sürekli olarak yeni hedefler oluşturmasına olanak tanır. Bu sistem, HolyPoly’de oyuncuyu hem ödüllendirici hem de zorlayıcı bir deneyimle meşgul eden temel bir mekanizmadır .

Şekil . HolyPoly’deki craft sisteminin Survival kategorisi

metin, bina, kapı, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 8. HolyPoly’deki craft sisteminin Tools kategorisi

Şekil 8 ve 9’da HolyPoly oyunundaki üretme ekranları gösterilmiştir. Tools (araçlar) ve Survival (hayatta kalma) tablolarının oyun içi resimleridir.

## 6.4 Ağaç Kesme Sistemi

Ağaç kesme sistemi, HolyPoly’de oyuncuların çevredeki ağaçları keserek odun gibi temel kaynakları toplamasını sağlayan bir mekanizmadır. Bu sistem, oyuncunun hem kaynak yönetimini hem de çevreyle etkileşim düzeyini artıran önemli bir oyun içi unsurdur. HolyPoly’de bir araç kullanımı gerektirir; bu araç baltadır. Oyuncunun başlangıçta taş ve çubuk gibi kaynakları toplamasını ve ağaç kesmek için bir balta üretmesini teşvik eder.

Ağaç kesme işlemi, oyuncunun belirli bir ağaca yaklaşarak onu hedef alması ve kesme eylemini başlatmasıyla gerçekleşir. Kesme işlemi, yalnızca balta ile tek bir vuruşla tamamlanacak şekilde tasarlanmıştır. Kesildikten sonra kazanılan malzemeler daha sonra crafting, yapı inşa etme veya yakacak olarak kullanılabilir.

Bu sistem, kaynak toplama, yapı inşa etme ve craft sistemleriyle bütünleşik çalışır. Oyuncuya somut bir ilerleme hissi verirken, keşif ve stratejik karar alma süreçlerini zenginleştirir.

## 6.5 Dayanıklılık Sistemi

Dayanıklılık sistemi, oyuncunun belirli eylemleri gerçekleştirmek için sahip olduğu enerji seviyesini temsil eden ve sınırlı bir kaynak olarak sunulan bir mekanizmadır. Bu sistem, oyuncunun fiziksel aktiviteleri sırasında tüketilen enerjiyi yönetmesini zorunlu kılar ve stratejik bir oyun deneyimi sağlar.

Dayanıklılık sistemi, enerji çubukları ile görselleştirilir ve oyuncu koşma, zıplama, ağaç kesme ve dövüş gibi eylemler gerçekleştirdiğinde bu çubuk azalır. Dayanıklılık tamamen tükendiğinde, oyuncunun bu tür eylemleri geçici olarak yapamaz hale gelmesi veya daha yavaş hareket etmesi gibi kısıtlamalar devreye girer. HolyPoly’de tükenmiş dayanıklılık, oyuncunun saldırılara daha açık hale gelmesine veya ölmesine sebep olabilir.

Bu sistem, dinlenme, yemek yeme ve su içme gibi çeşitli yollarla yeniden doldurulabilir.

Dayanıklılık sistemi, oyuncunun hareketlerini ve eylemlerini dikkatlice planlamasını gerektirerek hem risk-ödül dengesi sağlar hem de gerçekçi bir deneyim sunar. Bu mekanizma, oyuncunun sürekli olarak çevresel koşullara ve mevcut kaynaklara uyum sağlamasını zorunlu kılarak, hayatta kalma ve aksiyon odaklı oyunlarda önemli bir rol oynar.

ekran görüntüsü, renklilik, gökyüzü, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 10. HolyPoly’deki dayanıklılık çubukları

Şekil 10’da kullanıcının gördüğü oyuncunun oyundaki yaşam enerjisi, açlık ve susuzluk barları gösterilmiştir. Bu şekilde oyunun her anında kullanıcının bu değerlerden haberi olması sağlanmıştır.

## 6.6 Düşman Karakterler ve Avlanma Sistemi

Düşman Karakterler ve Avlanma Sistemi, HolyPoly’de hem bir tehdit unsuru hem de kaynak elde etme aracı olarak önemli bir rol oynar. Bu sistem, oyuncunun oyun dünyasında karşılaştığı yapay zekâ tarafından kontrol edilen düşmanlar ve avlanabilecek hayvanlar gibi karakterleri kapsar. Örnek olarak düşman karakterin resmi Şekil 11’de verilmiştir.

Düşman karakterler, oyuncunun hayatta kalma becerilerini test eden ve oyuna gerilim katan varlıklardır. Bu düşmanlar, karmaşık yapay zekâ ile donatılmış vahşi hayvanlar ve iskeletlerdir. Çeşitli saldırı desenlerine ve davranışlara sahip olan düşmanlar, oyuncuyu stratejik kararlar almaya ve savunma veya saldırı için hazırlıklı olmaya zorlar. Bazı düşmanlar oyuncuya doğrudan saldırırken, bazıları yalnızca oyuncu belirli bir alana girdiğinde veya tetiklendiğinde tepki verir. Bu da oyuncuya, düşmanları gözlemleme ve duruma göre bir strateji belirleme imkânı tanır.

Avlanma sistemi, yiyecek, deri, kemik veya diğer kaynakları elde etmek için oyuncunun hayvanları avlamasını sağlar. Avlanma sırasında, oyuncu farklı araçlar kullanabilir. Hayvanlar, genellikle gerçekçi davranışlar sergiler.

**ağaç, bilgisayar oyunu, ekran görüntüsü, video oyunu yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**Bu iki sistem birbiriyle bütünleşik bir şekilde çalışır; oyuncu, düşman hayvanları avlayarak kaynak elde ederken, aynı zamanda kendisini savunmak veya saldırı için hazırlık yapmak zorunda kalabilir. Düşman karakterler ve avlanma sistemi, oyun dünyasına gerçekçilik ve dinamiklik katarak, oyuncunun hayatta kalma çabalarını hem zorlaştırır hem de ödüllendirici bir deneyime dönüştürür.

Şekil 11. HolyPoly’deki düşman örneği

Şekil 11’de oyundaki ayı düşmanının oyuncuyu kovalama anı oyun içi ekran görüntüsü olarak verilmiştir. Oyundaki ayılar bu resimdeki gibi gözükmektedir.

## 6.7 Yemek Pişirme Sistemi

Yemek pişirme sistemi, HolyPoly’de oyuncunun topladığı malzemeleri kullanarak daha besleyici ve dayanıklılık artırıcı yemekler hazırlamasını sağlayan bir sistemdir. Bu sistem, oyuncunun hayatta kalmasını kolaylaştırırken stratejik karar alma ve kaynak yönetimi gibi öğeleri de güçlendirir.

Yemek pişirme bir kamp ateşi gerektirir (Şekil 12). Oyuncu, hayvanları avladığında toplayabileceği rastgele miktarda et kazanır. Eğer oyuncu eti pişirmeden yer ise HolyPoly’de sağlığını kaybetmesine neden olacak şekilde cezalandırılır. Bu sebeple oyuncu kamp ateşi üretmeli veya haritanın farklı noktalarında bulunan yemek pişirme araçlarına erişmeli ve eti pişirmelidir (Şekil 13). HolyPoly’de pişirilmiş et susuzluk, açlık ve sağlığı da pozitif yönde etki ettiği için oyuncunun oyunu keşfetmesini teşvik eder. Yemek pişirme sistemi, hayatta kalma deneyimini zenginleştiren ve oyuncunun çevresiyle daha derin bir etkileşim kurmasını sağlayan temel bir oyun mekanizmasıdır. Bu sistem hem pratik hem de yaratıcı bir deneyim sunarak, oyunun genel oynanışına **metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**önemli bir katkıda bulunur.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 13. HolyPoly’deki kamp ateşinin pişirme işlemi bittikten sonra ekranı

Şekil 12. HolyPoly’deki kamp ateşinin pişirme öncesi ekranı

Şekil 12 ve 13’de yemek pişirme ekranı ve pişirme işleminin bittikten sonra kullanıcıya gözüken kullanıcı arayüzlerinin oyun içi resimleri gösterilmiştir. Kullanıcı bu şekilde yemek pişirme işlemlerini gerçekleştirebiliyor.

## 6.8 Gece Gündüz Sistemi

Gece-gündüz sistemi, HolyPoly’de zamanın akışını simüle eden bir sistemdir ve oyuncunun oyun dünyasına daha fazla gerçekçilikle bağlanmasını sağlar. Bu sistem, görsel ve oynanış unsurlarında belirgin değişikliklere yol açarak dinamik bir oyun deneyimi sunar.

Gece-gündüz sistemi, günün farklı zaman dilimlerini temsil eden döngülerden oluşur. Bu döngü sırasında ışık, gölgeler ve çevresel sesler gibi görsel ve işitsel unsurlar değişir. Gündüz saatlerinde ortam daha aydınlık, güvenli ve keşif için uygunken, gece saatlerinde karanlık, düşük görüş mesafesi ve artan tehlike gibi unsurlar öne çıkar. HolyPoly’de gece daha güçlü düşmanlarla başa çıkması gerekir.

Bu sistem oyuncunun davranışlarını ve stratejilerini etkiler. Örneğin, gündüz kaynak toplamak veya yapı inşa etmek için uygunken, gece tehlikelerden kaçınmak veya saklanmak gerekebilir.

Gece-gündüz sistemi, aynı zamanda görsel estetik ve atmosfer yaratma açısından da önemlidir. Oyuncular, güneşin batışı veya ay ışığının etkisi gibi doğal olayları deneyimleyerek oyun dünyasının dinamik bir parçası haline gelir. Bu sistem, oyun içinde çevresel bir meydan okuma ve çeşitlilik sunarak hem gerçekçi hem de sürükleyici bir oynanış sağlar.

## 6.9 Kayıt Alma Sistemi

Kayıt alma sistemi, oyuncunun oyun içindeki ilerlemesini korumasını sağlayan bir sistemdir ve modern oyunların vazgeçilmez bir parçasıdır. Bu sistem, oyuncunun başarısız olduğunda veya oyunu kapattığında kaydedilmiş noktadan devam etmesine olanak tanır, böylece tekrar aynı aşamaları geçme zorunluluğunu ortadan kaldırır ve oyun deneyimini daha erişilebilir hale getirir.

Kayıt alma sistemi HolyPoly’de Manuel kayıt (elle yapılan), oyuncunun belirli bir menü aracılığıyla ilerlemesini istediği zaman kaydetmesine olanak tanır. Bu yöntem, oyuncuya kayıt noktalarını seçme özgürlüğü sunar.

Kayıt sistemi, aynı zamanda teknik detaylarla da güçlendirilir. Örneğin, oyuncunun bulunduğu konum, sahip olduğu kaynaklar ve diğer önemli bilgiler kaydedilir ve bu veriler, oyunun yeniden yüklendiğinde aynı durumun korunmasını sağlar.

Bu sistem, oyuncuya hem güvenlik hissi verir hem de oyun dünyasında cesurca keşif yapma özgürlüğü tanır. Kayıt alma sistemi, özellikle büyük ve karmaşık oyunlarda, oyuncunun zamanını ve çabasını koruyan, hayati bir oyun mekanizmasıdır.

# 7.SONUÇ VE ÖNERİLER

Bu oyunu geliştirme süreci, teknik becerilerin yanı sıra yaratıcı düşüncenin, problem çözme yeteneklerinin ve ekip içi iş birliğinin önemini bir kez daha ortaya koymuştur. Geliştirme sırasında karşılaşılan başlıca zorluklar, proje yönetimi, oyun mekaniklerinin optimize edilmesi, kaynak yönetimi ve kullanıcı deneyimini iyileştirme üzerine yoğunlaşmıştır. Bu zorlukların üstesinden gelmek için yenilikçi yaklaşımlar ve iteratif geliştirme yöntemleri benimsenmiştir.

Sürecin en dikkat çekici yönü, teknik ve yaratıcı alanların birleştiği kritik kararlarda ortaya çıkmıştır. Özellikle oyun içi sistemlerin dengelenmesi, performans sorunlarının giderilmesi ve görsel-işitsel unsurların optimize edilmesi, detaylı test ve geri bildirim mekanizmaları ile sağlanmıştır

Sonuç olarak, bu proje yalnızca teknik bilgi ve becerilerin değil, aynı zamanda dayanıklılık, öğrenme isteği ve iş birliği becerilerinin de önemini vurgulamıştır. Gelecekteki projelerde bu süreçte elde edilen deneyimlerin rehberlik edeceği ve daha etkili geliştirme süreçlerinin uygulanacağı öngörülmektedir. Bu çalışma, oyun geliştirme alanındaki zorluklara yönelik uygulamalı çözümler sunarak, sektördeki uygulayıcılara ve akademik araştırmacılara faydalı bir referans niteliği taşımaktadır.

# KAYNAKLAR

[1] Unity Documentation, "Unity Oyun Motoru Belgeleri," Unity Technologies. <https://docs.unity3d.com>.

[2] Demirci, E. (2019). ULUSAL YAZINDA OYUN TEORİSİ ÜZERİNE GENİŞLETİLMİŞ BİR LİTERATÜR TARAMASI. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *11*(29), 530-549. [3] İstanbul Gelişim Üniversitesi, “Türkiye’de oyun geliştirme alanında yaygınlaşan grafik tasarımı istihdam alanlarının incelenmesi”, Sanat ve Tasarım Dergisi, 2022.

[3]Leblebici, E. (2022). Türkiye’de Oyun Geliştirme Alanında Yaygınlaşan Grafik Tasarımı İstihdam Alanlarının İncelenmesi. *ART/icle: Sanat ve Tasarım Dergisi*, *2*(1), 56-71.

[4] Ş. Kavak Gökçek And D. Akbulut, "A FIELD STUDY TO EXPLORE NEEDS, PROBLEMS, SIMILARITY AND DIFFERENCES IN THE DESIGN PROCESS WHEN DEVELOPING INDIE VIDEO GAMES," *Sanat ve Tasarım Dergisi* , no.30, pp.187-207, 2022

[5] İsmail Kızıltaş BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ LİSANS BİTİRME ÇALIŞMASI, Muğla Sıtkı Koçman Üniversitesi, 2020.

[6] Unity Asset Store, "Unity Asset Store Resmi Sitesi," Unity Technologies. <https://assetstore.unity.com/packages/3d/props/tools/stylized-3d-tools-91080>

[7] Unity Asset Store, "Unity Asset Store Resmi Sitesi," Unity Technologies.

<https://assetstore.unity.com/packages/2d/gui/fantasy-wooden-gui-free-103811>

[8] Unity Asset Store, "Unity Asset Store Resmi Sitesi," Unity Technologies.

<https://assetstore.unity.com/packages/3d/characters/animals/free-stylized-bear-rpg-forest-animal-228910>

[9] Unity Asset Store, "Unity Asset Store Resmi Sitesi," Unity Technologies.

<https://assetstore.unity.com/packages/2d/textures-materials/nature/realistic-terrain-textures-free-279940>

[10] Unity Asset Store, "Unity Asset Store Resmi Sitesi," Unity Technologies.

<https://assetstore.unity.com/packages/2d/textures-materials/sky/allsky-free-10-sky-skybox-set-146014>

[11] UnityEngine.AI. Documentation, "Unity Oyun Motoru Belgeleri AI sınıfı," Unity Technologies <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/AI.NavMesh.html>

[12] Unity Asset Store, "Unity Asset Store Resmi Sitesi," Unity Technologies. <https://assetstore.unity.com>.

[13] Microsoft Docs, "C# Guide," Microsoft Corporation.

<https://learn.microsoft.com/en-us/dotnet/csharp/>

[14] Visual Studio Code Documentation, "Visual Studio Code Resmi Belgeleri," Microsoft Corporation. <https://code.visualstudio.com/docs>

# EKLER

Bu projenin yazılımının tüm kodları ve raporun docx belgesi https://github.com/Ldrsfurkan/HolyPolyScriptleri linkinde verilmiştir. Ayrıca geliştirilen oyunun bir çalıştırma videosu <https://www.youtube.com/watch?v=k9U8ACti290> linkinde verilmiştir.

# ÖZGEÇMİŞ

Furkan Gölemez 2002 yılında İzmir/Konakta doğdu. İlk ve ortaokul öğrenimini Vali Rahmi Bey İlkokulunda tamamladı. 2020 yılında 85.Yıl Anadolu Lisesi sayısal bölümünden mezun oldu. 2021 yılında Muğla Sıtkı Koçman Üniversitesi Teknoloji Fakültesi Bilişim Sistemleri Mühendisliği Bölümü’nü kazandı. Halen aynı bölümün 4. sınıf öğrencisidir.