

GAZİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ



BM495/ BM496 BİLGİSAYAR
PROJESİ

DÖNEM SONU RAPORU
Sihirli Ayna

Doç. Dr. Murat Yılmaz

191180758-Berat Berkay Erken
191180086-Ahmed Senih Yıldırım
191180008-Tuğba Akın

Kelime Sayısı: 5229

Haziran 2024

İNTİHAL BEYANI

Bu çalışmadaki tüm bilgilerin akademik kurallara ve etik davranışa uygun olarak alındığını ve sunulduğunu ve bu belgede alıntı yaptığımı belirttiğim yerler dışında sunduğum çalışmanın kendi çalışmam olduğunu, Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesinde belirtilen bilimsel araştırma ve yayın etiği ilkelerine uygun olduğunu beyan ederim.

Numara: 191180086, 191180008, 191180758

Ad Soyad: Ahmed Senih Yıldırım, Tuğba Akın, Berat Berkay Erken

Tarih: 21/06/2024

Ahmed Senih Yıldırım

Tuğba Akın

Berat Berkay Erken

İmza:



İmza:



İmza:



İÇİNDEKİLER TABLOSU

1. GİRİŞ	6
2. LİTERATÜR TARAMASI.....	7
2.1 Literatür Taramasının Kapsamı ve Yöntemi	7
2.2 Literatür Taramasının Bulguları.....	7
2.2.1 Turing Testi ile İlgili Çalışmalar	7
2.2.2 Sohbet Botları ve AI İletişimi ile İlgili Çalışmalar	7
2.3.3. Analitik Düşünme Testleri ile İlgili Çalışmalar	8
3. SRS.....	9
3.1 Gereksinimler	9
3.1.1 Gerekli Durum ve Modlar	9
3.1.2 Fonksiyonel Gereksinimler	9
3.1.3 Dış Arayüz Gereksinimleri.....	9
3.1.4 Dahili Arayüz Gereksinimleri	9
3.1.5 Dahili Veri Gereksinimleri.....	9
3.1.6 Performans Gereksinimleri	9
3.1.7 Diğer Gereksinimler	9
3.1.8 Gereksinimlerin Önceliği ve Kritikliği	9
3.2. Yazılım Kalite Faktörleri	9
3.2.1 Güvenilirlik	9
3.2.2 Güvenlik	10
3.2.3 Taşınabilirlik	10
3.2.4 Yeniden Kullanılabilirlik.....	10
3.2.5 Test Edilebilirlik.....	10
3.2.6 Esneklik.....	10
3.3. Tasarım ve Uygulama Kısıtları	10
3.3.1 Yazılım Kısıtları	10
3.3.2 Donanım Kısıtları	10
4.SDD.....	11
4.1 Kavramsal Model	11
4.2 Yazılım Yaşam Döngüsü İçerisinde SDD.....	11
4.3 Fonksiyonel Gereksinimler	11
4.4. Arayüz Gereksinimleri	11
4.5. Performans Gereksinimleri	11
4.6. Güvenlik ve Gizlilik	11
4.7 Yapısal Tasarım.....	11
4.8 Tasarım Paydaşları	11
4.9 Tasarım Bakış Açıları.....	12
4.10 Tasarım Öğeleri.....	12
4.11. Tasarım Gerekçeleri	12
4.12. Tasarım Bakış Açıları.....	12
5.STD	13
5.1. Test Planı.....	13
5.2 Test Senaryoları.....	14
6. KULLANILAN TEKNOLOJİLER.....	16
6.1. Uygulama Geliştirme ve Araçları	16
6.1.1. Unreal Engine.....	16
6.1.2. MetaHuman Creator	16
6.2. Kullanılan API ve Eklentiler	17

6.2.1. ChatGPT API	17
6.2.2. Quixel Bridge	18
6.2.3. Varest Plugin	18
6.2.4. Elevenlabs API.....	19
6.2.5. Runtime Audio Importer	19
6.2.6. Convai	20
6.2.7. Speech-to-Text	20
6.3. Uygulamanın Genel İşleyişi	20
6.3.1.Kullanıcı Girişi ve Ses Tanıma	20
6.3.2. Konuşma Tanıma ve Metne Dönüştürme.....	20
6.3.3. Metnin ChatGPT API'ye Gönderilmesi	21
6.3.4. API Yanıtlarının İşlenmesi ve Ses Dosyasına Dönüştürme	21
6.3.5. Ses Dosyasının Oynatılması.....	21
6.3.6. Yüz Animasyonları ve Karakter Mimikleri	21
6.3.7. Gerçek Zamanlı Etkileşim ve Kullanıcı Deneyimi	21
7. KULLANIM SENARYOLARI	22
7.1. Senaryo-1- Yeni Oyuna Başlamak	22
7.2. Senaryo-2- Metin Girişi Yoluyla İletişim Kurma	22
7.3. Senaryo-3- Ses Girişi Yoluyla İletişim Kurma	23
7.4. Senaryo-4- Sistem Hatalarını Ele Alma	24
7.5. Senaryo-5- Oturum Zaman Aşımı.....	25
8. SONUÇ	27
9. KAYNAKLAR.....	28

ŞEKİLLER TABLOSU

Şekil 1: Metahuman Creator	17
Şekil 2: GPT API Blueprint	18
Şekil 3: Elevenlabs API Blueprint	19
Şekil 4: Senaryo-1 Use Case Diyagramı	22
Şekil 4: Senaryo-2 Use Case Diyagramı	23
Şekil 6: Senaryo-3 Use Case Diyagramı	24
Şekil 7: Senaryo-4 Use Case Diyagramı	25
Şekil 8: Senaryo-5 Use Case Diyagramı	26

1. GİRİŞ

Bu rapor, "Sihirli Ayna" projesinin tüm aşamalarını kapsayan kapsamlı bir çalışmayı sunmaktadır. Proje, ziyaretçilere müze ve tarihi alanlarda daha etkileyici ve öğretici bir deneyim sunmayı hedefleyen interaktif bir yazılım geliştirmeyi amaçlamaktadır. Bu yazılım, tarihi figürlerle gerçek zamanlı etkileşim imkanı sağlayarak kullanıcıların tarihsel bilgilerini arttırmalarına yardımcı olmaktadır.

Projenin temel amacı, kullanıcıların sesli ve yazılı komutlar aracılığıyla tarihi figürlerle etkileşimde bulunabilecekleri bir platform oluşturmaktır. Bu amaç doğrultusunda, Unreal Engine, MetaHuman Creator, ChatGPT API, Elevenlabs API, Runtime Audio Importer ve Convai gibi ileri teknolojiler kullanılarak gerçekçi ve etkileyici karakterler oluşturulmuştur.

Rapor, projenin gelişim sürecini ve elde edilen çıktıları dört ana başlık altında toplamaktadır:

Literatür Taraması: Bu bölüm, interaktif sohbet oyunları ve yapay zeka tabanlı iletişim sistemleri konusundaki en son gelişmeleri ve trendleri derinlemesine inceleyerek projenin teorik temelini oluşturmaktadır. Turing Testi, sohbet botları ve analitik düşünme testleri gibi konular detaylı bir şekilde ele alınmaktadır.

Yazılım Gereksinimleri Spesifikasyonu (SRS): Bu bölüm, geliştirilen yazılımın fonksiyonel ve fonksiyonel olmayan gereksinimlerini detaylı bir şekilde belirleyerek, projenin kapsamını ve hedeflerini net bir şekilde çizmektedir. Kullanıcıların tarihi karakterlerle etkileşim kurabilmeleri için gerekli teknik ve işlevsel gereksinimler tanımlanmaktadır.

Yazılım Tasarım Dokümanı (SDD): Bu bölüm, yazılımın mimarisini, arayüzlerini, veri akışını ve kullanılan teknolojileri görsel ve yazılı olarak açıklamaktadır. Sistemin genel yapısı, modüller arasındaki etkileşimler ve veri akışları ayrıntılı bir şekilde ele alınmaktadır.

Yazılım Test Dokümanı (STD): Bu bölüm, yazılımın işlevselliğini ve performansını doğrulamak için uygulanan test senaryolarını, test verilerini, test sonuçlarını ve hata raporlarını içermektedir. Yazılımın doğru ve güvenilir çalışmasını sağlamak amacıyla yapılan test faaliyetleri detaylı olarak sunulmaktadır.

Ayrıca, raporun ilerleyen bölümlerinde aşağıdaki iki ana başlığa da detaylı şekilde yer verilecektir:

Kullanılan Teknolojiler: Bu bölümde, "Sihirli Ayna" projesinde kullanılan çeşitli ileri teknolojiler ayrıntılı olarak ele alınacaktır. Unreal Engine, MetaHuman Creator, ChatGPT API, Elevenlabs API, Runtime Audio Importer ve Convai gibi araçların nasıl kullanıldığı, entegrasyon süreçleri ve projeye sağladıkları katkılar detaylı bir şekilde açıklanacaktır.

Kullanım Senaryoları: Bu bölümde, "Sihirli Ayna" projesinin kullanım senaryoları ayrıntılı olarak ele alınacaktır. Kullanıcıların tarihi figürlerle nasıl etkileşimde bulunacakları, sesli ve yazılı komutların nasıl işleneceği ve sistemin nasıl çalışacağı senaryolar üzerinden açıklanacaktır. Senaryolar, kullanıcı deneyimini en üst düzeye çıkarmak için tasarlanmış çeşitli etkileşim akışlarını içerecektir.

Bu rapor, projenin tüm aşamalarında elde edilen bulguları, karşılaşılan zorlukları ve bu zorluklara bulunan çözümleri detaylı bir şekilde sunarak, "Sihirli Ayna" projesinin geliştirilme sürecine ışık tutmaktadır. Raporun sonunda ise, projenin gelecekteki potansiyel geliştirme alanlarına ve önerilere yer verilmektedir.

2. LİTERATÜR TARAMASI

Sihirli Ayna Projesi, ChatGPT API'sini kullanarak kullanıcıların tarihi karakterlerle interaktif sohbetler yapabileceği ve Turing Testi tabanlı bir oyun deneyimi sunmayı amaçlamaktadır. Bu bağlamda yapılan literatür taraması, Turing Testi, sohbet botları ve analitik düşünme testleri konularında mevcut çalışmaların incelenmesini içermektedir.

2.1 Literatür Taramasının Kapsamı ve Yöntemi

Kapsam: Literatür taraması üç ana konuya odaklanmıştır:

- Turing Testi ve bu testin uygulamaları
- Sohbet botları ve AI iletişimi
- Analitik düşünme testleri ve teknikleri

Yöntem: Google Scholar, PubMed, JSTOR ve diğer akademik veri tabanları kullanılarak ilgili akademik makaleler, kitaplar, tezler ve raporlar toplanmış ve analiz edilmiştir. Kaynaklar, projenin hedefleri doğrultusunda değerlendirildi ve özetlendi.

2.2 Literatür Taramasının Bulguları

2.2.1 Turing Testi ile İlgili Çalışmalar

- **Alan Turing'in Orijinal Çalışması:** Turing, 1950'deki makalesinde, bir makinenin insan gibi düşünebildiğini göstermek için bir test önerdi. Bu testte, bir insan değerlendirici, insan ve makine arasında yapılan yazılı bir iletişimden hangisinin insan olduğunu belirleyemezse, makine testi geçmiş kabul edilir.
- **ELIZA:** 1966'da Joseph Weizenbaum tarafından geliştirilen ELIZA, kullanıcıların girdilerini soru biçiminde yeniden ifade ederek yanıtlar üretir. ELIZA, insan-makine etkileşiminin erken bir örneğidir ve Turing Testi'ne benzer şekilde insanı yanıltmayı hedefler.[1,2,3]
- **Google Duplex:** 2018'de duyurulan Duplex, doğal dilde konuşmalar yürütür ve belirli görevleri yerine getirir. Bu teknoloji, Turing Testi'nin pratik bir uygulamasını temsil eder ve telefon görüşmeleri gibi gerçek dünya senaryolarında kullanılır. [4,5]

2.2.2 Sohbet Botları ve AI İletişimi ile İlgili Çalışmalar

- **Hello History:** GPT-4 teknolojisi ile çalışan bu uygulama, kullanıcıların tarihi figürlerle gerçekçi konuşmalar yapmasını sağlar. Bu, projenin hedeflediği tarihi karakterlerle interaktif sohbet deneyimine benzer bir uygulamadır.[6]
- **Marcus Aurelius AI:** Kullanıcılara Roma İmparatoru Marcus Aurelius ile mesajlaşma imkânı tanır. Bu yapay zeka, tarihi bir figürle etkileşimi simüle ederek, projenin hedeflediği deneyimi destekler. [7,8]

- **Character.AI:** Büyük dil modelleri kullanarak sanal karakterlerle gerçek zamanlı ve bağlamına uygun yanıtlar üretir. Hem gerçek hem de hayali karakterlerle kullanıcıların etkileşimde bulunmasını sağlar.[9]

2.3.3. Analitik Düşünme Testleri ile İlgili Çalışmalar

- **Wason'un Seçim Görevi:** Mantıksal çıkarımlar yapma yeteneklerini ölçen bu test, kullanıcıların analitik düşünme becerilerini geliştirmeyi hedefler. Projenin bilmeceler ve bulmacalar sunma amacına hizmet eder. [10]
- **Linda Problem:** Temsil heuristiği yanılığını gösteren bu problem, katılımcıların mantıksal düşünme süreçlerini analiz eder. Bu tür problemler, kullanıcıların analitik düşünme yeteneklerini geliştirmede kullanılabilir.[11]
- **Silogistik Akıl Yürütme Testi:** İki önermeden bir sonuç çıkarılmasını içeren bu test, mantıksal bilgi kullanımı ve doğru sonuca ulaşma yeteneğini değerlendirir. Projenin analitik düşünmeyi teşvik edici yapısıyla uyumludur.[12]

3. SRS

3.1 Gereksinimler

3.1.1 Gerekli Durum ve Modlar

Proje, karakter seçim ekranı, sohbet ekranı ve ayarlar ekranı olmak üzere üç ana modda çalışacaktır.

3.1.2 Fonksiyonel Gereksinimler

- Kullanıcılar, tarihi karakterlerle 3 boyutlu model üzerinden iletişim kurabilecektir.
- Kullanıcılar, karakterlerin sorduğu bilmeceleri çözmeye çalışacaktır.
- Sesli giriş ve çıkış özellikleri desteklenecektir.

3.1.3 Dış Arayüz Gereksinimleri

ChatGPT API'nin doğru çalışabilmesi için internet bağlantısı gereklidir.

3.1.4 Dahili Arayüz Gereksinimleri

Unreal Engine tabanlı grafik arayüzü ve oyun motoru ile entegre çalışacaktır.

3.1.5 Dahili Veri Gereksinimleri

Kullanıcı verileri tutulmayacak, ancak oyun senaryoları dahili olarak depolanacaktır.

3.1.6 Performans Gereksinimleri

Yanıt süreleri ve karakter etkileşimleri hızlı olmalıdır (örneğin, 2-4 saniye içinde).

3.1.7 Diğer Gereksinimler

Güvenlik ve gizlilik önlemleri alınacaktır.

3.1.8 Gereksinimlerin Önceliği ve Kritikliği

- Temel iletişim işlevselliği ve bilmecelerin doğru işlemesi yüksek önceliklidir.
- Güvenlik ve performans optimizasyonları orta önceliklidir.

3.2. Yazılım Kalite Faktörleri

3.2.1 Güvenilirlik

Kullanıcı gizliliği ve veri güvenliğine önem verilmektedir.

3.2.2 Güvenlik

Kaynak kodun korunması ve hassas bilgilerin güvenliğini sağlanması hedeflenmektedir.

3.2.3 Taşınabilirlik

Uygulama Windows işletim sistemi üzerinde çalışacaktır ve farklı sürümlerde sorunsuz çalışması beklenmektedir.

3.2.4 Yeniden Kullanılabilirlik

Yazılım kodu modüler yapıda olup yeni özelliklerin kolayca eklenmesine olanak tanır.

3.2.5 Test Edilebilirlik

Yazılım geniş test senaryolarına uygun şekilde tasarlanmıştır.

3.2.6 Esneklik

Yazılım, kullanıcı ihtiyaçlarına kolayca uyum sağlayabilecek esneklikte tasarlanmıştır.

3.3. Tasarım ve Uygulama Kısıtları

3.3.1 Yazılım Kısıtları

Uygulama yalnızca belirli programlama dilleri ve araçları kullanılarak geliştirilecektir.

3.3.2 Donanım Kısıtları

Uygulamanın çalışması için belirli donanım gereksinimleri bulunmaktadır.

4.SDD

4.1 Kavramsal Model

Kavramlar ve bağlam hakkında bilgiler içeren bu bölümde, yazılım tasarımı iki ana bileşene ayrılır:

- **ChatGPT API'si ile Sesli İşlem Bileşeni:** Kullanıcının sesli komutlarını anlar, cevap üretir ve cevabı iletir.
- **Kullanıcı Arayüzü:** Kullanıcı, tarihi karakterlerle etkileşimde bulunur ve karakterleri seçer.

4.2 Yazılım Yaşam Döngüsü İçerisinde SDD

Bu bölüm, yazılım geliştirme süreçlerini kapsar ve yazılımın genel yapısını, işleyişini ve bileşenler arasındaki ilişkileri detaylı bir şekilde ele alır. Performans gereksinimleri, güvenlik önlemleri, modüler yapı ve diğer önemli tasarım unsurları incelenir.

4.3 Fonksiyonel Gereksinimler

- Kullanıcılar, tarihi karakterlerle metin tabanlı veya sesli arayüz üzerinden iletişim kurabilir.
- Bilmece çözme ve sesli giriş/çıkış özellikleri bulunmaktadır.

4.4. Arayüz Gereksinimleri

- **Dış Arayüz:** ChatGPT API, internet bağlantısı gerektirir.
- **Dahili Arayüz:** Unity veya Unreal Engine tabanlı kullanıcı arayüzü ile oyun motoru entegrasyonu sağlanır.

4.5. Performans Gereksinimleri

- Karakter yanıt süresi, kullanıcı eylemlerine 2 saniye içinde cevap vermelidir.
- Karakter seçimi sonrası bilgi verme işlemi 4 saniye içinde başlamalıdır.

4.6. Güvenlik ve Gizlilik

Kullanıcı bilgileri güvenli bir şekilde işlenmeli ve gizlilik politikalarına uygun olarak korunmalıdır.

4.7 Yapısal Tasarım

Tasarım sürecini destekleyen iki ana kısım:

- **Backend:** ChatGPT API'sinin ve ses işleme işlemlerinin yönetimi.
- **Frontend:** Karakterlerin görselleri ve konuşma animasyonları.

4.8 Tasarım Paydaşları

Geliştiriciler, kullanıcılar ve ChatGPT API'sini sağlayan OpenAI, projenin paydaşlarıdır.

4.9 Tasarım Bakış Açıları

Kullanıcı dostu bir arayüz ve kullanıcıların tarihi karakterlerle etkileşimde bulunabileceği, analitik düşünme yeteneklerini geliştirebileceği bir ortam sağlanır.

4.10 Tasarım Öğeleri

Kullanıcı arayüzü, tarihi karakterler, bilmeceler ve sesli giriş seçeneği temel öğelerdir. Kullanıcı, arayüz üzerinden bir karakter seçer, tanıtımı dinler ve sohbet başlar.

4.11. Tasarım Gerekçeleri

Tasarımın amacı, kullanıcıların hem eğlenceli hem de zihin açıcı bir deneyim yaşamasını sağlamaktır.

4.12. Tasarım Bakış Açıları

- **Bağlam Bakış Açısı:** Sistem ve çevresi arasındaki ilişkileri, bağımlılıkları ve etkileşimleri tanımlar.
- **Kompozisyon Bakış Açısı:** Sistemin bileşenleri arasındaki ilişkileri detaylandırır.
- **Etkileşim Bakış Açısı:** Sistemin bileşenleri arasındaki etkileşimleri tanımlar.
- **Mantıksal Bakış Açısı:** Sistemin işlevsel ve yapısal öğelerini tanımlar.

5.STD

Genel Bakış

Sihirli Ayna projesi, müze ve tarihi alan ziyaretçilerine tarihi figürlerle etkileşimli bir deneyim sunan bir yazılımdır. Proje, ChatGPT API kullanılarak geliştirilmiş ve kullanıcıların tarihi karakterlerle konuşmalarını, bilmeceler çözmelerini sağlamaktadır. Bu belge, projenin test planını ve senaryolarını ayrıntılı olarak sunar, yazılımın doğru ve güvenilir çalışmasını sağlamak için yapılacak test faaliyetlerini tanımlar.

Test Yaklaşımı

Yazılımın belirlenen gereksinimlere uygun olup olmadığını doğrulamak amacıyla entegrasyon testi, sistem testi, kabul testi, fonksiyonel testler, performans testleri ve kullanılabilirlik testleri gibi farklı test türleri uygulanacaktır. Bu testler, yazılım geliştirme sürecinde hataları erken tespit etmeyi ve düzeltmeyi amaçlar

5.1. Test Planı

Test Edilecek Özellikler:

- **Tarihi Karakter Seçimi:** Kullanıcıların tarihi karakterleri doğru bir şekilde seçebilmesi ve bu karakterler hakkında bilgi alabilmesi.
- **Sohbet Fonksiyonu:** Kullanıcıların tarihi karakterlerle metin ve sesli olarak etkileşime girebilmesi.
- **Ses Girişi ve Çıkışı:** Kullanıcıların sesli komutlar verebilmesi ve yazılımın bu komutları doğru bir şekilde işleyerek yanıt verebilmesi.
- **ChatGPT API Entegrasyonu:** API'nin doğru çalışması ve kullanıcı sorularına uygun yanıtlar vermesi.
- **Bilmeceler ve Bulmacalar:** Tarihi karakterlerin bilmeceler sorması ve kullanıcıların bunları çözebilmesi.
- **Performans ve Yanıt Süresi:** Yazılımın performansının ve yanıt süresinin kullanıcı beklentilerini karşılaması.
- **Kullanıcı Arayüzü:** Kullanıcı dostu, kolay anlaşılır ve kullanılabilir olması.

Test Edilmeyecek Özellikler:

- **Arka Uç Veritabanı Performansı:** Üçüncü taraf hizmet tarafından yönetildiği için test edilmeyecek.
- **Özel Tarihi İçerik Doğruluğu:** Yazılımın işlevselliği odak noktası olduğundan, tarihsel doğruluk test edilmeyecek.
- **Grafik ve Animasyon Kalitesi:** Temel işlevsellik sağlandığı sürece test edilmeyecek.
- **Çoklu Dil Desteği:** Sağlanmamışsa test edilmeyecek.
- **Donanım Spesifik Optimizasyonlar:** Donanımların optimizasyonu test edilmeyecek.

Test Ortamı ve Araçları:

- **Ortam:** Windows 11 (64-bit), Intel i7 işlemci, 16GB RAM, NVIDIA GeForce RTX 2060 ekran kartı, yüksek hızlı internet bağlantısı.
- **Araçlar:** JUnit (birim testleri), Postman (API testleri), LoadRunner (performans testleri), Jira (test yönetimi ve hata takibi).

5.2 Test Senaryoları

Senaryo-1: Tarihi karakter seçimi.

- **Amaç:** Kullanıcıların tarihi karakterleri doğru seçebilmesini ve seçilen karakter hakkında bilgi alabilmesini doğrulamak.
- **Girişler:** Uygulamanın başlatılması, tarihi karakter seçim ekranına geçiş, karakter seçimi.
- **Beklenen Sonuçlar:** Kullanıcı doğru yönlendirilir ve karakter bilgileri eksiksiz görüntülenir.
- **Geçme/Kalma Kriterleri:** Karakter seçim ekranına sorunsuz erişim ve doğru bilgi görüntüleme.
- **Test Prosedürleri:** Uygulama başlatma, karakter seçim ekranına geçiş, karakter seçimi ve bilgi doğrulama.
- **Sonuç:** Test başarılı. Karakter bilgileri doğru ve eksiksiz görüntülendi.

Senaryo-2: Sesli komutlarla etkileşim.

- **Amaç:** Kullanıcıların tarihi karakterlerle sesli komutlar aracılığıyla etkileşim kurabilmesi ve doğru tepkiler alabilmesi.
- **Girişler:** Uygulamanın başlatılması, tarihi karakter seçimi, sesli komut kullanımı.
- **Beklenen Sonuçlar:** Karakter doğru ve bağlama uygun yanıtlar verir.
- **Geçme/Kalma Kriterleri:** Sesli komutlarla sorunsuz etkileşim ve anlamlı yanıtlar.
- **Test Prosedürleri:** Uygulama başlatma, karakter seçimi, sesli komutla etkileşim.
- **Sonuç:** Test başarılı. Karakter doğru ve anlamlı yanıtlar verdi.

Senaryo-3: Animasyonların senkronizasyonu.

- **Amaç:** Karakterlerin sesli yanıt verirken animasyonlarının doğru ve senkronize çalıştığını doğrulamak.
- **Girişler:** Uygulamanın başlatılması, tarihi karakter seçimi, sesli komut kullanımı.
- **Beklenen Sonuçlar:** Karakterlerin el ve ağız animasyonları doğru ve senkronize çalışır.
- **Geçme/Kalma Kriterleri:** Animasyonların doğru çalışması ve senkronizasyon.
- **Test Prosedürleri:** Uygulama başlatma, karakter seçimi, sesli komutla etkileşim ve animasyon doğrulama.

- **Sonuç:** Test başarılı. Animasyonlar doğru ve senkronize çalıştı.

Senaryo-4: Bilmece ve bulmacalar.

- **Amaç:** Tarihi karakterlerin bilmeceler sorması ve kullanıcıların bunları çözebilmesi.
- **Girişler:** Uygulamanın başlatılması, tarihi karakter seçimi, bilmece sorma.
- **Beklenen Sonuçlar:** Karakterler doğru bilmeceler sorar ve kullanıcıların yanıtlarını tanır.
- **Geçme/Kalma Kriterleri:** Doğru bilmece sorma ve yanıt tanıma.
- **Test Prosedürleri:** Uygulama başlatma, karakter seçimi, bilmece sorma ve yanıt doğrulama.
- **Sonuç:** Test başarılı. Bilmece ve bulmacalar doğru soruldu ve yanıtlar tanındı.

6. KULLANILAN TEKNOLOJİLER

Sihirli Ayna projesini geliştirmek için çeşitli ileri teknolojilerden faydalanılmıştır. Bu teknolojiler, projenin kullanıcılarına tarihi figürlerle interaktif ve sürükleyici bir deneyim sunmasını sağlamak üzere entegre edilmiştir. Projenin başarılı bir şekilde hayata geçebilmesi için kullanılan bu teknolojiler, doğal dil işleme, yüz animasyonları, ses sentezi, gerçek zamanlı veri işleme ve grafik motorları gibi birçok alanı kapsamaktadır. Her bir teknoloji, ziyaretçilerin tarihi figürlerle etkileşimde bulunurken hem eğlenceli hem de öğretici bir deneyim yaşamalarını garantilemek için özenle seçilmiştir. Bu kapsamlı ve detaylı teknolojik altyapı, "Sihirli Ayna" projesinin hedeflerine ulaşmasını ve kullanıcılarına benzersiz bir deneyim sunmasını sağlamaktadır. Aşağıda, projenin temel teknolojileri ve bu teknolojilerin nasıl kullanıldığına dair detaylı açıklamalar yer almaktadır.

6.1. Uygulama Geliştirme ve Araçları

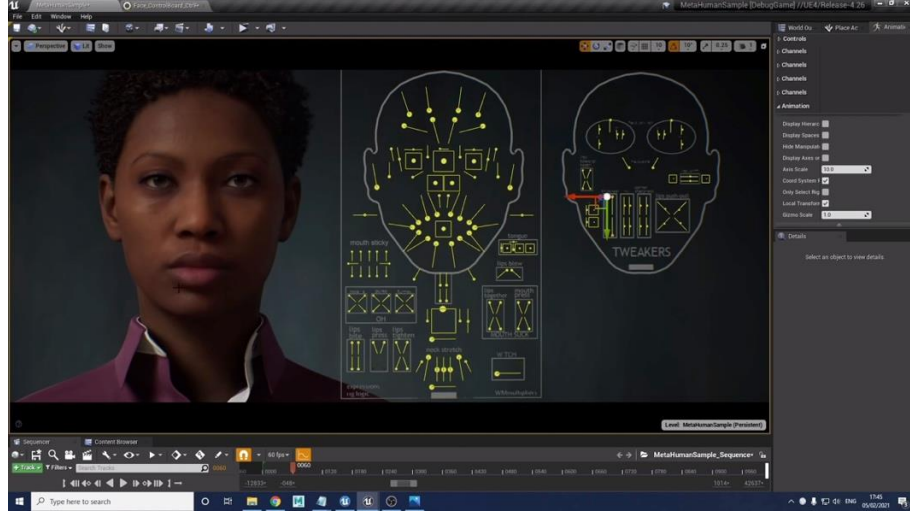
"Sihirli Ayna" projesinin geliştirilmesinde çeşitli yazılım geliştirme araçları ve ortamları kullanılmıştır. Bu araçlar, projenin farklı bileşenlerinin verimli bir şekilde entegre edilmesini ve çalışmasını sağlamıştır.

6.1.1. Unreal Engine

Unreal Engine, "Sihirli Ayna" projesinin temel grafik motorudur. Bu güçlü ve esnek oyun motoru, 3D modelleme, animasyon konusunda oldukça öne çıkan bir motordur.[13] Unreal Engine, tarihi figürlerin gerçekçi bir şekilde canlandırılmasını ve kullanıcı arayüzlerinin oluşturulmasını sağlar. Kullanıcıların tarihi figürlerle etkileşime geçmesini sağlayan arayüz, bu motor üzerinde inşa edilmiştir. Ayrıca, Unreal Engine'in Varest Plugin ile API entegrasyonlarını kolaylaştırması, proje geliştirme sürecini hızlandırmıştır.

6.1.2. MetaHuman Creator

MetaHuman Creator, Unreal Engine'in bir parçası olarak, yüksek kaliteli ve gerçekçi insan modelleri oluşturmak için kullanılır. Sihirli Ayna projesinde, tarihi figürlerin detaylı ve gerçekçi 3D modellerini oluşturmak için bu araç kullanılmıştır. MetaHuman Creator, karakterlerin fiziksel özelliklerini, yüz ifadelerini ve hareketlerini son derece ayrıntılı bir şekilde tasarlamaya olanak tanır. Bu teknoloji, karakterlerin inandırıcı ve çekici bir şekilde görünmesini sağlar, bu da kullanıcıların tarihi figürlerle daha derinlemesine ve etkileyici bir şekilde etkileşime geçmelerine yardımcı olur.[14]



Şekil 1: Metahuman Creator

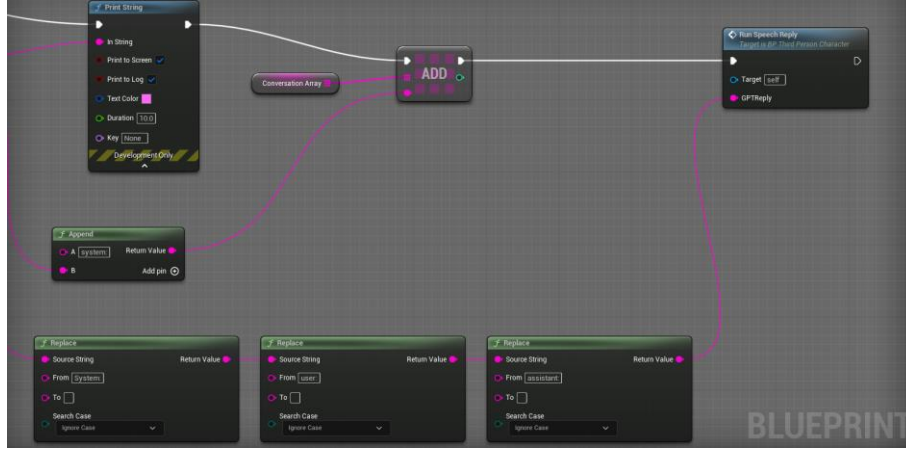
6.2. Kullanılan API ve Eklentiler

6.2.1. ChatGPT API

ChatGPT API, "Sihirli Ayna" projesinin temel interaktif bileşenidir. Bu API, doğal dil işleme (NLP) yetenekleri ile kullanıcıların tarihi figürlerle gerçekçi sohbetler yapabilmesini sağlar. Kullanıcı girdilerini işleyerek, anlamlı ve ilgili yanıtlar oluşturur. Bu sayede ziyaretçiler, tarihi figürlerle etkileyici ve eğitici bir diyalog kurabilirler. ChatGPT API, kullanıcıların sorularını ve tepkilerini analiz ederek, onlara uygun ve tarihi bilgilerle dolu yanıtlar verir. API'nin yüksek doğruluk oranı, esnek entegrasyon kabiliyetleri ve geniş dil desteği, projenin başarısının temel taşlarıdır.

ChatGPT API'sinin bağlanması tamamlandığında ortaya çıkan önemli bir sorun, API'nin sohbet geçmişi oluşturmaması ve her yanıtın bir öncekinden bağımsız olmasıdır. Bu durum, daha önce verilen yanıtların hatırlanmasını ve bağlamın korunmasını zorlaştırır. Bu sorunun üstesinden gelmek için, kullanıcının verdiği her cevabı ve ChatGPT'den gelen her yanıtı etiketleyerek, her yeni soruda bu etiketli geçmişi mesajın üst kısmında yeniden gönderiyoruz. Bu sayede, API önceki sohbet bağlamı hakkında bilgi sahibi olabiliyor ve daha tutarlı ve ilgili yanıtlar üretebiliyor.

Teknik olarak, bu süreç şu şekilde gerçekleşir: Kullanıcıdan gelen her yeni giriş ve API'den dönen her yanıt bir dizi olarak saklanır. Bu diziler, her seferinde mesajın başına eklenir. Bu işlem, Unreal Engine üzerinde Blueprint sistemi kullanılarak gerçekleştirilir.



Şekil 2: GPT API Blueprint

Blueprint sisteminde, her yeni kullanıcı girişi ve API yanıtı bir diziye dönüştürülür ve ardından bu diziler birleştirilir. Yukarıdaki görselde, "Append" düğümü, yeni girişleri mevcut sohbete eklemek için kullanılır. Bu işlem, kullanıcının ve API'nın tüm geçmiş mesajlarını içeren bir metin dizisi oluşturur. Daha sonra, bu dizin ChatGPT API'ye gönderilir, böylece API önceki bağlamı dikkate alarak yanıt verebilir.

Ayrıca, görselde gösterilen "Replace" düğümleri, kullanıcı ve sistem etiketlerini doğru bir şekilde güncelleyerek her mesajın bağlamını korur. Bu, "System", "User" ve "Assistant" etiketlerinin her biri için ayrı ayrı yapılar. Son olarak, "Run Speech Reply" düğümü, bu metin dizisini alır ve API çağrısını gerçekleştirir.

Bu teknik yapı, sohbetin bağlamını korumak ve kullanıcıya daha tutarlı yanıtlar sunmak için kritik öneme sahiptir. Bu sayede, "Sihirli Ayna" projesi, kullanıcıların tarihi figürlerle etkileşimlerinde daha derin ve anlamlı bir deneyim yaşamalarını sağlar.

6.2.2. Quixel Bridge

Quixel Bridge, 3D varlıklar ve yüzeyler için bir yönetim ve indirme platformudur. Quixel'in MegaScans kütüphanesine erişim sağlanarak yüksek kaliteli, fotogerçekçi varlıkları projelerine dahil etmelerini kolaylaştırır. Yüksek kaliteli varlıkları sıfırdan oluşturmak yerine Quixel Bridge'den indirerek zamandan tasarruf edebilmektedir.[15] "Sihirli Ayna" projesinde de nanite kalitede 3D varlıklar ve yüzeyler kullanılmıştır.

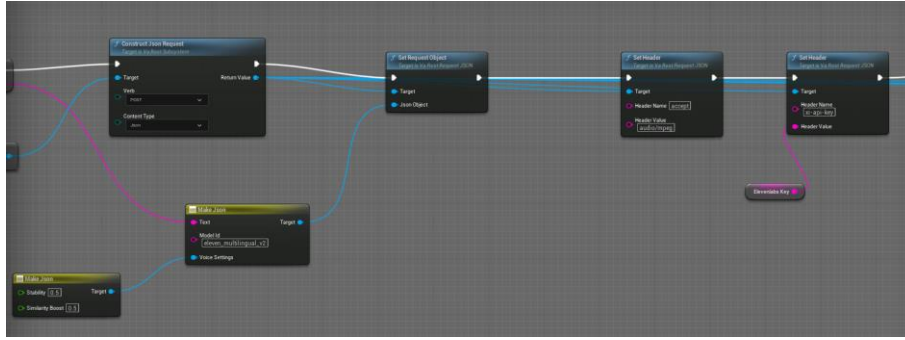
6.2.3. Varest Plugin

Varest Plugin, Unreal Engine ile RESTful API'lerin entegrasyonunu kolaylaştıran bir araçtır. "Sihirli Ayna" projesinde, bu plugin, ChatGPT API ve diğer web hizmetleriyle iletişimi yönetmek için kullanılır. Plugin, HTTP isteklerini basit ve verimli bir şekilde göndermeye olanak tanır. Bu sayede, proje içerisindeki veri alışverişi hızlı ve güvenilir bir şekilde gerçekleştirilir. Varest Plugin'in kolay yapılandırılabilir olması, geniş dokümantasyon desteği ve yüksek performansı, projenin API entegrasyon ihtiyaçlarını karşılar.[16]

6.2.4. Elevenlabs API

Elevenlabs API, "Sihirli Ayna" projesinde ses sentezi ve doğal konuşma üretimi için kullanılır. Bu API, tarihi figürlerin sesli yanıtlar verebilmesini sağlar. Metin girdilerini yüksek kaliteli ses dosyalarına dönüştürür. Bu sayede, ziyaretçiler tarihi figürlerle sadece yazılı olarak değil, aynı zamanda sesli olarak da etkileşimde bulunabilirler. Elevenlabs API'nin gerçekçi ve doğal ses sentezi, çoklu dil ve aksan desteği, esnek kullanım seçenekleri, kullanıcı deneyimini zenginleştirir ve tarihi figürlerin daha inandırıcı bir şekilde iletişim kurmasını sağlar. [17]

Elevenlabs API'sinin sağladığı önemli avantajlardan biri, birden çok dilde ve çeşitli seslerle destek sunabilmesidir. Bu özellik sayesinde, "Sihirli Ayna" projesindeki her tarihi karakter için birbirinden farklı ve karaktere özgü sesler kullanabiliyoruz. Süreç şu şekilde işler: Konuşulmak istenen metni ve kullanmak istediğimiz sesin benzersiz kimlik numarasını (unique ID) API'ye göndeririz. API, bu bilgiler doğrultusunda metni işler ve bize bir ses dosyası olarak yanıt döner. Böylece, her karakterin sesi, kullanıcıya daha gerçekçi ve kişiselleştirilmiş bir deneyim sunmak üzere özenle seçilip üretilmiş olur. Bu özellik, karakterlerin etkileşimini daha inandırıcı ve etkileyici kılar.



Şekil 3: Elevenlabs API Blueprint

6.2.5. Runtime Audio Importer

Runtime Audio Importer, gerçek zamanlı ses dosyası işleme ve oynatma işlemleri için kullanılır. "Sihirli Ayna" projesinde, bu araç sayesinde ses dosyaları dinamik olarak yüklenebilir ve oynatılabilir. Bu teknoloji, uygulama çalışırken ses dosyalarını yüklemeye ve entegre etmeye olanak tanır.[18] Bu, projeye esneklik ve dinamik içerik ekleme kabiliyeti sağlar. Runtime Audio Importer'ın hızlı işlem yapabilme, geniş format desteği ve kolay entegrasyon imkanı, projenin ses yönetimini etkin bir şekilde gerçekleştirmesine yardımcı olur. Proje üzerinde Runtime Audio Importer'ı, Elevenlabs API'ndan dönen MP3 formatındaki sesi oynatmak için kullanıyoruz. Bu süreç, kullanıcının etkileşimi sırasında gerçek zamanlı olarak işlenir. İlk olarak, kullanıcıdan alınan metin ve kullanmak istediğimiz sesin unique ID'si, Elevenlabs API'na gönderilir. API, metni işleyerek uygun ses dosyasını üretir ve bu dosyayı MP3 formatında geri döner.

Runtime Audio Importer, bu MP3 dosyasını alır ve ses dosyasını projeye dinamik olarak entegre eder. Böylece, tarihi karakterlerin sesleri gerçek zamanlı olarak oluşturulur ve kullanıcıya anında sunulur. Bu işlemler, Unreal Engine üzerinde gerçekleştirilen ses yönetim sistemi aracılığıyla yapılır.

Bu süreç, kullanıcı deneyimini daha etkileyici hale getirmek için oldukça kritiktir. Kullanıcıların söyledikleri her şey anında işlenir ve tarihi figürler bu girdilere uygun tepkiler verir. Runtime Audio Importer, bu süreci kesintisiz ve verimli bir şekilde yönetir, böylece kullanıcılar doğal ve akıcı bir etkileşim yaşarlar. Sonuç olarak, "Sihirli Ayna" projesi, ziyaretçilere tarihi figürlerle zenginleştirilmiş, gerçek zamanlı bir sesli iletişim deneyimi sunar.

6.2.6. Convai

Convai, yüz animasyonları ve karakter mimikleri için kullanılır. "Sihirli Ayna" projesinde, tarihi figürlerin daha gerçekçi ve etkileşimli olmasını sağlamak için bu teknoloji kullanılır. Convai, Metahumanlar için konuşmalarına uygun yüz animasyonları üretir. Bu, tarihi figürlerin daha inandırıcı ve doğal görünmesini sağlar.

6.2.7. Speech-to-Text

Sihirli Ayna projesinde konuşma tanıma teknolojisi, kullanıcıların sesli komutlarını metin girdilerine dönüştürmek için kullanılır. Bu teknoloji, kullanıcıların tarihi figürlerle doğal ve etkileyici bir şekilde etkileşime geçmesini sağlar. Konuşma tanıma süreci, ziyaretçilerin söylediklerini gerçek zamanlı olarak analiz eder ve bu konuşmaları metne dönüştürür.

Konuşma tanıma süreci, ziyaretçinin konuşmasının kaydedilmesiyle başlar. Bu işlem, Unreal Engine üzerinde oluşturulan bir ses tanıma modülü aracılığıyla gerçekleştirilir. Ziyaretçi bir komut verdiğinde, ses tanıma modülü bu komutu yakalar ve dijital bir ses dalgasına dönüştürür. Bu ses dalgası daha sonra işlenmek üzere konuşma tanıma algoritmalarına gönderilir.

Ses dalgası, konuşma tanıma algoritmaları tarafından analiz edilir ve metne dönüştürülür. Unreal Engine'de kullanılan bu modül, yüksek doğruluk oranına sahip olup, çoklu dil desteği sunar. Bu özellikler, projenin farklı dillerdeki ziyaretçiler tarafından kullanılabilmesini sağlar. Gerçek zamanlı işlem yapabilme yeteneği sayesinde, ziyaretçiler söyledikleri her şeyi anında metin olarak görebilir ve bu metinler anında işlenir.[19]

6.3. Uygulamanın Genel İşleyişi

6.3.1. Kullanıcı Girişi ve Ses Tanıma

Uygulamanın ilk aşaması, kullanıcının sesli komutlarını tanımadır. Ziyaretçi, uygulamayla etkileşime geçmek için konuşmaya başlar. Bu sesli komut, Unreal Engine üzerinde oluşturulan bir ses tanıma modülü aracılığıyla yakalanır. Ses tanıma modülü, ziyaretçinin konuşmasını dijital bir ses dalgasına dönüştürür. Bu işlem, "Start Speech Recognition" ile başlatılır ve "Capture Audio" ile ses dalgası yakalanır.

6.3.2. Konuşma Tanıma ve Metne Dönüştürme

Yakalanan ses dalgası, konuşma tanıma algoritmaları tarafından analiz edilerek metne dönüştürülür. Unreal Engine'de kullanılan bu modül, yüksek doğruluk oranına sahip olup, çoklu dil desteği sunar. Gerçek zamanlı işlem yapabilme yeteneği sayesinde, ziyaretçiler söyledikleri her şeyi anında metin olarak görebilirler. Bu süreçte, "Process Audio Data" düğmesi ses

verilerini işler ve "OnRecognitionFinished_Event" düğmesi tanıma sürecinin tamamlandığını belirtir.

6.3.3. Metnin ChatGPT API'ye Gönderilmesi

Metne dönüştürülen ses verileri, ChatGPT API'ye gönderilir. API, bu metinleri analiz ederek uygun yanıtları üretir. ChatGPT API, doğal dil işleme (NLP) yetenekleri sayesinde kullanıcı girdilerini anlamlı ve ilgili yanıtlarla karşılar. Bu süreç, ziyaretçilerin tarihi figürlerle gerçekçi ve bilgi dolu diyaloglar kurmasını sağlar. Metin verileri, API'ye gönderilmeden önce "Append" ve "Replace" düğmeleri kullanılarak geçmiş sohbet bağlamı ile birlikte yapılandırılır.

6.3.4. API Yanıtlarının İşlenmesi ve Ses Dosyasına Dönüştürme

ChatGPT API'den dönen yanıtlar, Elevenlabs API'ye gönderilerek ses dosyalarına dönüştürülür. Elevenlabs API, metin girdilerini yüksek kaliteli ses dosyalarına dönüştürme yeteneğine sahiptir. Bu işlem sırasında, konuşulmak istenen metin ve kullanmak istediğimiz sesin unique ID'si API'ye iletilir. API, bu bilgileri kullanarak uygun ses dosyasını üretir ve MP3 formatında geri döner.

6.3.5. Ses Dosyasının Oynatılması

API'den dönen MP3 formatındaki ses dosyası, Runtime Audio Importer aracılığıyla projeye dinamik olarak entegre edilir. Runtime Audio Importer, bu ses dosyasını alır ve gerçek zamanlı olarak oynatır. Bu aşama, tarihi figürlerin gerçek zamanlı olarak kullanıcıya sesli yanıt vermesini sağlar. Unreal Engine üzerinde oluşturulan bu ses yönetim sistemi, kullanıcıya doğal ve akıcı bir etkileşim deneyimi sunar.

6.3.6. Yüz Animasyonları ve Karakter Mimikleri

Sesli yanıtların yanı sıra, tarihi figürlerin yüz animasyonları ve mimikleri de Convai teknolojisi kullanılarak oluşturulur. Bu teknoloji, karakterlerin konuşmalarına uygun yüz animasyonları üretir ve bu animasyonlar gerçek zamanlı olarak ekranda gösterilir. Convai'nin yüksek doğrulukta yüz tanıma ve animasyon üretme yetenekleri, karakterlerin daha inandırıcı ve etkileyici görünmesini sağlar.

6.3.7. Gerçek Zamanlı Etkileşim ve Kullanıcı Deneyimi

Tüm bu aşamalar, ziyaretçilere kesintisiz ve etkileyici bir etkileşim deneyimi sunmak üzere tasarlanmış ve birleştirilmiştir. Ziyaretçiler, sesli komutlarını vererek tarihi figürlerle doğal bir diyalog kurabilirler. Gerçek zamanlı ses tanıma, metne dönüştürme, yanıt üretme ve sesli yanıt oynatma süreçleri, kullanıcı deneyimini üst seviyeye çıkarır. Ayrıca, çoklu dil desteği ve yüksek doğruluk oranı, projenin farklı dillerdeki ziyaretçiler tarafından rahatlıkla kullanılabilmesini sağlar. Sonuç olarak, "Sihirli Ayna" projesi, kullanıcıların tarihi figürlerle zengin ve anlamlı bir şekilde iletişim kurmalarına olanak tanır. Projenin her aşaması, ziyaretçilere unutulmaz bir deneyim sunmak için özenle tasarlanmış ve optimize edilmiştir. Bu teknolojik altyapı, projenin hedeflerine ulaşmasını ve kullanıcı memnuniyetini en üst düzeye çıkarmasını sağlar.

7. KULLANIM SENARYOLARI

7.1. Senaryo-1- Yeni Oyuna Başlamak

Aktörler: Oyuncu, Oyun Sistemi, ChatGPT, Oyun Karakteri

Önkoşullar:

- Oyuncu oyuna başarıyla giriş yaptı.
- Oyun oyuncunun cihazında yüklü ve çalışıyor.

Ana akış:

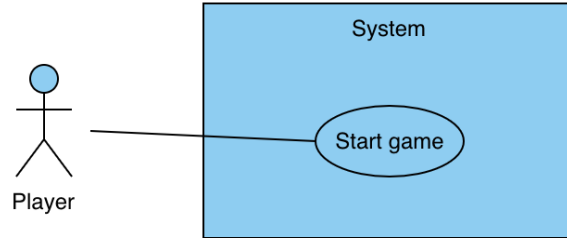
1. Oyuncu ana menüden bir karakter seçer.
2. Oyun sistemi sohbet penceresini ve ekrandaki oyun karakterini başlatır.
3. Oyun sistemi, dudak senkronizasyonu animasyonlarına sahip metin-konuşma modülünü kullanarak oyun karakterinden bir hoş geldiniz mesajı görüntüler.
4. Oyuncu artık sohbet penceresindeki metin girişini veya "T" tuşunu basılı tutarak ses girişini kullanarak etkileşime geçmeye hazırdır.

Alternatif Akışlar:

- Oyuncunun mikrofonu algılanmazsa oyun sistemi, oyuncudan mikrofonu etkinleştirmesini veya bunun yerine metin girişini kullanmasını ister.

Son koşullar:

- Oyun oturumu aktiftir ve oyuncu, oyun karakteriyle etkileşime girmeye başlayabilir.



Şekil 4: Senaryo-1 Use Case Diyagramı

7.2. Senaryo-2- Metin Girişi Yoluyla İletişim Kurma

Aktörler: Oyuncu, Oyun Sistemi, ChatGPT, Oyun Karakteri

Önkoşullar:

- Bir oyun oturumu etkindir.
- Ekranda sohbet penceresi ve karakter görünür.

Ana akış:

1. Oyuncu sohbet penceresine bir mesaj yazar ve "Enter" tuşuna basar.
2. Oyun sistemi metin girişini yakalar ve bunu bir istek olarak GPT API'sine gönderir.
3. GPT API girişi işler ve bir yanıt döndürür.

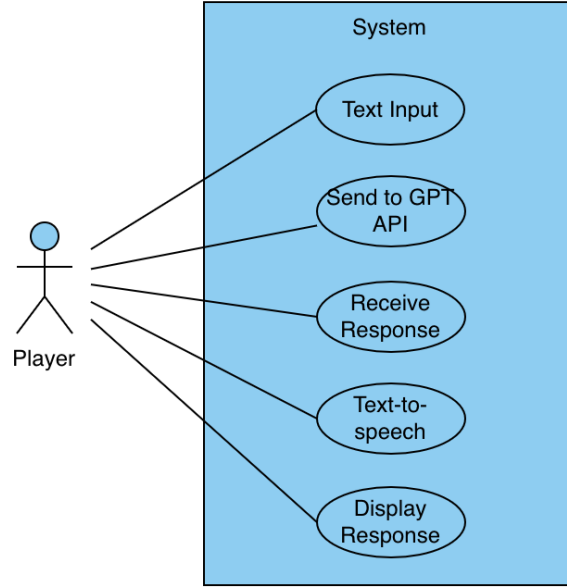
4. Oyun sistemi, yanıt karakterin dudak senkronizasyonu animasyonlarıyla birleştirir ve yanıtı seslendirmek için text-to-speech modülünü kullanır.
5. Oyun karakteri oyuncuya yanıt iletir.

Alternatif Akışlar:

- GPT API'si yanıt vermiyorsa oyun sistemi bir hata mesajı görüntüler ve oynatıcının daha sonra tekrar denemesini ister.

Son koşullar:

- Oyuncu, oyun karakterinden bir yanıt alır ve etkileşim döngüsü devam eder.



Şekil 4: Senaryo-2 Use Case Diyagramı

7.3. Senaryo-3- Ses Girişi Yoluyla İletişim Kurma

Aktörler: Oyuncu, Oyun Sistemi, Konuşmayı Metne Dönüştürme Modülü, ChatGPT, Oyun Karakteri

Önkoşullar:

- Bir oyun oturumu etkindir.
- Oynatıcının çalışan bir mikrofonu vardır.

Ana akış:

1. Oyuncu “T” tuşuna basıp basılı tutar ve Türkçe olarak bir şey söyler.
2. Oyun sistemi ses girişini yakalar ve bunu metne dönüştürmek için çok dilli konuşmayı metne dönüştürme modülünü kullanır.
3. Dönüştürülen metin GPT API'sine istek olarak gönderilir.
4. GPT API girişi işler ve bir yanıt döndürür.

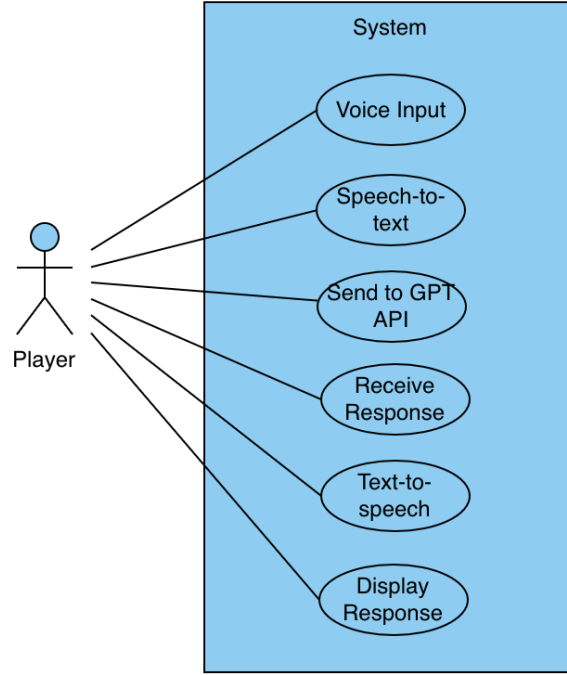
5. Oyun sistemi, yanıt karakterin dudak senkronizasyonu animasyonlarıyla birleştirir ve yanıtı seslendirmek için metinden konuşmaya modülünü kullanır.
6. Oyun karakteri yanıtı oyuncuya iletir.

Alternatif Akışlar:

- Konuşmayı metne dönüştürme modülü girişi tanıyamazsa, oyun sistemi oyuncudan mesajı tekrarlamasını ister.

Son koşullar:

- Oyuncu, oyun karakterinden bir yanıt alır ve etkileşim döngüsü devam eder.



Şekil 6: Senaryo-3 Use Case Diyagramı

7.4. Senaryo-4- Sistem Hatalarını Ele Alma

Aktörler: Oyuncu, Oyun Sistemi

Önkoşullar:

- Bir oyun oturumu etkindir.
- Oyun sırasında beklenmeyen bir hata meydana gelir.

Ana akış:

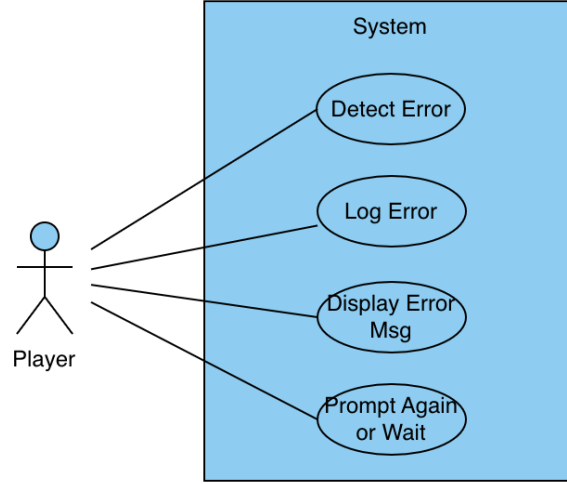
1. Bir hata oluştuğunda (örn. ağ hatası, API hatası vb.).
2. Oyun sistemi hatayı tespit eder ve günlüğe kaydeder.
3. Oyun sistemi oyuncuya bir hata mesajı görüntüler.
4. Oyuncu eylemi yeniden denemeyi veya ana menüye dönmeyi seçebilir.

Alternatif Akışlar:

- Hata otomatik olarak çözülürse oyun sistemi oyuncuyu bilgilendirir ve normal çalışmasına devam eder.
- Oyuncu yeniden denemeyi seçerse oyun sistemi eylemi yeniden dener.

Son koşullar:

- Hata ya çözülür ya da oyuncu bilgilendirilir ve bir sonraki adıma karar verebilir.



Şekil 7: Senaryo-4 Use Case Diyagramı

7.5. Senaryo-5- Oturum Zaman Aşımı

Aktörler: Oyuncu, Oyun Sistemi

Önkoşullar:

- Bir oyun oturumu etkindir.
- Oyuncu önceden belirlenen bir süre boyunca oyunla etkileşime girmedir.

Ana akış:

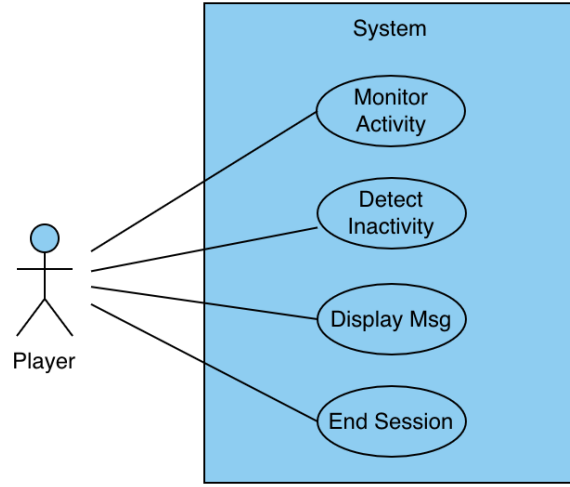
1. Oyun sistemi, oyun oturumu sırasında oyuncu etkinliğini izler.
2. Önceden tanımlanmış bir süre boyunca (örn. 10 dakika) oynatıcıdan herhangi bir etkileşim algılanmadı.
3. Oyun sistemi, hareketsizlik nedeniyle oturumun zaman aşımına uğrayacağını belirten bir uyarı mesajı görüntüler.
4. Oyuncu kısa bir ek süre içinde yanıt vermezse oyun sistemi oturumu otomatik olarak sonlandırır.
5. Oyun sistemi oyuncunun ilerlemesini kaydeder ve ana menüye döner.

Alternatif Akışlar:

- Oyuncunun ek süre içinde oyunla etkileşime girmesi durumunda oyun oturumu normal şekilde devam eder ve hareketsizlik zamanlayıcısı sıfırlanır.

Son koşullar:

- Oyun oturumu, hareketsizlik nedeniyle sona erer veya oyuncunun zamanında tepki vermesi durumunda devam eder.



Şekil 8: Senaryo-5 Use Case Diyagramı

8. SONUÇ

"Sihirli Ayna" projesi, müze ve tarihi alan ziyaretçilerine tarihi figürlerle etkileşimli bir deneyim sunmayı amaçlayan yenilikçi bir yazılım geliştirme çalışmasıdır. Proje, kullanıcıların sesli ve yazılı komutlar aracılığıyla tarihi figürlerle gerçek zamanlı sohbetler yapabilmesini ve bilmece çözerek analitik düşünme becerilerini geliştirebilmesini sağlamaktadır. Bu raporda, projenin tüm aşamaları detaylı bir şekilde ele alınmış ve kullanılan teknolojiler, yazılım gereksinimleri, tasarım detayları ve test süreçleri kapsamlı olarak sunulmuştur.

Projenin gerçekleştirilmesinde Unreal Engine, MetaHuman Creator, ChatGPT API, Elevenlabs API, Runtime Audio Importer ve Convai gibi ileri teknolojiler kullanılmıştır. Bu teknolojiler sayesinde tarihi figürler gerçekçi bir şekilde canlandırılmış ve kullanıcılarla etkileşimde bulunmaları sağlanmıştır. Ayrıca, projenin teorik temelleri ve literatür taraması, interaktif sohbet oyunları ve yapay zeka tabanlı iletişim sistemleri konusundaki en son gelişmeleri ve trendleri içermektedir.

"Sihirli Ayna" projesinin başarıyla tamamlanması, kullanıcıların tarihi figürlerle zengin ve öğretici bir şekilde etkileşim kurmalarını sağlayarak müze deneyimlerini daha ilgi çekici hale getirmiştir. Projenin gereksinimlerine uygun olarak yazılımın fonksiyonel ve fonksiyonel olmayan gereksinimleri belirlenmiş, mimari ve tasarım detayları açıklanmış ve kapsamlı test süreçleri uygulanarak yazılımın doğruluğu ve güvenilirliği sağlanmıştır.

Sonuç olarak, "Sihirli Ayna" projesi, tarihi figürlerle yapılan interaktif sohbetler ve bilmece aracılığıyla kullanıcıların tarihsel bilgilerini arttırmayı, analitik düşünme becerilerini geliştirmeyi ve müze ziyaretlerini daha etkileyici hale getirmeyi başarmıştır. Gelecekte, projenin farklı tarihi figürlerle zenginleştirilmesi ve yeni etkileşim senaryolarının eklenmesiyle kullanıcı deneyiminin daha da iyileştirilmesi mümkündür. Bu bağlamda, "Sihirli Ayna" projesi, müze ve tarihi alanlarda interaktif deneyimlerin geleceği için önemli bir adım olarak değerlendirilmektedir.

9. KAYNAKLAR

1. Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45.
2. Jarow, O. (2023). How the first chatbot predicted the dangers of AI more than 50 years ago. [Eriřim Tarihi: 30 Ekim 2023], <https://www.vox.com/future-perfect/23617185/ai-chatbots-eliza-chatgpt-bing-sydney-artificial-intelligence-history>
3. Wikipedia contributors. (2023). ELIZA — Wikipedia, The Free Encyclopedia. [Eriřim Tarihi 30 Ekim 2023], <https://en.wikipedia.org/w/index.php?title=ELIZA&oldid=1179533368>
4. Leviathan, Y., & Matias, Y. (2018). Google Duplex: An AI system for accomplishing real-world tasks over the phone.
5. O’Leary, D. E. (2019). GOOGLE’S Duplex: Pretending to be human. *Intelligent Systems in Accounting, Finance and Management*, 26(1), 46–53.
6. "Hello History AI." [Eriřim Tarihi: 30 Ekim 2023], <https://www.hellohistory.ai/?via=topaitools>
7. "Intellectual Giants: Marcus Aurelius." [Eriřim Tarihi: 30 Ekim 2023], <https://fs.blog/intellectual-giants/marcus-aurelius/>
8. "Marcus Aurelius AI." [Eriřim Tarihi: 30 Ekim 2023], <https://marcusaurelius.ai/>
9. "Character AI Beta." [Eriřim Tarihi: 30 Ekim 2023], <https://beta.character.ai/?via=topaitools>
10. Wikimedia Foundation. (2023, Ekim 20). Wason Selection Task. Wikipedia. https://en.wikipedia.org/wiki/Wason_selection_task
11. Olszewski, A. (2020). Linda problem – the tame solution in question. *Analecta Cracoviensia*, 51, 209-217. doi:10.15633/acr.3641.
12. Evans, J. St. B. T. (2002). Logic and human reasoning: An assessment of the deduction paradigm. *Psychological Bulletin*, 128(6), 978-996.
13. Wikipedia contributors. (2023). Unreal Engine. Wikipedia. https://tr.wikipedia.org/wiki/Unreal_Engine
14. Epic Games. (2023). MetaHuman Documentation. [Eriřim Tarihi: 14 Haziran 2024], <https://dev.epicgames.com/documentation/en-us/metahuman/metahuman-documentation>
15. Quixel. (2023). Quixel Bridge. [Eriřim Tarihi: 14 Haziran 2024], <https://quixel.com/bridge>
16. UFNA. (2023). VaRest UE4 Plugin. [Eriřim Tarihi: 14 Haziran 2024], <https://ufna.notion.site/VaRest-UE4-Plugin-40b98c54fc184033b60a42e0e4753536>
17. Eleven Labs. (2023). Text to Speech API Reference. [Eriřim Tarihi: 19 Haziran 2024], <https://elevenlabs.io/docs/api-reference/text-to-speech>
18. Unreal Engine Marketplace. (2023). Runtime Audio Importer. [Eriřim Tarihi: 19 Haziran 2024], <https://www.unrealengine.com/marketplace/en-US/product/runtime-audio-importer>
19. Unreal Engine Marketplace. (2023). Runtime Speech Recognizer. [Eriřim Tarihi: : 19 Haziran 2024], <https://www.unrealengine.com/marketplace/en-US/product/runtime9speech-recognizer>