



**T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ARTIRILMIŞ GERÇEKLIK VE COĞRAFI BİLGİ SİSTEMLERİ
YAKLAŞIMLARI KULLANILARAK TARİHİ ESER
RESTORASYONU İÇİN MOBİL UYGULAMA GELİŞTİRME**

MURAT EREN SARI

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN
DR. ÖĞR. ÜYESİ METİN TOZ**

DÜZCE, 2020

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**Artırılmış Gerçeklik ve Coğrafi Bilgi Sistemleri Yaklaşımları
Kullanılarak Tarihi Eser Restorasyonu İçin Mobil Uygulama
Geliştirme**

Murat Eren SARI tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Dr. Öğr. Üyesi Metin Toz

Düzce Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Metin Toz

Düzce Üniversitesi

Dr. Öğr. Üyesi Serdar Biroğul

Düzce Üniversitesi

Dr. Öğr. Üyesi Selçuk Kızır

Kocaeli Üniversitesi

Tez Savunma Tarihi: 10/07/2020

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

10 Temmuz 2020

MURAT EREN SARI

TEŐEKKÜR

Yüksek Lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Dr. Öğr. Üyesi Metin TOZ'a en içten dileklerle teşekkür ederim.

Tez çalışmamdaki yardımlarından dolayı 2 Numaralı Kültür Varlıklarını Koruma Bölge Kuruluna teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

10 Temmuz 2020

Murat Eren SARI

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ	x
KISALTMALAR.....	xi
ÖZET	xii
ABSTRACT	xiii
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	5
3. TEZ ÇALIŞMASINDA KULLANILAN ARTIRILMIŞ GERÇEKLIK (AG) VE COĞRAFİ BİLGİ SİSTEMLERİ (CBS) TEKNOLOJİLERİ	10
3.1. ARTIRILMIŞ GERÇEKLIK.....	10
3.1.1. Artırılmış Gerçeklik Teknolojisinin Donanımları	13
3.1.2. Kameralar.....	14
3.1.3. Takip ve Duyarlılık Sistemleri	14
3.1.4. İşlemciler	16
3.1.5. Göstericiler	17
3.1.5.1. Takılabilir Göstericiler.....	17
3.1.5.1.1. Optik Göstericiler.....	17
3.1.5.1.2. Retinal Göstericiler	19
3.1.5.1.3. Projektörler.....	20
3.1.5.2. Taşınabilir Göstericiler	21
3.1.5.3. Uzaysal Göstericiler.....	21
3.2. COĞRAFİ BİLGİ SİSTEMLERİ (CBS)	22
3.3. COĞRAFİ BİLGİ SİSTEMLERİNİN BİLEŞENLERİ	24
3.3.1. Donanım.....	24
3.3.2. Yazılım.....	24
3.3.3. Veri	24
3.3.4. Metot.....	25
3.3.5. İnsan.....	25
3.3.6. Coğrafi Bilgi Sistemlerinin Kullanım Alanları	25
4. GELİŞTİRİLEN MOBİL UYGULAMA.....	27
4.1. UYGULAMANIN GELİŞTİRİLMESİNDE KULLANILAN BİLEŞENLER	27
4.1.1. Android.....	27
4.1.2. Android Studio	27
4.1.3. Unity.....	27

4.1.4. Vuforia.....	28
4.1.5. 3DS Max	28
4.1.6. Google Maps	28
4.1.7. Google Maps API	29
4.2. GALATA KULESİ	29
4.3. RESTORASYON ÇALIŞMALARI	34
4.4. SANAL MODELLERİN OLUŞTURULMASI.....	37
4.5. TETİKLEYİCİLERİN SİSTEME ENTEGRASYONU.....	41
4.6. ARTIRILMIŞ GERÇEKLIK UYGULAMASININ GELİŞTİRİLMESİ	49
4.7. UYGULAMAYA YENİ MODEL EKLENMESİ.....	64
4.8. HARİTA UYGULAMASININ GELİŞTİRİLMESİ.....	65
4.9. HAZIRLANAN UYGULAMANIN ÇALIŞMA ŞEKLİ VE DENEYSEL ÇALIŞMALAR	76
5. SONUÇLAR VE ÖNERİLER.....	82
6. KAYNAKLAR	83
ÖZGEÇMİŞ.....	89

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. Örnek AG uygulaması [4].	1
Şekil 1.2. Anadolu’da kurulan medeniyetler (uygarlıklar) [6].	2
Şekil 1.3. 2014-2015-2016 yıllarında Restorasyon Daire Başkanlığı tarafından tamamlanan proje ve restorasyon çalışmaları [7].	3
Şekil 1.4. İsis Mozaği a) restorasyondan önce b) restorasyondan sonra [8].	3
Şekil 1.5. Günümüzde Galata Kulesi [9].	4
Şekil 3.1. AG teknolojisi ile yapılmış örnek bir uygulama.	11
Şekil 3.2. Gerçeklik-Sanallık Sürekliliği [37].	12
Şekil 3.3. Google Glass [40].	13
Şekil 3.4. AG donanımları ve işlevsel akış şeması [41].	14
Şekil 3.5. Resim tabanlı AG örneği [45].	15
Şekil 3.6. Konum tabanlı AG örneği [46].	15
Şekil 3.7. AG tür bakımından yıllara göre kullanımı.	16
Şekil 3.8. Artırılmış Gerçeklik gösterici çeşitleri.	17
Şekil 3.9. Video see through HMD gözlüğü [50].	18
Şekil 3.10. Video see through HMD çalışma prensibi [51].	18
Şekil 3.11. Optical see through HMD gözlüğü [52].	19
Şekil 3.12. Optical see through HMD çalışma prensibi [51].	19
Şekil 3.13. Retinal lens [53].	20
Şekil 3.14. SixthSense sistemi [54].	20
Şekil 3.15. Taşınabilir gösterici örneği [55].	21
Şekil 3.16. Uzaysal gösterici örneği [56].	22
Şekil 3.17. Uzaysal göstericinin çalışma mantığı.	22
Şekil 3.18. CBS’nin veri eşleştirme yapısı [58].	23
Şekil 3.19. CBS’nin katmanlı yapısı [59].	23
Şekil 3.20. CBS’nin ana bileşenleri.	24
Şekil 4.1. Matrakçı Nasuh’un Galata Kulesi ve Surlarının Minyatürü [72].	29
Şekil 4.2. Matrakçı Nasuh’un Galata Kulesi minyatürü [72].	30
Şekil 4.3. Antoine-Laurent Castellan’ın 19. yüzyılın başlarındaki Galata Kulesi’nin çizimi [73].	31
Şekil 4.4. 1850’li yıllarda Galata Kulesi [71].	32
Şekil 4.5. Sabiha Rüştü Bozcalı’nın 1831 yılı sonrası Galata Kulesi’nin çizimi [75].	32
Şekil 4.6. 1875 yılı sonrasındaki Galata Kulesi [76].	33
Şekil 4.7. Behcet Cantok’un 1875 yılı sonrasındaki Galata Kulesi’nin çizimi [74].	33
Şekil 4.8. Galata Kulesi’nin günümüzdeki hali [77].	34
Şekil 4.9. Kasnak panosundaki altıgenin ortasına sonradan eklenen yıldız motifleri [78].	35
Şekil 4.10. Tonoz tuğlalarının hatalı döşendiği a) 23 numaralı oda tavanı [78] b) 27 numaralı oda tavanı [78].	35
Şekil 4.11. Karakaya Hanının restorasyon sonrası dış cephe görünümü [79].	36
Şekil 4.12. Karakaya Hanının restorasyon sonrası iç görünümü [79].	36
Şekil 4.13. Gök Medrese’nin duvar örgüsü [78].	37

Şekil 4.14. 1794 yılındaki Galata Kulesi modelinin giydirilmemiş hali.	38
Şekil 4.15. 1794 yılındaki Galata Kulesi modelinin giydirilmiş hali.	38
Şekil 4.16. 1875 yılındaki Galata Kulesi modelinin giydirilmemiş hali.	39
Şekil 4.17. 1875 yılındaki Galata Kulesi modelinin giydirilmiş hali.	39
Şekil 4.18. 1967 yılındaki Galata Kulesi modelinin giydirilmemiş hali.	40
Şekil 4.19. 1967 yılındaki Galata Kulesi modelinin giydirilmiş hali.	40
Şekil 4.20. Bıblonun 7,5 derecelik açılarla çekilmiş görüntüleri.	41
Şekil 4.21. Veri tabanı oluşturma ekranı.	42
Şekil 4.22. Tetikleyici ayar ekranı.	42
Şekil 4.23. Veri tabanı inceleme ekranı.	43
Şekil 4.24. Veri tabanı indirme ekranı.	43
Şekil 4.25. Lisans anahtarı oluşturma ekranı.	44
Şekil 4.26. Lisans anahtarı inceleme ekranı.	44
Şekil 4.27. Görüntüdeki tanımlayıcılar [80].	45
Şekil 4.28. Özellik sayısına göre görüntü kalitesini [80].	45
Şekil 4.29. Histogramlara göre görüntü kalitesi [80].	46
Şekil 4.30. Vuforia mimarisi [81].	47
Şekil 4.31. Vuforia kütüphanesini indirme ekranı.	49
Şekil 4.32. Unity üzerinde Vuforia nesnelere.	50
Şekil 4.33. Artırılmış kamera ayarları.	50
Şekil 4.34. CylinderTarget nesnesinin ayarları.	51
Şekil 4.35. 1967 yılındaki Galata Kulesi'nin model dosyaları.	51
Şekil 4.36. Uygulamaya eklenen Galata Kulesi modeli.	52
Şekil 4.37. Butonlar eklenmiş uygulama ekranı.	52
Şekil 4.38. Focus scriptinin kodu.	53
Şekil 4.39. ArCamera'nın inspector ekranı.	54
Şekil 4.40. GalataSelector scriptinin kodu.	54
Şekil 4.41. GalataSelection'in inspector ekranı.	55
Şekil 4.42. InfoButton scriptinin kodu- 1.	55
Şekil 4.43. InfoButton scriptinin kodu- 2.	56
Şekil 4.44. InfoButton scriptinin kodu- 3.	57
Şekil 4.45. InfoManager'ın inspector ekranı.	58
Şekil 4.46. SoundEffect scriptinin kodu- 1.	58
Şekil 4.47. SoundEffect scriptinin kodu- 2.	59
Şekil 4.48. SoundManager'ın inspector ekranı.	60
Şekil 4.49. Galata1'in inspector ekranı.	61
Şekil 4.50. Uygulamanın son hali.	62
Şekil 4.51. Build Settings ayarları.	62
Şekil 4.52. Player Settings ayarları.	63
Şekil 4.53. AG tarafının akış diyagramı.	63
Şekil 4.54. Library Dependency ekranı.	65
Şekil 4.55. Task Loaded Callback kodu.	65
Şekil 4.56. Data Parser kodu- 1.	66
Şekil 4.57. Data Parser kodu- 2.	67
Şekil 4.58. Points Parser kodu.	68
Şekil 4.59. Fetch URL kodu.	69
Şekil 4.60. Maps Activity kodu- 1.	70
Şekil 4.61. Maps Activity kodu- 2.	71
Şekil 4.62. Maps Activity kodu- 3.	72
Şekil 4.63. Maps Activity kodu- 4.	73

Şekil 4.64. Maps Activity kodu- 5.	74
Şekil 4.65. Activity Maps kodu.	75
Şekil 4.66. Google Maps Api kodu.	75
Şekil 4.67. Harita tarafının akış diyagramı.	76
Şekil 4.68. Araç için güzergâh gösterimi.	77
Şekil 4.69. Yürüme için güzergâh gösterimi.	78
Şekil 4.70. a) Gerçek Galata Kulesi b) Galata Kulesi üzerinde 1967 yılındaki modelin gösterimi c) Galata Kulesi üzerinde 1875 yılındaki modelin gösterimi d) Galata Kulesi üzerinde 1794 yılındaki modelin gösterimi.	79
Şekil 4.71. 1967 yılındaki Galata Kulesi'nin uygulamada a) uzaktan gösterimi b) ekran görüntüsü c) modelin döndürülüp büyütülmüş hali.	79
Şekil 4.72. 1875 yılındaki Galata Kulesi'nin uygulamada a) uzaktan gösterimi b) ekran görüntüsü c) modelin döndürülüp büyütülmüş hali.	80
Şekil 4.73. 1794 yılındaki Galata Kulesi'nin uygulamada a) uzaktan gösterimi b) ekran görüntüsü c) modelin döndürülüp büyütülmüş hali.	80
Şeki 4.74. Uygulamada modeli a) 360 derece döndürme hareketi b) büyütme-küçültme hareketi.	81

ÇİZELGE LİSTESİ

Sayfa No

Çizelge 3.1. AG tür bakımından yıllara göre kullanımı16



KISALTMALAR

AG	Artırılmış Gerçeklik
API	Application Programming Interface
CBS	Coğrafi Bilgi Sistemleri
GPS	Global Positioning System



ÖZET

ARTIRILMIŞ GERÇEKLİK VE COĞRAFI BİLGİ SİSTEMLERİ YAKLAŞIMLARI KULLANILARAK TARİHİ ESER RESTORASYONU İÇİN MOBİL UYGULAMA GELİŞTİRME

Murat Eren SARI
Düzce Üniversitesi
Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi
Danışman: Dr. Öğr. Üyesi Metin TOZ
Temmuz 2020, 88 sayfa

Bu çalışmada artırılmış gerçeklik teknolojisi ile coğrafi bilgi sistemleri yaklaşımlarını bir arada kullanarak tarihi eserler üzerinde yapılan restorasyon çalışmalarında izlenecek yolu göstermek ve yapılabilecek geri dönüşü olmayan hataları önlemek için bir mobil uygulama geliştirilmiştir. Bu uygulama bir tarihi eseri ve onun üzerinde yapılan restorasyon çalışmalarını tarihsel süreç içerisinde kullanıcıya sunmayı ve dolayısıyla muhtemel bir sonraki restorasyon için yol göstermeyi hedeflemektedir. Çalışma kapsamında örneklendirme işlemleri İstanbul'un simgesi olan Galata Kulesi üzerinde yapılmıştır. Çalışmada Galata Kulesi üzerinde yapılan restorasyon çalışmaları yıllar temel alınarak gösterilmiştir. 2 Numaralı Kültür Varlıklarını Koruma Bölge Kurulu'ndan Galata Kulesi'nin modellenmesinde doğru sonuçlara ulaşmak için gerekli dokümanlar alınmıştır. Çalışma coğrafi bilgi sistemleri (CBS) ile desteklenerek verimlilik ve ulaşım noktalarında güçlendirilmiştir. Bu kapsamda birden fazla uzmanlık alanı kullanıldığı için daha etkin sonuçlar elde edileceği ön görülmektedir. Galata Kulesi'nin zamansal değişimleri net bir şekilde kullanıcıya aktarılabilirdiği için çalışma turistik anlamda da dikkat çekmektedir.

Anahtar sözcükler: Artırılmış Gerçeklik, Galata Kulesi, Restorasyon, 3b Modelleme.

ABSTRACT

DEVELOPMENT OF A MOBILE APPLICATION FOR A SAMPLE HISTORICAL ARTIFACT BY USING THE GEOGRAPHICAL INFORMATION SYSTEM AND THE AUGMENTED REALITY APPROACHES

Murat Eren SARI

Düzce University

Institute of Science, Department of Computer Engineering

Master's Thesis

Supervisor: Assist. Prof. Dr. Metin TOZ

July 2020, 88 pages

In this study, a mobile application has been developed by using both augmented reality technology and geographical information systems together in order to lead to the path that will be followed for restoration of historical artifacts and to prevent any irreversible mistakes that could be made. This application aims to provide users with information about a historical artifact and its restoration works in a historical context, and a guideline for the next possible restoration. Sampling has been done on Galata Tower which is one of Istanbul's symbols within this study. Restorations that have been done on Galata Tower have been shown based on years. Corresponding documents have been taken from No.2 Regional Board Directorate for Protection of Cultural Heritage in order to obtain precise outcomes to model Galata Tower. The study has been enhanced in terms of efficiency and transportation via geographical information systems. Because multiple expertise methods have been adopted, it is predicted that more effective consequences would be obtained. Since historical transformation of Galata Tower is reflected to users in a clear way, the study attracts attention in touristic manner as well.

Keywords: Augmented Reality, Galata Tower, Restoration, 3d Modeling.

1. GİRİŞ

Artırılmış gerçeklik (AG) teknolojisi son yıllarda birçok alanda aktif olarak kullanılmaktadır. AG ile ilgili yapılan araştırmalar bu teknolojinin kullanıcının duyularını güçlendirdiğini ve algısının arttırdığını göstermektedir [1]. Bu sayede AG teknolojisi, üzerinde çalışılan işin algılanmasına, detayları ile incelenmesine ve yapılacak işlemlerin doğuracağı sonuçlara geniş bir açıdan bakılmasına olanak sağladığı için hata oranını en aza indirme imkânı sunmaktadır. Lopez ve arkadaşları tarafından yapılan çalışmaya göre, AG teknolojisini kullanan öğrencilerin öğrenme hızlarının arttığı ve daha kalıcı bir öğrenme sağlandığı sonucuna ulaşılmıştır [2]. Standart yöntemlere göre AG teknolojisinin verimliliğinin daha fazla olduğunu gösteren başka bir çalışmada yer alan incelemeye göre, AG teknolojisini kullanan öğrencilerin eğitim sürecinde konu anlatımını 3 boyutlu olarak takip edebildikleri ve konuyla direkt etkileşim sağladıkları için algılarında pozitif bir artış olduğu gözlemlenmiştir [3]. Şekil 1.1’de örnek bir AG uygulaması gösterilmiştir.



Şekil 1.1. Örnek AG uygulaması [4].

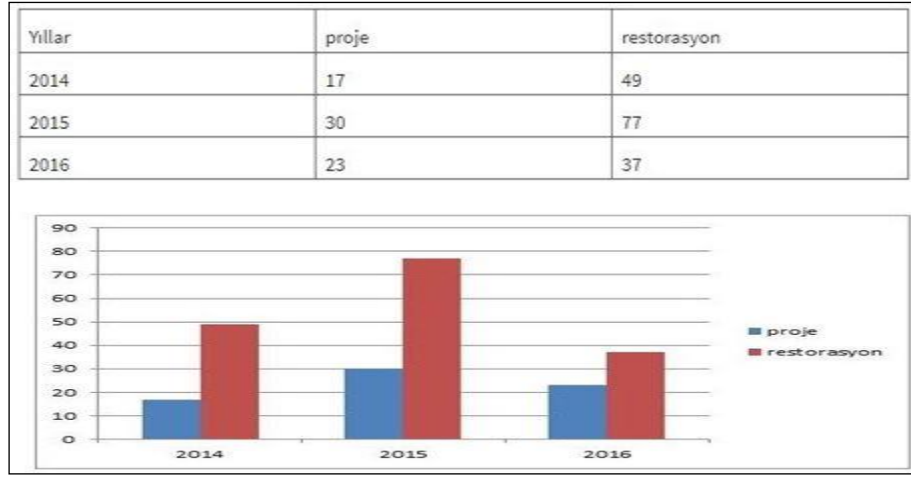
Türkiye coğrafi konumuna göre Avrupa ve Asya kıtalarının üzerinde yer almaktadır ve birçok medeniyetin beşiğidir. Ayrıca Türkiye'nin üç tarafı da denizlerle çevrilidir. Tarihi olarak incelendiğinde ise Türkiye topraklarına yerleşen birçok kavim, devlet ve topluluk olduğu görülmektedir. Yerleşen kavim ve kültürler örnek olarak Türkiye'nin

doğusunda kavimler olarak Hititlerden Urartulara, Urartulardan Selçuklulara kadar uzanırken kültür bakımından Hıristiyanlık, Müslümanlık, Musevilik dinlerinin etkileri görülmektedir. Türkiye'nin batısında ise kavimler olarak Romalılardan Yunanlılara kadar uzanmaktadır ve Türkiye'nin güney doğusunda ise kavimler olarak Avrupa kavimlerine kadar uzanırken kültür bakımından ise antik politeist Yunan-Roma kültürleri vb. kadar uzanan örnekler verilebilir [5]. Türkiye eski medeniyetlerin kesiştiği bir noktada olduğu için, Türkiye topraklarında çok sayıda önemli tarihi eserler ve yapılar bulunmaktadır. Anadolu'da kurulan bazı medeniyet ve uygarlıklar aşağıda harita üzerinde gösterilmiştir [6].



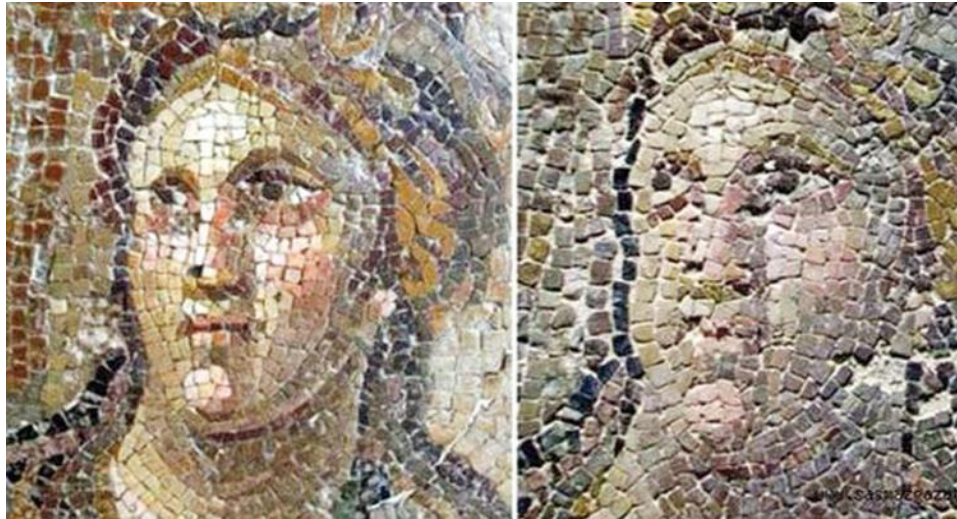
Şekil 1.2. Anadolu'da kurulan medeniyetler (uygarlıklar) [6].

Geçmişten günümüze gelen tarihi eserler incelendiğinde uzun süre çevresel etkenlere maruz kaldıkları için eserler üzerinde deformasyonlar meydana gelmektedir. Eserlerin orijinal hatlarını korumak ve çevresel etkenlerin hasarlarını engellemek adına eserlere restorasyon işlemleri gerçekleştirilir. Örnek olarak Türkiye bu doğrultuda 2014-2015-2016 yıllarında 70 proje, 163 restorasyon işlemi gerçekleştirmiştir [7]. Bu veriler aşağıda grafik halinde sunulmuştur.



Şekil 1.3. 2014-2015-2016 yıllarında Restorasyon Daire Başkanlığı tarafından tamamlanan proje ve restorasyon çalışmaları [7].

Restorasyon işlemleri sırasında geri dönüşü çok zor veya mümkün olmayan hatalar yapılabilmektedir. Yapılan hatalar incelendiği zaman, yapılması planlanan işlemlerin sonucunun net bir biçimde ortaya koyulamamasından kaynaklandığı görülmektedir. Bu doğrultuda milattan sonra 2. yüzyıldan kalmış olan İsis Mozaïği'ne yapılan ve aşağıda sunulan restorasyon çalışması incelendiğinde, restorasyon işlemleri sonrasında eserin özgünlüğünün kaybolduğunu ve tarihi yapısından uzaklaştığı görülmektedir [8].



a)

b)

Şekil 1.4. İsis Mozaïği a) restorasyondan önce b) restorasyondan sonra [8].

Eserin restorasyon işlemlerinin doğru belirlenememesi, izlenecek adımların sonucunun tam olarak görülememesi sebebi ile eserin restorasyon işlemi sonucunda Şekil 1.4.a'daki yapıdan Şekil 1.4.b'deki yapı ortaya çıkmıştır.

Bu tez çalışmasında AG teknolojisini CBS yaklaşımıyla destekleyerek restorasyon işlemleri sırasında meydana gelebilecek hataların önüne geçebilecek ve hata seviyesini en düşük seviyeye indirecek bir uygulama geliştirilmesi hedeflenmiştir. Bu çalışmanın sonucunda İsis Mozaïği'nde görüldüğü gibi muhtemel hataların önlenmesi ve yaşanacak maddi ve manevi zararın önüne geçilmesi hedeflenmiştir. Çalışma doğrultusunda hazırlanan uygulama için tarihi geçmiş ve uygulanan restorasyon işlemleri nedeniyle örnek model olarak İstanbul'un simgelerinden olan Galata Kulesi seçilmiştir (Şekil 1.5).



Şekil 1.5. Günümüzde Galata Kulesi [9].

Çalışmanın gerçeğe uygun olması adına Galata Kulesi'nin tarihi incelenmiş ve fiziki durumu üzerinde çeşitli araştırmalar yapılmıştır. Bu doğrultuda modelleme işleminin en doğru sonucu vermesi için 2 Numaralı Kültür Varlıklarını Koruma Bölge Kurulu'ndan gerekli dokümanlar alınmıştır.

CBS mekânsal ve mekânsal olmayan verileri toplayarak depolayan ve verileri analiz ederek kontrolünü sağlayıp kullanıcıya sunan sistemlerdir. Toplanan veriler noktalar, çizgiler, alanlar halinde tanımlanabilen yapılardadır [10]. Çalışmada harita üzerinde kişinin ve eserin konum verileri toplanmaktadır. Elde edilen veriler analiz ve kontrolden geçirilerek uygun şartları sağlaması durumunda AG kısmının aktifleştirilmesi gerçekleştirilir.

Tez çalışmasının ikinci bölümünde konu ile ilgili literatür çalışmalarına, üçüncü bölümünde tez kapsamında geliştirilen uygulama için kullanılan teknolojilere yer verilmiş, dördüncü bölümde geliştirilen uygulama tanıtılmış ve son olarak beşinci bölüm ile çalışma sonuçlandırılmıştır.

2. LİTERATÜR TARAMASI

Artırılmış gerçeklik ve coğrafi bilgi sistemleri ile ilgili literatürde yer alan çalışmalara bakıldığında bu teknolojilerin son yıllarda hayatın hemen her alanında kullanıldığı görülmektedir. Bu bölümde bu teknolojilerle ilgili literatür çalışmalarından örneklerle yer verilmiş ve tez çalışmasının bu açıdan literatüre katkısı değerlendirilmiştir.

Weiqin Chen [11], çalışmasında AG teknolojilerinden ve GPS verilerinden yararlanarak Oslo şehri için yapıların ilk halini gösteren bir çalışma geliştirmiştir. Geliştirmiş olduğu uygulama ile kullanıcıya şehirdeki yapıların eski hallerinin fotoğrafları üzerinden inceleyebilme imkânı sunulmuştur. Kullanıcı bu sayede geçmişten günümüze yapıların değişimlerini inceleyebilmektedir.

Irene Efstathiou ve ark. [12], çalışmalarında GPS verilerinden yararlanarak AG teknolojisinin tarihsel empatinin ve anlayışın gelişmesindeki etkisini öğrenciler üzerinde incelemişlerdir. Yapmış oldukları çalışmada öğrencilerin gezi sırasında AG teknolojisini kullanarak yerinde incelemeler ile klasik yöntemlere oranla daha hızlı öğrenme gösterdiklerini ortaya koymuşlardır.

Chao Hung Wang ve ark. [13], çalışmalarında telefon tamiri için klasik yönlendirme yöntemi ile AG teknolojisini kullanarak yönlendirme yöntemi arasındaki başarı farklarını ortaya koymuşlardır. Yapmış oldukları çalışmada kullanıcılar Google Glass aracılığıyla kullandıkları AG teknolojisi ile klasik yöntemlere göre hata oranlarının daha düşük olduğunu ve daha kaliteli iş çıkardıklarını ortaya koymuşlardır.

Tomasz Oleksy ve ark. [14], çalışmalarında harita sistemlerini ve AG teknolojisini kullanarak hazırladıkları çalışma ile Polonya'da bulunan Muranow bölgesinin geçmiş tarihini ortaya koymuşlardır. Çalışma ile kullanıcıya bölgeye ait yerlerin eski fotoğrafları ile bölgeyi inceleme imkânı sunulmuştur.

Yu Lien Chang ve ark. [15], çalışmalarında AG teknolojisini ve klasik yöntemleri kullanarak tarihi yerler konusunda üniversite öğrencileriyle yapılan incelemeler ile kullanılan yöntemler arasındaki başarı farklarını ortaya koymuşlardır. AG teknoloji ile öğrenciler klasik yöntemlere göre daha fazla detay öğrenirken kişilerin öğrenme motivasyonlarının da arttığı gözlemlenmiştir.

Florin Girbacia ve ark. [16], çalışmalarında AG teknolojisini kullanarak Romanya’da bulunan Kara Kilise’nin dışındaki heykeller üzerine bir yapı kurmuşlardır. Kurmuş oldukları yapı ile tahrip olmuş heykellerin ilk halini gösteren bir çalışma ortaya koymuşlardır. Kullanıcıya bu sayede tahrip olmuş heykeli günümüzde kilise üzerinde inceleyebilme imkânı sunulmuştur.

Levent Sabah ve Mehmet Şimşek [17], çalışmalarında harita sistemlerinden ve AG teknolojisinden yararlanarak mobil bir kampüs bilgilendirme sistemi yapmışlardır. Hazırladıkları çalışmada kişilerin konum bilgilerini kullanarak kampüs içerisindeki gezintilerinde yardımcı bir yapı oluşturmuşlardır. Çalışma ile kullanıcı kampüs içerisinde ihtiyaç duyabileceği bilgilere ulaşımı sağlanmaktadır.

Çağdaş Erbaş ve Veysel Demirer [18], çalışmalarında eğitimde AG teknolojisinin durumunu incelemişlerdir. Yaptıkları çalışmada AG teknolojisi ile eğitim verilirken kişiye sağlayabileceği avantajlar üzerine araştırmalar yapmışlardır. AG teknolojisinin sağladığı avantajları göz önüne alarak eğitim alanında kullanılması için çeşitli önerilerde bulunmuşlardır.

Serkan Kıvrak ve Gökhan Arslan [19], çalışmalarında inşaat sektörünü temel alarak AG teknolojisi ile inşaat sırasında uygulanan adımlar için bir sistem geliştirmişlerdir. Yaptıkları çalışma inşaat sahasında çalışan birçok meslek grubunun kullanabileceği yapıdadır. Geliştirilen sistem ile kişiye inşaat alanında uygulanacak adımları aşamalarla anlatarak bir yol haritası sunmaktadır.

Merve Polat [20], çalışmasında konumsal verileri ve AG teknolojisini kullanarak turistik amaçlı mobil bir uygulama geliştirmiştir. Kullanıcı çalışmada kullanılan AG teknolojisi ile ilgili konumdaki tarihi eser hakkında detaylı bilgiye erişebilmektedir.

Mehmet Albayrak ve Volkan Altıntaş [21], çalışmalarında AG teknolojisini kullanarak öğrencilere video ve metin destekli ders notlarından hazırladıkları bir yapı oluşturmuşlardır. Öğrencileri 2 gruba ayırarak bir grup klasik yöntemlerle hazırlanmış ders notlarını kullanırken diğer grup ise AG teknolojisi ile hazırlanmış ders notlarını kullanmışlardır. Yaptıkları çalışma ile iki grup arasındaki başarı farkını ortaya koymuşlardır.

Vassilios Vlahakis ve ark. [22], çalışmalarında ArcheoGuide adını verdikleri projede GPS verilerinden ve AG teknolojisinden yararlanarak bir yapı geliştirmişlerdir. Proje ile kullanıcı arkeolojik alanlarda tarihi eserlerin kalıntıları üzerinden ilk halini

deneyimlemeyebilmektir. Bu sayede kişi günümüzde yapının ilk halini turistik açıdan inceleyebilme imkânına sahip olabilmektedir.

Anne-Cecilie Haugstvedt ve ark. [23], çalışmalarında harita sistemlerinden ve AG teknolojisinden yararlanarak tarihi bir sokağın geçmişini kullanıcıya sunan bir uygulama geliştirmişlerdir. Yapmış oldukları uygulama sayesinde günümüzde uygulamada belirtilen yıllardaki sokağın görünümünü fotoğraflar aracılığı ile kullanıcıya inceleme imkânı sunulmaktadır.

Gun A. Lee ve ark. [24], çalışmalarında harita sistemlerinden ve AG teknolojisinden yararlanarak depremle büyük zarar görmüş Yeni Zelanda da bulunan Christchurch şehrinin depremden öncesi için bir uygulama geliştirmişlerdir. Hazırlanan uygulama sayesinde depremin vermiş olduğu zarardan öncesini kişiye inceleme imkânı sunulmuştur. Kullanıcı bu sayede yıkılan yapıların yıkılmadan önceki halini görüntüleyebilmektedir.

George Papagiannakis ve ark. [25], çalışmalarında sanal gerçeklik teknolojisi ile antik Pompeii kentinin yok olmadan önceki halini ortaya koymuşlardır. Yaptıkları çalışma ile kullanıcı kullandığı gözlük sayesinde döneme ait olan freks ve boyamalar gibi eserleri ve döneme ait kıyafetler giymiş çalışanları inceleme imkânı sunulmaktadır.

Andrea F. Abate ve ark. [26], çalışmalarında AG teknolojinden yararlanarak müzelerde sergilenen tarihi eserlerin ilk hallerini gösteren bir yapı ortaya koymuşlardır. Geliştirdikleri yapı sayesinde çeşitli sebeplerden dolayı hasar görmüş eserlerin orijinal durumlarını kullanıcıya deneyimleyebilme imkânı sunulmaktadır.

Vedat Can Ak [27], çalışmasında AG teknolojisini kullanarak eğitim alanında kullanılmak üzere mobil bir uygulama geliştirmiştir. Çalışmasında AG teknolojisinin algıda pozitif artış sağlaması özelliğini üzerinden prototip bir uygulama ortaya koymuştur. Uygulama ile kullanıcıya mobil cihazlar üzerinden insan vücudunun bazı organlarını 3 boyutlu inceleme imkânı sunulmuştur.

Gizem Sucaklı [28], çalışmasında AG teknolojisinin müzecilik alanında kullanımını araştırmıştır. Çalışmasında AG teknolojisi ile modern müzecilik anlayışını yakalayan yurt içi ve yurtdışı müzeleri incelemiştir. İncelemeler sonucunda AG teknolojisinin müzecilik alanında gelişimini sağlanabilmesi adına önemini ortaya koymuştur.

Oğuz Balas [29], çalışmasında AG teknolojisini kullanarak sergileme alanında “Kutsal Evlilik” adını verdiği mobil bir uygulama geliştirmiştir. Çalışmasında AG

teknolojisinden yararlanmak için ayna ışın birleştiricisi tasarımı yapmıştır. Yapmış olduğu çalışma ile kullanıcılara sergi esnasında AG ile görsel duyuları artırarak farklı bir açıdan eserleri inceleme imkânı sunmuştur.

İmran Kazan [30], çalışmasında AG teknolojisinin turizm sektöründe kullanımının sonuçlarını araştırmıştır. Yaptığı araştırmalar ile AG teknolojisinin turizm sektörüne sağlayacağı avantajları incelemiştir. AG teknolojisinin kişi üzerinde algı artışı sağladığı için turizm alanında kullanılmasıyla sektörün verimli, etkili ve daha faydalı hale getirilebileceğini ortaya koymuştur.

Emmanuel Durand ve ark. [31], AG teknolojisinden yararlanarak tarihi eserlerin gösterimi üzerine çalışma ortaya koymuşlardır. Çalışmalarında AG teknolojisinden yararlanarak Fransa'da bulunan tarihi Cluny Kilisesi'nin orijinal halinin gösterimini gerçekleştirmişlerdir. Bu sayede çalışma ile yapının orijinal halini kullanıcıya turistik açıdan inceleme imkânı sunulmuştur.

Faruk Can Ünal [32], çalışmasında harita sistemlerini ve AG teknolojisini mimarlık alanında kullanımını incelemiştir. Çalışmasında elde ettiği yerel verileri işleyerek AG teknoloji aracılığı ile konum tabanlı olacak şekilde kullanıcıya sunmuştur. Çalışması konum ile direkt etkileşim halinde olduğu için kullanıcının yerel verileri direkt yerinde incelemesine imkân verilmektedir.

Gürcan Serbest [33], çalışmasında AG teknolojisinden yararlanarak kültürel miraslar üzerine bir çalışma yapmıştır. Çalışma kültürel mirasın kişilere aktarımını kolaylaştırıp, istenilen detaylara ulaşılmasını sağlamaktadır. Çalışma ile sanal ve AG teknolojilerinin kültürel açıdan öğrenme ve eğitim noktalarındaki önemini ortaya koymaya çalışılmıştır.

Fulya Demircioğlu [34], çalışmasında GPS verilerini ve AG teknolojisini kullanarak altyapı alanında bir çalışma yapmıştır. Çalışmasında doğalgaz alt yapısı verilerini kullanarak yapılmış tesisatın gerçek konumlarını AG ve CBS teknolojileri ile kullanıcıya sunmaktadır. Çalışma ile kullanıcının daha verimli sonuçlar almasına olanak sağlayan bir platform oluşturmak hedeflenmiştir. Kurulan yapı ile saha da yapılacak operasyonlara karar verme ve rehber olması adına bir sistem oluşturulmuştur.

Çalışmalar incelendiğinde AG teknolojisi ile kullanılan 3B sanal nesnelerin oranı 2B sanal nesnelerin oranından fazla olduğu gözlemlenmiştir. Bu tez çalışmasında da AG teknolojisini daha gelişmiş bir yapı olan 3B nesnelerle etkileşimli bir şekilde kurulmuştur. İncelenen çalışmaların genelinde CBS sisteminden yararlanıldığını fakat

rota oluřturma iřlemine daha az yer verildiđi gzlemlenmiřtir. Bu tez alıřmasında CBS sisteminden yararlanılmıř ve kullanıcıya alternatif yntemlerle kullanabileceđi rota oluřturma zelliđi sunulmuřtur. İncelenen alıřmaların genelinde bir tetikleyici bir sanal nesne ile etkileřimli olacak řekilde yapı kurulduđu ve birden fazla sanal nesneli yapıların az sayıda olduđu gzlemlenmiřtir. Hazırlanan alıřmada bir tetikleyici birden fazla sanal nesne ile etkileřimli olacak řekilde bir yapı kurulmuřtur. İncelenen alıřmaların genelinde AG teknolojisini grsel algıyı artırma ynnde kullanıldıđı gzlemlenmiřtir. Hazırlanan alıřmada ise hem grsel algı hem de iřitsel algının artırılması zerine bir yapı kurulmuřtur.

Yukarıdaki literatr taramasına gre sunulan tez alıřmasına bakıldıđında bu alıřmanın literatre katkıları řu řekilde zetlenebilir.

- Hazırlanmıř olan uygulama ile birlikte AG ve CBS yaklařımları bir arada kullanılmıřtır. Kullanıcıya CBS yapısı ile farklı yntemlerle Galata Kulesi'ne ulařım imknı sunulmuřtur. Kullanıcı harita zerinden ara ve yrme yntemleri ile en kısa gzergha eriřilebilmektedir.
- AG teknolojisi ile ise kullanıcı Galata Kulesi'nin farklı zamanlarda geirmıř olduđu birbirinden farklı 3 boyutlu restorasyon modellerini inceleyebilmektedir. Hazırlanan uygulama ile oluřturulan Galata Kulesi modellerini dndrme, bytme-kltme hareketleri ile kullanıcıya modelleri kolaylıkla incelemesi iin seenekler sunulmuřtur. Uygulama ierisinde her model iin oluřturulan bilgilendirme metni ile kullanıcı bilgilendirmesi yapılmaktadır.
- Her model iin oluřturulan ses kayıtları ile kullanıcının iřitsel algısında artıř sađlanmıřtır.

Sonuç olarak bu tez kapsamında geliřtirilen uygulama ile CBS yaklařımıyla desteklenen AG teknolojisi ile kullanıcıda hem grsel hem de iřitsel algısında artıř sađlayan bir yapı oluřturulmuřtur.

3. TEZ ÇALIŞMASINDA KULLANILAN ARTIRILMIŞ GERÇEKLİK (AG) VE COĞRAFI BİLGİ SİSTEMLERİ (CBS) TEKNOLOJİLERİ

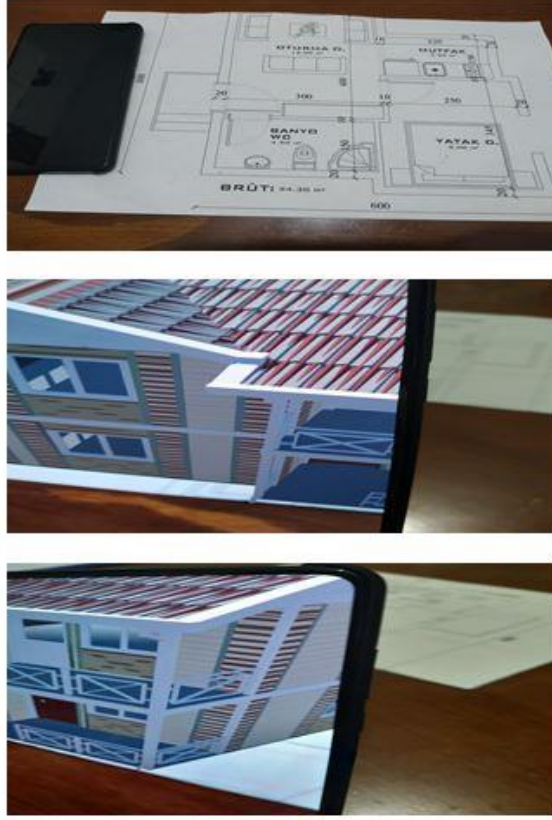
Bu bölümde tez çalışmasında geliştirilen uygulama için kullanılan artırılmış gerçeklik ve coğrafi bilgi sistemleri teknolojileri tanıtılmıştır.

3.1. ARTIRILMIŞ GERÇEKLİK

Artırılmış gerçeklik teknolojisinin amacı gerçek dünya üzerinde sanal nesnelere oluşturarak, kullanıcıya sanal nesnelere gerçek dünya üzerinde deneyimletmektir. Kullanıcı bu deneyimi yaşayabilmesi için, yazılımla hazırlanmış olan sanal nesnelere gerçek dünya üzerinde eş zamanlı olarak görüntü üzerinde birleştirilerek kullanıcıya gösterilmektedir [35].

AG teknolojisinin ortaya çıkışı incelendiği zaman, gerçeklik sanallık düzlemi ile 1960 yılında Ivan Sutherland ile birlikte Utah Üniversitesi'ndeki öğrencisi tarafından gerçekleştirilen gösterimle ortaya çıkmıştır [36].

AG teknolojisi sayesinde kullanıcıya sunulan sanal nesnelere ile gerçek dünyanın birleştiği ortam kullanıcının algısında artış sağlamaktadır. Kullanıcıda gerçekleşen algıda artış sonucunda kişinin olayları algılama sürecinin kısılması ve bilgi görsellerle detaylandırıldığı için bilginin kalıcılığının artmasına imkân sağlanmaktadır. Örnek bir uygulama aşağıdaki şekilde sunulmuştur.



Şekil 3.1. AG teknolojisi ile yapılmış örnek bir uygulama.

Artırılmış gerçeğin yapısı incelenirken sanal gerçeklik kavramı ile karıştırılabilmektedir. Sanal gerçeğin yapısı incelendiği zaman çalışma mantığı AG gibi görünmesine rağmen içlerinde keskin ayrımlar bulunmaktadır. AG kişiye gerçek dünyada sanal nesnelere etkileşim imkânı sunmaktayken sanal gerçeklik ise kişiye dünyadan bağımsız, sanal nesnelere kendi yapısını kurduğu bir dünyada deneyim imkânı sunmaktadır. Burada kişinin yaşadığı tüm deneyim yazılımla geliştirilen dünyadaki nesnelere gerçekleştirilmektedir. Bu noktada AG'deki gibi gerçek zamanlı bir etkileşim söz konusu değildir.

Sanal ve gerçek ortamın anlaşılabilirliği için 1994 yılında Milgram ve Kishino "Gerçeklik Sanallık Sürekliliği" adında bir şekil oluşturmuşlardır [37]. Oluşturdukları şekil incelendiğinde, şeklin bir tarafına gerçek ortam ve diğer tarafına ise sanal ortam yerleştirilmiştir. Bilgisayar ortamında hazırlanan nesnelere yoğunluğuna göre kullanıcıya sunulan dünya bu yapı üzerine yerleştirilebilir. Kullanıcıya sunulan dünyadaki nesnelere sayısı ve gerçek dünya ile olan etkileşim göz önüne alınarak yapı incelendiğinde soldan sağa doğru gerçek dünya ile olan bağın zayıfladığı görülmektedir (Şekil 3.2).



Şekil 3.2. Gerçeklik-Sanallık Sürekliliği [37].

Literatürde, artırılmış gerçeklik sahnelerinin üç ortak özelliği şu şekilde listelenmektedir [38]:

- Sanal ve gerçek nesnelere gerçek dünyada bütünleştirilir,
- Bu sahneler gerçek zamanlı etkileşim içerir,
- İlişkili gerçek ve sanal nesnelere birbirlerine uygun şekilde harmanlanır [38].

Azuma'nın çalışmaları incelendiği zaman AG teknolojisinin havacılık, üretim - tamir, kişisel gelişim, sağlık, yol planlama alanları üzerindeki kullanımına dikkat çekmiştir [39]. AG teknolojisinin büyük gelişmelerini bu alanlarda gerçekleştirilmesi ile birlikte eğlence sektörlerinde ve reklamcılık alanlarında da oldukça büyük gelişmeler göstermiştir.

AG teknolojisi, 1999 yılında piyasa çıkan AR-ToolKit yazılımının açık kaynak kodlu olması nedeniyle bilimsel ve ticari alanlardaki gelişiminin hızlanmasına neden olmuştur. Geliştirilen AR-ToolKit ile bilgisayar ortamında AG teknolojinin kullanımına olanak sağlanmıştır. 1999 yılından itibaren gelişimine devam eden AR-ToolKit günümüzde kullanımı devam etmektedir.

Teknolojinin hızlı gelişimi ile birlikte 2000'li yıllarda ortaya çıkan akıllı telefonlar AG teknolojinin kullanım alanını artırarak, gelişiminin hızlanmasına neden olmuştur. Telefonların sahip olduğu kamera sistemi ile AG teknolojisinin sınırları oldukça genişlemiştir. Bu sebeple günlük hayatın birçok alanında AG teknolojisinin kullanımına zemin hazırlamıştır.

Bilgisayar ve akıllı telefonlarda kullanımına devam edilen AG teknolojisinin kullanım alanı 2013 yılında Google'ın çıkardığı “Google Glass” (Şekil 3.3) ile daha da genişlemiştir. Bu tarz bir gelişim ile AG teknolojisi bir adım daha ileriye giderek, günlük hayatın kullanılmayan birçok alanında kullanımına olanak sağlamıştır.



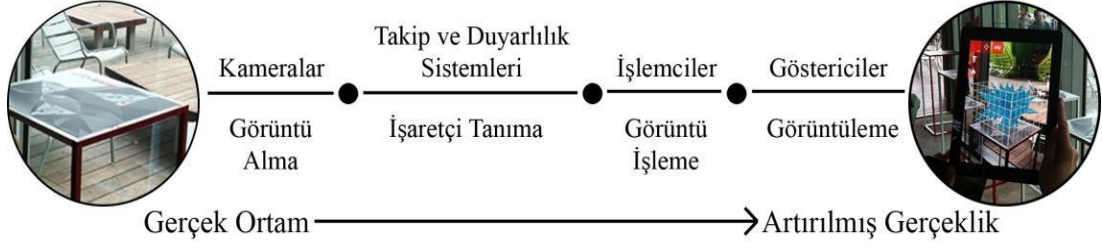
Şekil 3.3. Google Glass [40].

AG teknolojisinin gelişimi sadece görsel algıyı artırmak üzerine kurulu değildir. Geri kalan duyularımız üzerinde de algıda artış sağlayabilmektedir. Araç park etme sırasında herhangi bir nesneye yaklaşmasıyla ses ile uyarı sistemi devreye girmesi ses üzerine yapılmış bir AG uygulamasıdır. Kullanıcıya daha fazla veri sunmasından dolayı yapılan çalışmaların büyük kısmı görsel ortam yaratma üzerinedir [41].

AG teknolojisini yapısal olarak incelendiği zaman bilgisayar ortamında hazırlanmış olan 3B yapılar, videolar, sesler, konum bilgileri gibi verilerin gerçek dünya ile etkileşime sokularak kullanıcı deneyimine sunulmasıdır. Kullanıcının deneyimine sunulabilmesi için çeşitli donanım ve yazılımlara ihtiyaç duyulmaktadır.

3.1.1. Artırılmış Gerçeklik Teknolojisinin Donanımları

AG teknolojisinin çalışma mantığı incelendiğinde işlem sırası olarak görüntünün yakalanması, görüntüde bulunan yazılımla tanımlanmış tetikleyicinin algılanması, gerçek dünya üzerinde gösterilecek sanal nesnelerin oluşturulması ve son görüntünün gösterimini gerçekleştiren donanımlar olarak sıralayabiliriz. Sistem Şekil 3.4'te gösterildiği gibi kameralar, takip ve duyarlılık sistemleri, işlemciler ve göstericiler olarak incelenmiştir [41].



Şekil 3.4. AG donanımları ve işlevsel akış şeması [41].

3.1.2. Kameralar

AG teknolojisinin başlangıcı kameralardır. Kamera, hazırlanmış yazılımla birlikte çalışarak kullanıcının AG teknolojisini deneyimleyebilme imkânı verir. Gerçek dünyadaki görüntüyü alarak yazılımla tanımlanmış olan tetikleyicileri işlenmesine olanak sağlar. Tetikleyiciler, gerçek dünya üzerinde belirlenmiş noktalara yerleştirilen işaretçiler veya kameranın algılayabileceği desenlerden oluşmaktadır [42]. Bu sayede bir sonraki adım olan takip ve duyarlılık sisteminin aktifleşmesine imkân verir.

3.1.3. Takip ve Duyarlılık Sistemleri

AG teknolojisinin kameralardan sonraki adımı takip ve duyarlılık sistemleridir. Bu noktada yapılan yazılımla birlikte sistemin gerçek dünya üzerinde sistematik bir şekilde çalışması hedeflenir. Bilgisayar ortamında oluşturulan nesnelere gerçek dünya üzerinde yerleştirilmeleri ve uygun şekilde görünmeleri bu noktada incelenir. Uygulamanın verimli çalışabilmesi için sistemin oluşturacağı veriler oldukça önemlidir.

Takip sistemleri incelendiğinde resim ve konum tabanlı olarak iki ana başlığa ayrılmaktadırlar [43]. Sistemin alt başlıklarına inildiğinde ise resim tabanlı sistem, işaretçi tabanlı ve işaretçi tabanlı olmayan sistemler olmak üzere iki alt başlığa ayrılmaktadır.

Resim tabanlı sistem incelendiğinde AG teknolojisi ortamdaki nesnelere ilişkilendirilmiştir. Resim tabanlı AG gerçek dünyada bulunan nesnelere referans almaktadır. Bu nesne ortamdaki resim (işaretçi tabanlı sistemler) veya ses/el hareketi (işaretçi tabanlı olmayan sistemler) şeklinde olabilir [44] (Şekil 3.5).



Şekil 3.5. Resim tabanlı AG örneği [45].

Konum tabanlı sistem ise GPS koordinatlarından sağladığı verilerle AG teknolojisinde tetikleme yaparak kullanıcıya veriyi sunar (Şekil 3.6).

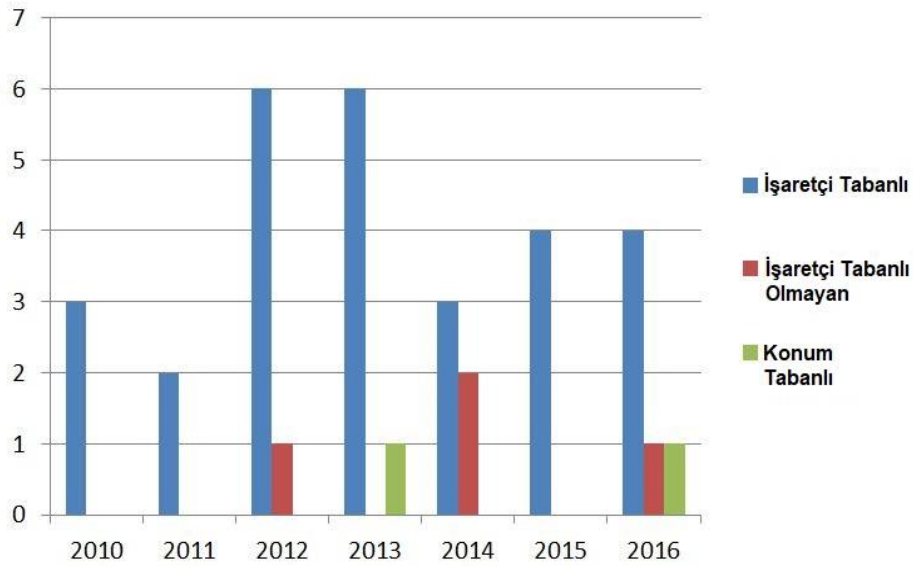


Şekil 3.6. Konum tabanlı AG örneği [46].

AG alanında yapılan çalışmalarda kullanılan yöntemlerin yıllara göre dağılımı ise Çizelge 3.1 ve Şekil 3.7’de gösterilmektedir.

Çizelge 3.1. AG tür bakımından yıllara göre kullanımı.

Yıl	Resim Tabanlı AG		Konum Tabanlı
	İşaretçi Tabanlı	İşaretçi Tabanlı Olmayan	
2010	3	0	0
2011	2	0	0
2012	6	1	0
2013	6	0	1
2014	3	2	0
2015	4	0	0
2016	4	1	1
Toplam	28	4	2



Şekil 3.7. AG tür bakımından yıllara göre kullanımı.

3.1.4. İşlemciler

İşlemciler AG sisteminin “beyin” görevini üstlenen temel bileşenlerindedir [47]. Bilgisayar ortamında hazırlanmış olan sanal nesnelerin gerçek dünya üzerinde doğru noktalarda oluşmasını sağlayan donanımdır. Yapılan yazılımın kapsamı ile doğru orantılı olarak ihtiyaç duyulan işlemcilerin özelliklerinde değişiklikler görülebilmektedir.

3.1.5. Göstericiler

AG teknolojisinin işlevsel akışta son elemanı olan göstericiler bilgisayar ortamında hazırlanmış olan sanal nesnelere kameranın sunduğu gerçek dünya ile etkileşimini kullanıcının deneyimlemesine imkân veren bileşendir. Hazırlanan takip ve duyarlılık sistemi işlemcinin performansına bağlı olarak AG deneyiminin kalitesini göstericiler aracılığı ile son kullanıcıya sunar.

Gösterici tipi, AG teknolojiyle üretilecek olan verinin tasarım kararına ve kullanım işlevine bağlı olarak değişiklik gösterebilir. Göstericiler, kullanıcının bakış açısına ve gerçek obje pozisyonuna göre takılabilir göstericiler, taşınabilir göstericiler ve uzaysal göstericiler olarak 3'e ayrılır [48].



Şekil 3.8. Artırılmış Gerçeklik gösterici çeşitleri.

3.1.5.1. Takılabilir Göstericiler

Takılabilir göstericiler kullanıcının görüş alanı üzerinde bulunarak AG görüntüsünü kullanıcıya sunar. Bu kapsamda takılabilir göstericileri optik göstericiler, retinal göstericiler ve projektörler olarak gruplamak mümkündür [48].

3.1.5.1.1. Optik Göstericiler

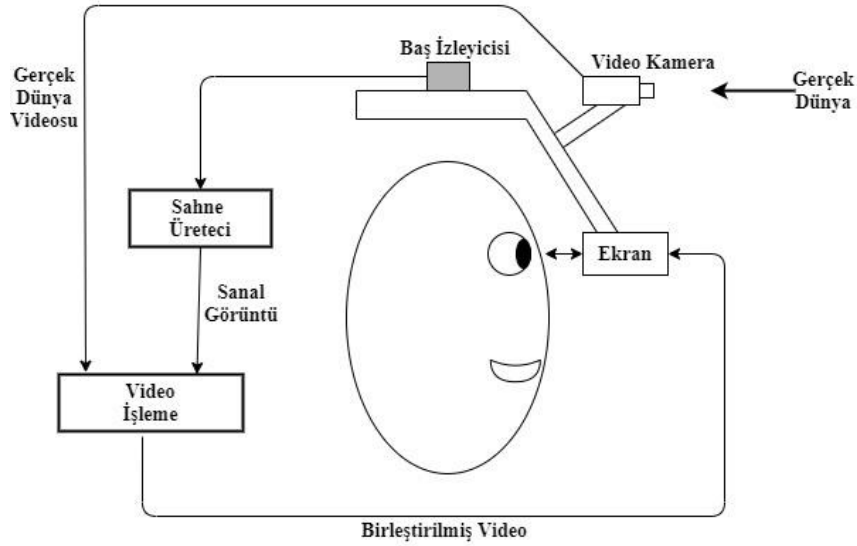
Optik göstericiler, gerçek dünyayı video olarak işleyerek sanal nesnelere ile iki görüntüyü birleştiren göstericiler (video see through HMD) ve oluşturulmuş sanal nesnelere gerçek dünya üzerine yerleştiren göstericiler (optical see through HMD) olarak ikiye ayrılır [49].

Video see through HMD yapısı incelendiğinde kullanıcı AG teknolojisini deneyimlemek için kullandığı gözlükteki kameralar ile gerçek dünyanın görüntüsünü kaydeder. Kullanıcı gözlüğü taktığında görüş açısı tamamen kapanmaktadır [49].



Şekil 3.9. Video see through HMD gözlüğü [50].

Yapının çalışma prensibi (Şekil 3.10) incelendiğinde kullanıcı hazırlanmış olan yazılımla gözlük içinde var olan ekran aracılığı ile gerçek ve sanal yapının birleşimini deneyimler.



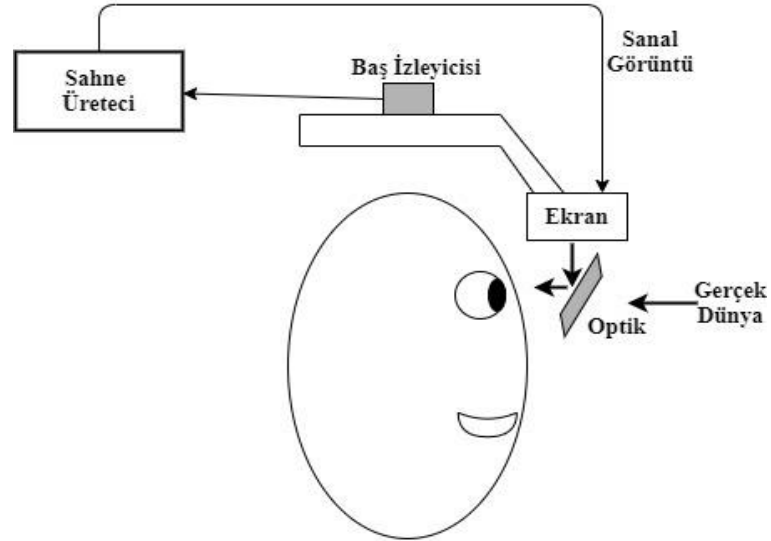
Şekil 3.10. Video see through HMD çalışma prensibi [51].

Optical see through HMD yapısı incelendiğinde ise video see through HMD gibi görünse de keskin ayrım noktaları vardır. Video see through HMD de kullanıcının görüşü tamamen kapalıyken, optical see through HMD de kullanıcının taktığı gözlük camı saydam şekildedir. Bu sayede kullanıcı gerçek dünyayı anlık olarak görebilmektedir. Gözlükte bulunan optik camlar aracılığı ile oluşturulmuş olan sanal nesnelere gerçek dünyadaki tetikleyiciler ile etkileşime geçtiklerinde optik üzerinden kullanıcıya sunulur [51].



Şekil 3.11. Optical see through HMD gözlüğü [52].

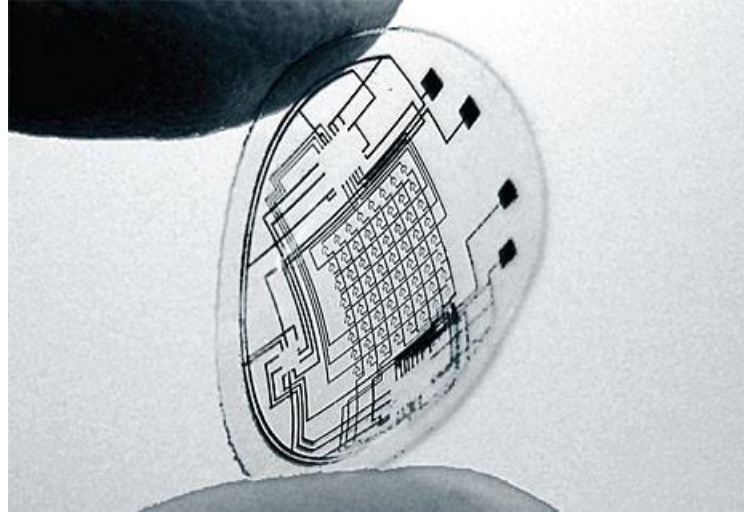
Yapının çalışma prensibi (Şekil 3.12) incelendiğinde, kullanıcı hazırlanmış olan yazılım sayesinde gözlük camındaki optik ekran aracılığı ile gerçek ve sanal yapının birleşimini deneyimler [51].



Şekil 3.12. Optical see through HMD çalışma prensibi [51].

3.1.5.1.2. Retinal Gösterciler

Aktif olarak üzerine çalışmalar yapılan kontakt lenslerin gelişimi devam etmektedir. Prof. Babak A. Parviz önderliğinde AG lens prototipleri hazırlanmıştır [53]. Lenslerin çalışma mantığı incelendiğinde optik gözlükler gibi kullanıcı sanal nesnelere lens aracılığı ile gerçek dünya üzerinde deneyimleyebilmektedir. Lensler direkt göz ile temas içinde çalıştığı için insan sağlığına etkileri üzerine incelemeler yapılmaktadır.



Şekil 3.13. Retinal lens [53].

3.1.5.1.3. Projektörler

Pranav Mistry'nin hazırlamış olduğu SixthSense, projektör kullanarak kullanıcıya AG deneyimi sunmaktadır [54]. Bu yapıda kişi üzerine yerleştirilmiş olan kamera gerçek dünyayı analiz etmektedir. Hazırlanan yazılımla sanal nesnelere projektör aracılığı ile gerçek dünyaya yansıtılır. Yazılımla tanımlanmış olan parmaklara takılan renkli işaretçiler aracılığı ile kullanıcı sistemi aktif olarak kullanımını gerçekleştirir. Üzerinde aktif olarak geliştirmelerin devam ettiği bu yapının yakın zamanda hayatımızın birçok alanında kullanılacağı ön görülmektedir [54].

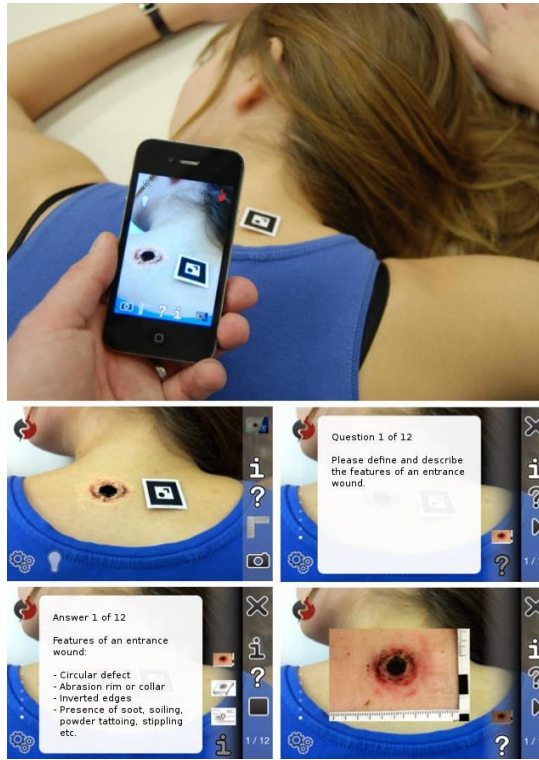


Şekil 3.14. SixthSense sistemi [54].

3.1.5.2. Taşınabilir Göstericiler

Taşınabilir göstericiler AG teknolojinin çalışabilmesi için her donanımın içinde bulunduğu sistemlerdir. Taşınabilir göstericiler grubundaki akıllı telefonlar ve tabletler en yaygın kullanılan cihazlardır. Bu cihazların taşınabilir olması, günlük hayattaki ihtiyaçların karşılanmasına uygun olması gibi sebeplerden dolayı kullanıcıların yaygın olarak tercih ettiği sistemlerdir [41].

Taşınabilir göstericiler video görüntülü sistemlerin çalışma mantığını kullanmaktadır. Telefonun kamerası gerçek dünyanın görüntüsünü hazırlanan yazılımla işleyerek telefonun ekranında tanımlanmış alanlara sanal nesnelerin yerleştirilmesiyle kullanıcıya AG teknolojisini deneyimleme fırsatı sunar. Örnek bir uygulama Şekil 3.15'te sunulmuştur.



Şekil 3.15. Taşınabilir gösterici örneği [55].

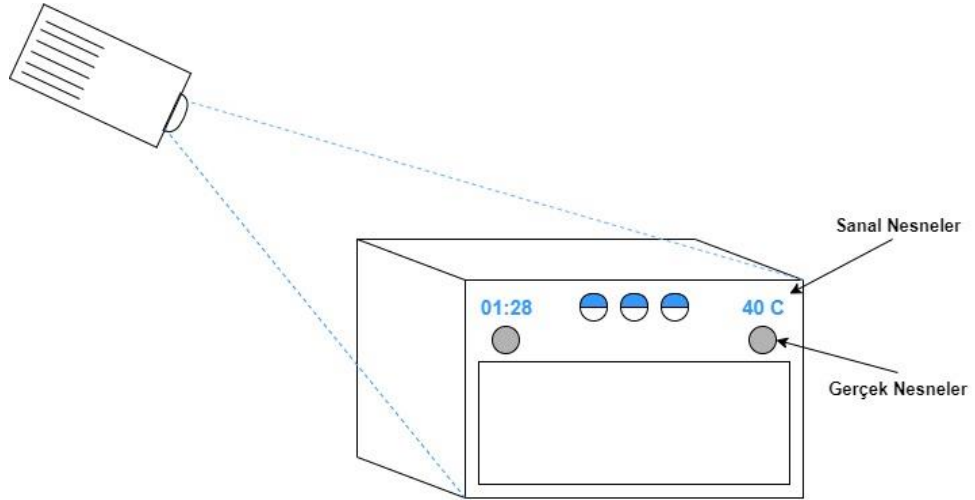
3.1.5.3. Uzaysal Göstericiler

Uzaysal göstericiler, projeksiyon aracılığı ile bilgisayar ortamında hazırlanmış olan görüntünün gerçek objeler üzerine yansıtılması ile AG deneyimi sunan sistemlerdir. Uzaysal göstericiler diğer sistemlere göre daha pahalı bir donanım sistemi kullanmaktadır. Günümüzde uzaysal göstericiler havada yapılan yansıtılmalarda aktif olarak kullanılmaktadır.



Şekil 3.16. Uzaysal gösterici örneği [56].

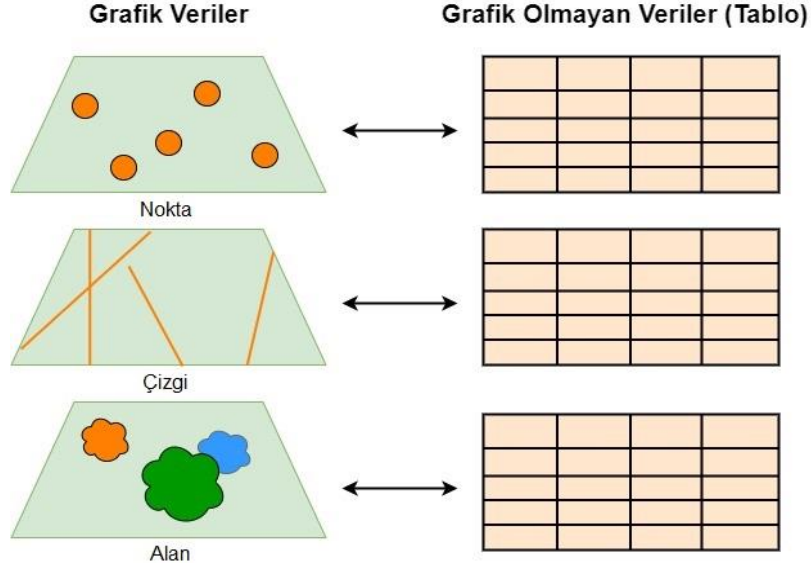
Sistemin çalışma mantığı Şekil 3.17’de örnek bir senaryo üzerinde gösterilmiştir.



Şekil 3.17. Uzaysal göstericinin çalışma mantığı.

3.2. COĞRAFI BİLGİ SİSTEMLERİ (CBS)

Coğrafi bilgi sistemleri ortam verilerinin toplandığı, kullanıcı isteği durumunda ilgili veriye erişebildiği bir yapı sunmaktadır. Verileri birbirleriyle ilişkilendirebilen, verilerin doğruluğunu sorgulayabilen, veriyi analiz ederek farklı kaynaklardan alınan verileri kaydedebilen bir yapıdadır. Verileri temel olarak yönetme, planlama ve analiz işlemlerinde destek sağlar. Tüm bu yapıyı standartlaştıran ve haritalar ile tabloları Şekil 3.18’deki gibi eşleştiren sistemlerdir [57].



Şekil 3.18. CBS'nin veri eşleştirme yapısı [58].

CBS'nin kullanım alanları incelendiği zaman mühendislik alanlarından finans sektörüne, temel bilimlerden beşeri bilimlere kadar günlük hayatta birçok alanda aktif olarak kullanılan sistemlerdir. Günlük olarak sıkça kullandığımız harita sistemlerinin temellerini de Şekil 3.19'da gösterilen CBS'nin katmanlı yapısı sağlamaktadır.



Şekil 3.19. CBS'nin katmanlı yapısı [59].

CBS'nin tarihsel gelişimi incelendiğinde harita yapımı ve kullanımları M.Ö. 4000 yıllarına kadar uzanmaktadır. Kavramsal olarak ele alındığında ise 1960'lı yıllarda Kanada'nın arazilerinde yapılan özelliklerine göre sınıflandırma işlemleri sırasında geliştirilen Kanada CBS ile ortaya çıkmıştır. İlk ortaya çıkışında kısıtlı kullanımı olan CBS, teknolojinin gelişimi ile birlikte sistemini güncellemiş ve içeriğini artırmıştır. Günümüzde birçok sektörde oldukça gelişmiş bir altyapı ile kullanılmaktadır.

3.3. COĞRAFİ BİLGİ SİSTEMLERİNİN BİLEŞENLERİ

CBS'nin bileşenleri (Şekil 3.20) incelendiğinde donanım, yazılım, veriler, metotlar ve insan olarak beş bileşen üzerinden incelenebilmektedir [60].



Şekil 3.20. CBS'nin ana bileşenleri.

3.3.1. Donanım

Günümüzde bakıldığında CBS'nin donanım olarak kullandığı yapılar günlük kullanılan bilgisayarlardan iş istasyonlarına kadar uzanmaktadır. Kurulan yapının içerisinde CBS'ye ait veriler depolanıp analiz edilerek kullanıcıya sunulur. Hazırlanmış olan sistemin kapsamına bağlı olarak kullanılan yapılarda farklılıklar görülebilmektedir [61].

3.3.2. Yazılım

CBS verilerinin analiz edileceği kısım yazılımdır. Yazılımın detaylı olması çıkacak sonucun kalitesini etkilemektedir. CBS'nin kapsamının ve işlemlerinin büyüklüğü göz önüne alındığında yapılan bir yazılım tüm kullanıcıların isteklerini karşılayamayacaktır. İstenilen işe özgü bir yazılım hazırlanarak daha spesifik sonuçlar elde etmek verimliliği artıracaktır [61].

3.3.3. Veri

Donanım ve yazılımın kalitesi kadar verinin de kalitesi oldukça önemlidir. CBS ne kadar donanım ve yazılım anlamında verimli olsa da işlenen verideki hatalar tüm sistemin verimini düşürecektir. Bu sebeple doğru veri toplama için çeşitli yöntemler bulunmaktadır. Toplanacak verilerin türüne bağlı olarak en uygun yöntem seçilerek

verilerin sisteme girdisi yapılmalıdır. Toplanan verilerin, yazılıma uygun olarak düzenlenmesinden verilerin saklanmasına kadar olan tüm yapının programlama tekniklerine uygun olarak yapılmasına "veri yönetimi" denilmektedir [62].

3.3.4. Metot

Metotlar CBS'nin işleyişinin belirlendiği kısımdır. CBS'nin verimli olması için birçok bileşeni bir arada çalışabilmesi ve yönetilmesi gerekmektedir. Eldeki kaynakların en iyi şekilde yönetilmesi için proje öncesi, proje süreci ve proje sonrası izlenecek yöntemler belirlenmelidir. Bu kapsamda sistemin sürdürülebilir ve yenilebilir olması gerekmektedir. İzlenecek yöntemler doğrultusunda elde edilecek verimlilik ortaya çıkacaktır.

3.3.5. İnsan

İnsan CBS'nin en kilit bileşenidir. İnsan kurulacak olan sistemi yöneten, geliştiren ve devamlılığını sağlayan faktördür. Sistemin katmanlarında görev alan kişilerin uzmanlık seviyeleri doğrultusunda başarılı bir sistem ortaya koyulmaktadır. Yapılan süreçlerde ortaya çıkacak hatalar veya eksiklikler yapıya oldukça büyük zararlar verebileceği için insan faktörünün çok iyi belirlenmesi gerekmektedir.

3.3.6. Coğrafi Bilgi Sistemlerinin Kullanım Alanları

CBS birçok alanda aktif olarak kullanılan bir yapıdır. Kullanıldığı alana bağlı olarak farklı çözümler oluşturmada yardımcı olur. Coğrafi verilerin işlenmesi söz konusu olduğu her noktada CBS kullanılması mümkündür. Bu kapsamda incelendiğinde en yaygın kullanım alanları şu şekildedir [63]:

- Alt yapılar
- Arama kurtarma faaliyetleri
- Arkeoloji
- Askeri uygulamalar
- Bilimsel araştırmalar
- Coğrafi tarih
- Doğal afet acil müdahalelerde
- Haritacılık
- Hava, deniz ve kara trafiği izleme araç takip sistemleri
- Kartografya
- Maden haritalama

- Meteoroloji
- Pazarlama
- Peyzaj mimarlığı
- Tarım ekili tarım alanlarının tespiti ve toplam mahsulün hesaplanması
- Varlık yönetimi
- Çevresel etki değerlendirmesi [63].

4. GELİŞTİRİLEN MOBİL UYGULAMA

Bu bölümde tez çalışması kapsamında geliştirilen mobil uygulama tanıtılmıştır. Bunun için ilk olarak uygulamanın geliştirilmesinde kullanılan bileşenler tanıtılmış ardından uygulama ve kullanılış detayları ile sunulmuştur.

4.1. UYGULAMANIN GELİŞTİRİLMESİNDE KULLANILAN BİLEŞENLER

4.1.1. Android

Android, Google firması tarafından geliştirilmiş ve gelişimi devam eden mobil bir işletim sistemidir. Geliştirilen bu sistem, aktif olarak milyonlarca kişiye hizmet vermektedir. Android açık kaynaklı bir sistemdir. Açık kaynaklı bir sisteme sahip olmasından dolayı geliştiriciler ve kullanıcılar tarafından tercih edilen bir yapıya sahiptir. Geliştirici ve kullanıcı sayısının fazla olmasından dolayı Android teknolojisi oldukça hızlı bir şekilde gelişimine devam etmektedir. Sistemin Linux üzerine kurulmuş olması da yapıyı daha esnek hale getirmektedir [64][65].

Geniş bir yapıya sahip olan Android içerisinde birçok kütüphane barındırmaktadır. Sağladığı Java desteği ile oldukça dinamik bir yapıya da sahiptir.

4.1.2. Android Studio

Android Studio, Google'ın Android geliştiricileri için hazırladığı geliştirme ortamıdır. İçerisinde barındırdığı esnek proje yapıları, tüm sürümlere ait hızlı emülatörler ve uygulama geliştirme imkânı, yardımcı araçlar ve kütüphaneler [66] gibi özellikleri ile Android'in gelişim sürecini hızlandırmıştır. Android Studio aktif olarak kullanılan ve gelişimi devam eden bir ortamdır.

4.1.3. Unity

2005 yılında piyasa çıkan Unity'nin yapısı basitçe incelendiğinde birden çok platforma (bilgisayar, mobil cihazlar, konsollar) uygulama geliştirmeye imkân veren çapraz platformlu bir oyun motorudur [67]. Unity çapraz platformlu olması nedeniyle bir işletim sistemine geliştirilen uygulama hızlı bir şekilde diğer işletim sistemlerine

entegreli hale getirilmesine imkân sunmaktadır. İçerisinde bulundurduğu ücretsiz kütüphaneler sayesinde zengin bir alt yapıya sahiptir. Unity 2B ve 3B objelerle yapı geliştirilmesini sağlayan kütüphanelerle geliştiriciye destek sağlamaktadır. Diğer birçok geliştirme ortamı ile eş zamanlı çalışması ise sağladığı büyük avantajlardandır. Program içerisinde Java, C# gibi dillere destek verdiği için geliştiriciye esnek bir yapı sunmaktadır. İçerisinde bulunan hazır birçok yapı ile birlikte sunulan imkânlar nedeniyle Unity’i kullanan geliştirici sayısı oldukça fazladır. Popüler bir uygulama olan Unity gelişimine devam etmektedir.

4.1.4. Vuforia

Vuforia, artırılmış gerçeklik uygulaması geliştirenler için hazırlanmış bir yazılım geliştirme kitidir. Kendi içerisinde kullandığı yapının kararlılığı nedeniyle geliştiriciler için avantaj sağlamaktadır. Vuforia sanal nesnelerin gösteriminde kullanılan tetikleyicilerin 2B veya birçok formatta 3B model kullanımına imkân vermektedir. Birçok yazılım diline destek sağladığı için oldukça esnek bir yapısı bulunmaktadır. Farklı işletim sistemlerine uyum sağlaması ve geliştiricilere ücretsiz olarak uygulama geliştirme imkânı veren Vuforia’yı kullanan geliştirici sayısı oldukça fazladır. Popüler bir yazılım geliştirme kiti olarak kullanılan Vuforia aktif olarak gelişimine de devam etmektedir.

4.1.5. 3DS Max

3DS Max Autodesk firmasının geliştirdiği bir programdır. İçerisinde barındığı geniş ve kapsamlı modelleme araçları nedeniyle karakter veya yapı modellemede kullanıcıların dikkatini çekmektedir. İçinde barındırdığı geniş özelliklere sahip render, hareket yakalama, karakter ve animasyon araçları, modelleme araçları gibi birçok özelliklerle geniş bir kullanıcı kitlesine hitap etmektedir [67]. Program mimarlık alanlarında, endüstriyel tasarımlarda, televizyon reklamlarında, bilimsel çalışmalarda, bilgisayar oyunlarında, özel efektlerin hazırlanmasında gibi birçok alanda aktif olarak kullanılmaktadır. Günümüzde en çok tercih edilen modelleme programlarından biri olan 3DS Max gelişimine devam etmektedir.

4.1.6. Google Maps

Google Maps, Google’ın desteklediği bir harita uygulamasıdır. Uygulamanın esnek yapısı sayesinde farklı işletim sistemleri üzerinden uygulamaya erişmek mümkündür.

Sunulan yapının ücretsiz ve birden fazla platformda sorunsuz çalışması nedeniyle günümüzde aktif olarak milyonlarca kullanıcısı bulunmaktadır. Google Maps'in oldukça gelişmiş bir alt yapısı bulunmaktadır ve kullanılan yapının gelişimi devam etmektedir.

4.1.7. Google Maps API

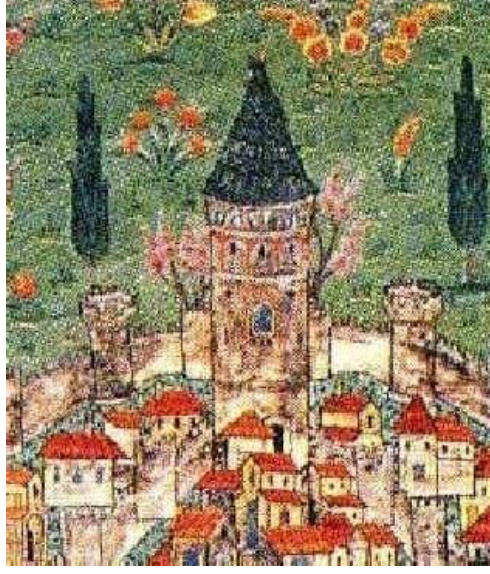
Google Maps API hizmeti, Google tarafından yazılım geliştiricilerin yaptıkları uygulamalara Google Maps'i rahatlıkla entegre edebilmeleri için geliştirdikleri bir hizmettir. Uygulama geliştiricilerine sunulan geliştirme ara yüzüne ticari yapılar hariç ücretsiz erişim sağlanabilmektedir [68]. İçerisinde barındırdığı işaret ekleyerek dinamik harita oluşturma sistemiyle uygulama geliştiricilerine esnek yapı kurmalarına imkân vermektedir [69]. Sunulan hizmetlerin içeriğinin kapsamlı ve verilerin doğruluğunun yüksek olması nedeniyle hizmeti kullanan uygulama geliştirici sayısı oldukça fazladır.

4.2. GALATA KULESİ

İstanbul'un simgelerinden biri olan Galata Kulesi çeşitli kaynaklarda 528 yılında Bizans İmparatoru Anastasius tarafından fener kulesi olarak kullanılmak üzere yaptırıldığı yazmaktadır [70-71]. Tarihsel akışta incelendiğinde yaşanan savaşlar ve İstanbul'da ortaya çıkan büyük yangınlar nedeniyle büyük oranda zarar gören Galata Kulesi 1348 yılında Cenevizliler tarafından İsa Kulesi adıyla neredeyse tekrar inşa edilmiştir. Kule 1445-1446 yıllarında ise tekrar tadilata girerek yükseltme işlemi yapılmıştır. Aşağıda Galata Kulesi'nin Matrakçı Nasuh tarafından çizilen minyatürleri sunulmuştur.



Şekil 4.1. Matrakçı Nasuh'un Galata Kulesi ve Surlarının Minyatürü [72].



Şekil 4.2. Matrakçı Nasuh'un Galata Kulesi minyatürü [72].

İstanbul'un fethi ile Galata Kulesi Türklerin hâkimiyetine geçmiştir. II. Bayezid zamanında yaşanan depremde Galata Kulesi ve surları hasar görmüştür. Kule Osmanlı elindeyken hemen her yüzyıl yenilenmiş ve tamir edilmiştir. Savunma amacıyla kullanılan Galata Kulesi, Osmanlı himayesi altındayken farklı amaçlar doğrultusunda da kullanılmıştır. 16. yüzyılda tersanede çalışan savaş esirlerinin kalacağı yer olarak kullanılan Galata Kulesi 18. yüzyılda ise mehteran ocağı tarafından kullanılmıştır [73]. 1874 yılına gelindiğinde ise konumu ve yüksekliği nedeniyle şehre hâkim olmasından dolayı yangın gözetleme amacıyla kullanılmıştır.

Yapısal değişiklikleri incelendiğinde ise III. Selim zamanında çıkan yangın nedeniyle kulenin büyük bölümü zarar görmüştür. Bu yangında kulenin tepesindeki külâhı tamamıyla yanmıştır. Bu nedenle Galata Kulesi'ne 1794 yılında restorasyon işlemi yapılmıştır. İşlemler sonrasında kulenin tepesine tekrardan bir külâh konulmuştur ve en üst kısmına dört tarafa çıkıntı şeklinde alanlar eklenmiştir. Şekil 4.3'te yenilenmiş hali gösterilmektedir.



Şekil 4.3. Antoine-Laurent Castellan'ın 19. yüzyılın başlarındaki Galata Kulesi'nin çizimi [73].

1831 yılına gelindiğinde II. Mahmud zamanında yaşanan depremler ve yangın nedeniyle zarar gören Galata Kulesi, tekrardan kullanılabilir hale getirilebilmesi için bugünkü görüntüsünü almıştır. Onarımla en üst kat değiştirilip kemerli ve büyük pencereler kullanılarak yeni halini almıştır. Kuleyi kurşun ile kaplanıp yerleştirilmiş ve pencere çevresi parmaklıklarla kapatılarak 360 derecelik bir görüş açısı sağlanmıştır [74]. Şekil 4.4 ve Şekil 4.5'te II. Mahmud zamanında yenilenen Galata Kulesi'nin yeni hali gösterilmektedir.



Şekil 4.4. 1850’li yıllarda Galata Kulesi [71].



Şekil 4.5. Sabiha Rüştü Bozcalı’nın 1831 yılı sonrası Galata Kulesi’nin çizimi [75].

1875 yılında ise çıkan fırtına nedeniyle kulenin külahı uçmuştur. Yine aynı yılda yapılan tamir ile kulenin görüntüsü değiştirilmiştir. Külahın bulunduğu noktaya 2 ufak oda yapılarak tepelerine bayrak direği dikilmiştir. Şekil 4.6 ve Şekil 4.7’de 1875 yılında yenilenen Galata Kulesi’nin yeni hali gösterilmektedir.



Şekil 4.6. 1875 yılı sonrasındaki Galata Kulesi [76].



Şekil 4.7. Behcet Cantok'un 1875 yılı sonrasındaki Galata Kulesi'nin çizimi [74].

1960lı yıllarda en üste bulunan odalardaki kirişlerin çürüyerek çökmesiyle tekrardan kulenin tamiri yapılmıştır. Alınan kararlar doğrultusunda Galata Kulesi II. Mahmud döneminde yapılan şekline dönüştürülmesi kararlaştırılmıştır. İstanbul Belediyesi'nin 1964 yılında başlattığı çalışmalar 1967 yılına kadar sürmüştür. 1967 yılında ise Galata Kulesi tekrardan kullanıma açılmıştır. Şekil 4.8'de 1967 yılında yenilenen Galata Kulesi'nin yeni hali gösterilmektedir.



Şekil 4.8. Galata Kulesi'nin günümüzdeki hali [77].

4.3. RESTORASYON ÇALIŞMALARI

Restorasyon yapılan eser, ilgili milletin hem tarihini hem de kültürünü yansıttığı için yapılan işlemlerin hata payı en aza indirgenmelidir. Olası bir hatanın geri dönüşü olmayan sonuçları ortaya çıkabilir. Restorasyon çalışmaları sırasında kullanılan klasik yöntemlerde insani duyu faktörleri de dâhil olmaktadır. İşlemler sırasında insani duyu faktörlerinin yetersiz olmasından dolayı çeşitli hatalar oluşabilmektedir. Yapılan çeşitli çalışmalarda AG teknolojisi insani duyu faktörlerini artırıp desteklediği ortaya konulmuştur. Bu kapsamda restorasyon çalışmalarında izlenecek adımlarda AG teknolojisi kullanılması ile hataların azalması ön görülmektedir. İncelenen restorasyon çalışmaları, klasik yöntemlerden farklı olarak daha doğru sonuçlara ulaşmak için yenilikçi bir yöntemin ihtiyacını göstermektedir.

İzzeddin Keykavus Darüşşifası'na yapılan çalışmalar incelendiğinde 13. yüzyıldan kalma olan yapıda gerçekleştirilen hatalı işlemler sonucunda yapının tarihi dokusu hasar görmüştür. Türbe kasmağındaki panolara yapılacak restorasyon projesinde altıgenin ortasında yıldız motifi yokken çalışma esnasında yıldız motifi konulmuştur [78].



Şekil 4.9. Kasnak panosundaki altıgenin ortasına sonradan eklenen yıldız motifleri [78].

Yapının tonoz tuğlaları ise düzgün sıra ile yerleştirilmediği, derz aralıklarının fazla bırakıldığı ve harçla doldurulduğu, düşey derzlerin her iki tonozda da mevcut sırayı takip etmediği Şekil 4.10'da gösterilmektedir [78].



a)

b)

Şekil 4.10. Tonoz tuğlalarının hatalı döşendiği a) 23 numaralı oda tavanı [78] b) 27 numaralı oda tavanı [78].

14. yüzyıldan günümüze kadar varlığını koruyan Karakaya Hanı Selçuklular döneminde inşa edilmiştir. Yapı günümüzde varlığını sürdürebilmesi için yenileme ihtiyacı ortaya çıkmıştır. Bu doğrultuda yapılan restorasyon çalışmaları incelendiğinde derzlere ve sıvalara yapılan işlemler tekniklere uygun gerçekleştirilmediği için kalitesiz bir işçilik ortaya çıkmıştır. Yapılan döşemelerin seviyeleri incelendiğinde orijinal halinden farklı olduğu kemerlerin yüksekliğinden fark edilmektedir [79].



Şekil 4.11. Karakaya Hanının restorasyon sonrası dış cephe görünümü [79].



Şekil 4.12. Karakaya Hanının restorasyon sonrası iç görünümü [79].

Gök Medrese incelendiğinde 13. yüzyılın sonlarına doğru III. Gıyaseddin Keyhüsrev zamanında Sahip Ata Fahreddin Ali tarafından yaptırılmıştır. Yapıya uygulanan restorasyon işlemleri sonucunda, yapının karakterini bozan durumlar ortaya çıkmıştır. Medresenin duvarlarına yapılan işlemler incelendiğinde medresenin tarihi dokusuna uyumsuz duvar örgüsü yapılmıştır [78].



Şekil 4.13. Gök Medrese'nin duvar örgüsü [78].

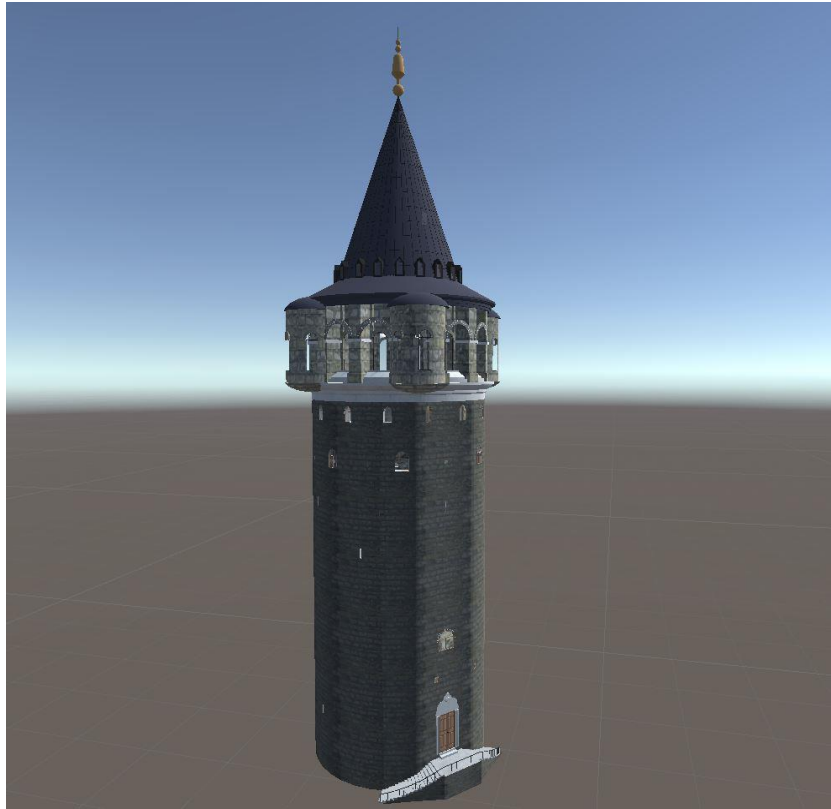
Bu tez çalışmasında geliştirilen uygulama Galata Kulesi örneği üzerinden aşağıdaki gibi sunulabilir.

4.4. SANAL MODELLERİN OLUŞTURULMASI

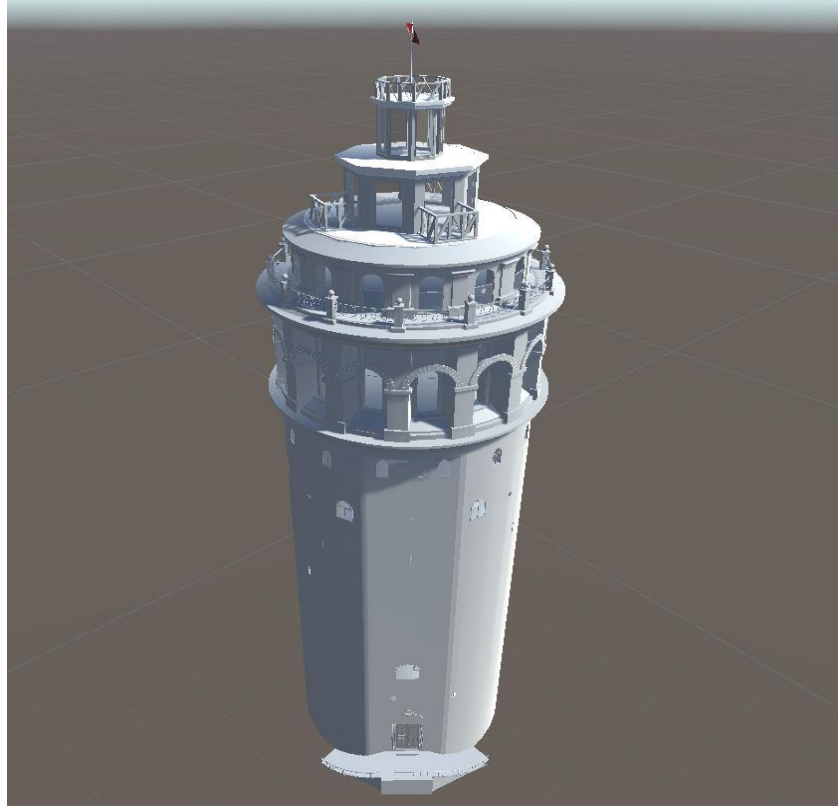
Uygulamada kullanılacak modeller için 3DS Max programı kullanılarak Galata Kulesi'nin belirlenmiş yıllara göre modelleme işlemleri aşağıdaki şekillerde gösterilmiştir.



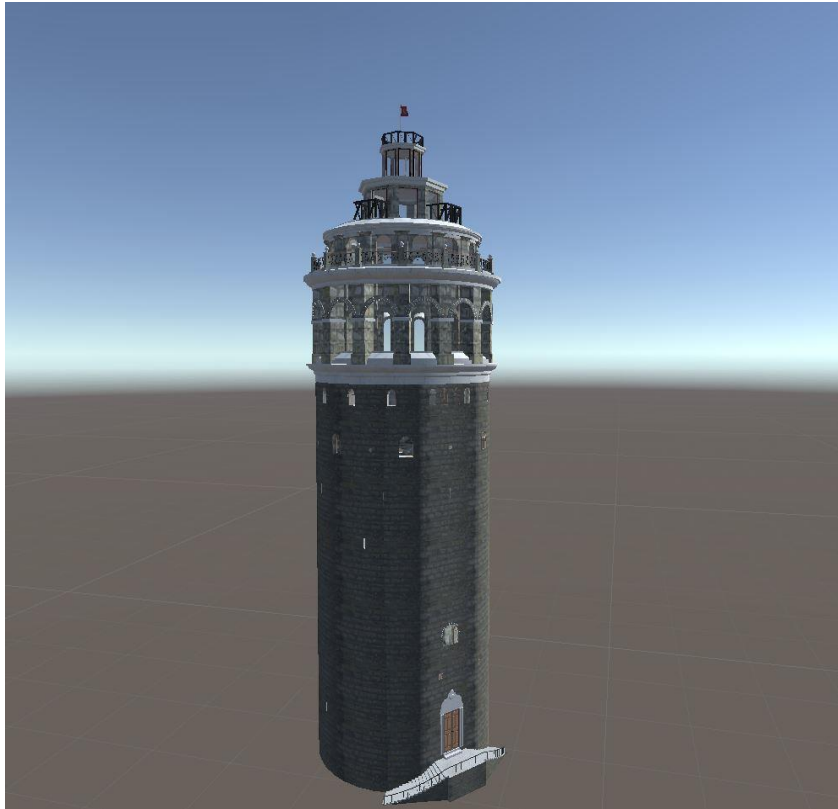
Şekil 4.14. 1794 yılındaki Galata Kulesi modelinin giydirilmemiş hali.



Şekil 4.15. 1794 yılındaki Galata Kulesi modelinin giydirilmiş hali.



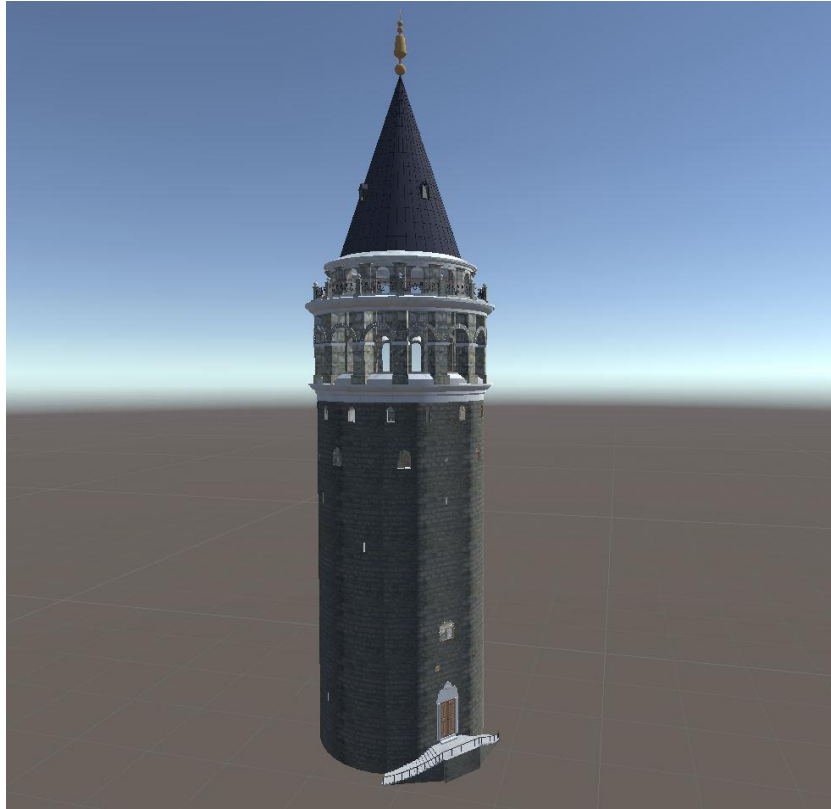
Şekil 4.16. 1875 yılındaki Galata Kulesi modelinin giydirilmemiş hali.



Şekil 4.17. 1875 yılındaki Galata Kulesi modelinin giydirilmiş hali.



Şekil 4.18. 1967 yılındaki Galata Kulesi modelinin giydirilmemiş hali.



Şekil 4.19. 1967 yılındaki Galata Kulesi modelinin giydirilmiş hali.

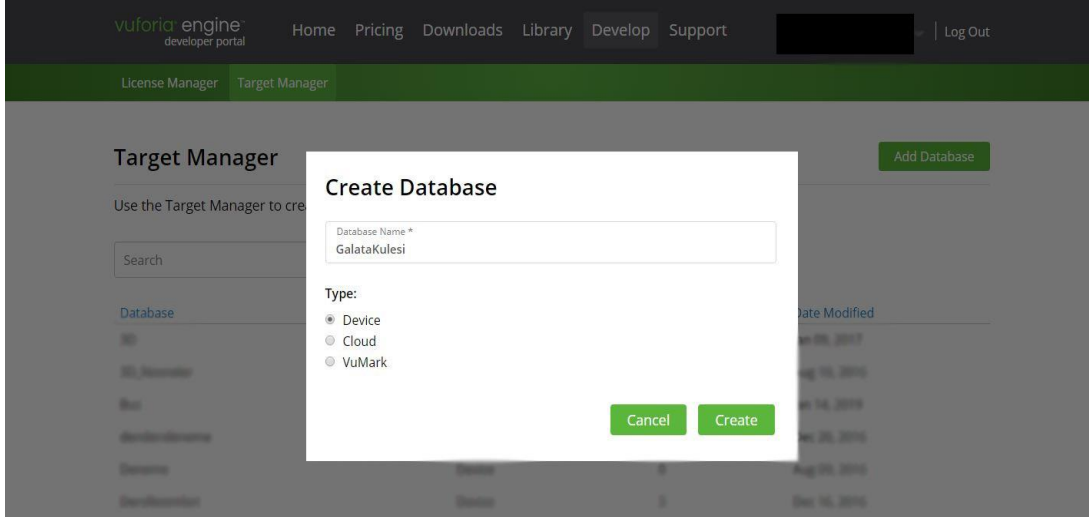
4.5. TETİKLEYİCİLERİN SİSTEME ENTEGRASYONU

Hazırlanan sanal modellerin gerçek dünya üzerinde kullanılması için tetiklenme işlemini gerçekleştirecek Galata Kulesi biblosunun 360 derecelik çekimi gerçekleştirilmiştir. Biblio, 7,5 derecelik dar açı ile döndürülerek gerçekleştirilen çekim işlemlerinin sonucunda Şekil 4.20'deki gibi görseller elde edilmiştir.



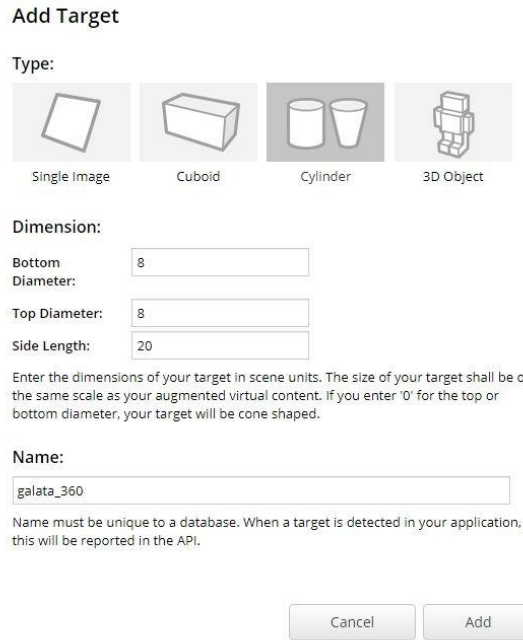
Şekil 4.20. Biblonun 7,5 derecelik açılarla çekilmiş görüntüleri.

Elde edilen görüntülerin kullanılması için Vuforia'nın yazılım geliştirme kiti kullanılarak veri tabanına kaydedilmesi gerekmektedir. Veri tabanına kaydetme işlemleri için öncelikle Vuforia'nın geliştiriciler için hazırlanmış olduğu <https://developer.vuforia.com> adresine giriş yapılarak Develop>Target Manager sekmesi altındaki Add Database seçeneği ile açılan ekrandan (Şekil 4.21) veri tabanı adını ve veri tabanı türü seçimi yapılarak veri tabanı oluşturulmalıdır.



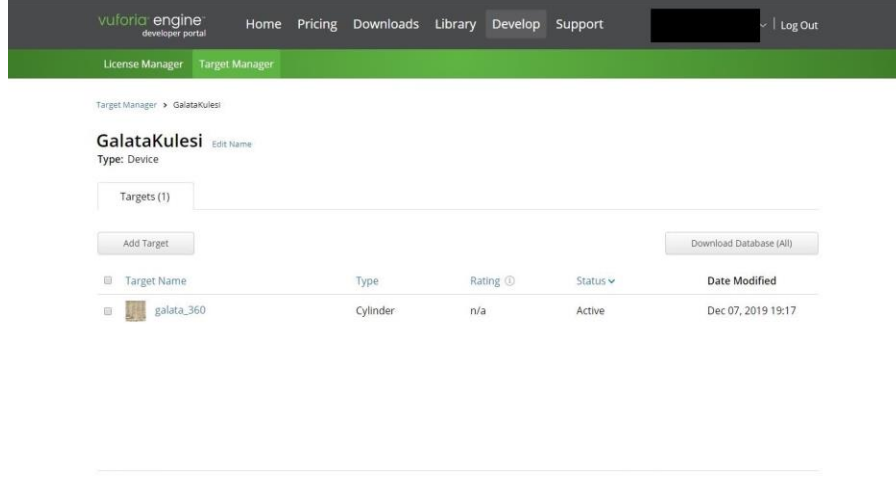
Şekil 4.21. Veri tabanı oluşturma ekranı.

Oluşturulan veri tabanına tıklanarak açılan ekranda Add Target butonu ile açılan ekranda (Şekil 4.22) tetikleyici olarak kullanılacak objenin sisteme eklenme işlemi yapılmalıdır. Açılan ekran üzerinden tür, boy ve isim ayarları yapılarak nesne oluşturulmalıdır.



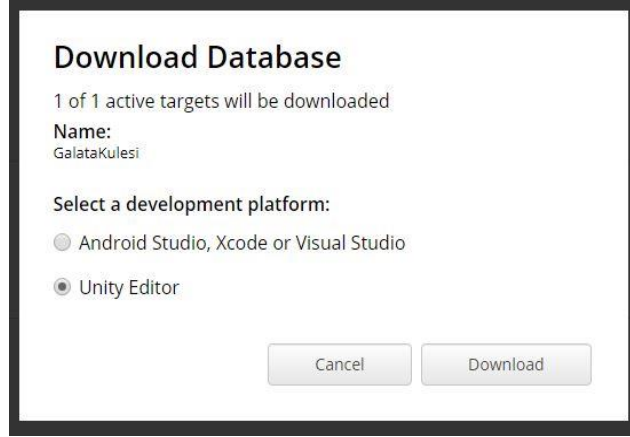
Şekil 4.22. Tetikleyici ayar ekranı.

Oluşturulan nesneye tıklanarak hazırlanmış olan görselin eklenmesiyle 3 boyutlu nesnenin sistemde tanımlanma işlemi tamamlanmaktadır. Şekil 4.23'te veri tabanı inceleme ekranı gösterilmektedir.



Şekil 4.23. Veri tabanı inceleme ekranı.

Sisteme kaydı yapılan biblonun Download Database butonu ile veri tabanı indirilerek Unity üzerinden gerçek dünyada tetikleyici olarak kullanıma uygun hale getirilmiştir. Şekil 4.24'te veri tabanı indirme ekranı gösterilmektedir.



Şekil 4.24. Veri tabanı indirme ekranı.

Unity üzerinde geliştirilen uygulamanın çalışması için Vuforia sistemi üzerinden geliştirme anahtarına ihtiyaç duyulmaktadır. Vuforia sistemi üzerinde License Manager başlığı altındaki Get Development Key butonuna tıklanarak açılan ekranda (Şekil 4.25) lisans anahtarına adı verilerek ihtiyaç duyulan anahtarın üretilmesi gerçekleştirilir.

vuforia engine developer portal Home Pricing Downloads Library Develop Support [User] Log Out

License Manager Target Manager

Back To License Manager

Add a free Development License Key

License Name *
GalataKulesi

You can change this later

License Key
Develop
Price: No Charge
Reco Usage: 1,000 per month
Cloud Targets: 1,000
VuMark Templates: 1 Active
VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the Vuforia Developer Agreement.

Cancel Confirm

Şekil 4.25. Lisans anahtarı oluşturma ekranı.

Üretilen lisansa tıklanarak, açılan ekranda (Şekil 4.26) çalışmada kullanılmak üzere üretilen lisans anahtarına ulaşılabilmektedir.

vuforia engine developer portal Home Pricing Downloads Library Develop Support [User] Log Out

License Manager Target Manager

License Manager > GalataKulesi

GalataKulesi

Edit Name Delete License Key

License Key Usage

Please copy the license key below into your app

```
AVN04AX // // AAAA... (truncated)
```




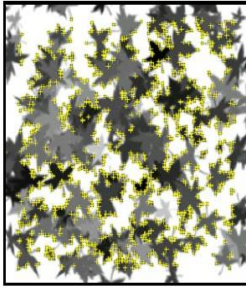
Şekil 4.26. Lisans anahtarı inceleme ekranı.

Vuforia incelendiğinde tanıma ve izleme için görüntü üzerinde kullanılacak özellikleri tanımlar ve görüntüyü gri tonlamalı olarak işleme alır. Şekil 4.27 incelendiğinde görüntüde tanımlanan özellikler anlatılmıştır.

Şekil	Açıklama
	Karede her köşe için bir tanımlayıcı içerir.
	Daire keskin veya kesikli yapısı olmadığı için tanımlayıcı içermez.
	Keskin her köşe için tanımlayıcı içerir.

Şekil 4.27. Görüntüdeki tanımlayıcılar [80].

Görüntü üzerinde ne kadar çok özellik bulunursa görüntünün tanımlanıp izlenme oranı doğru orantılı olarak artmaktadır [80].

Açıklama	Yüklenen Resim	Analiz Edilen Görüntü	Derecelendirilmesi
Az sayıda özelliğe sahip resim			☆☆☆☆
Çok sayıda özelliğe sahip resim			☆☆☆☆☆

Şekil 4.28. Özellik sayısına göre görüntü kalitesini [80].

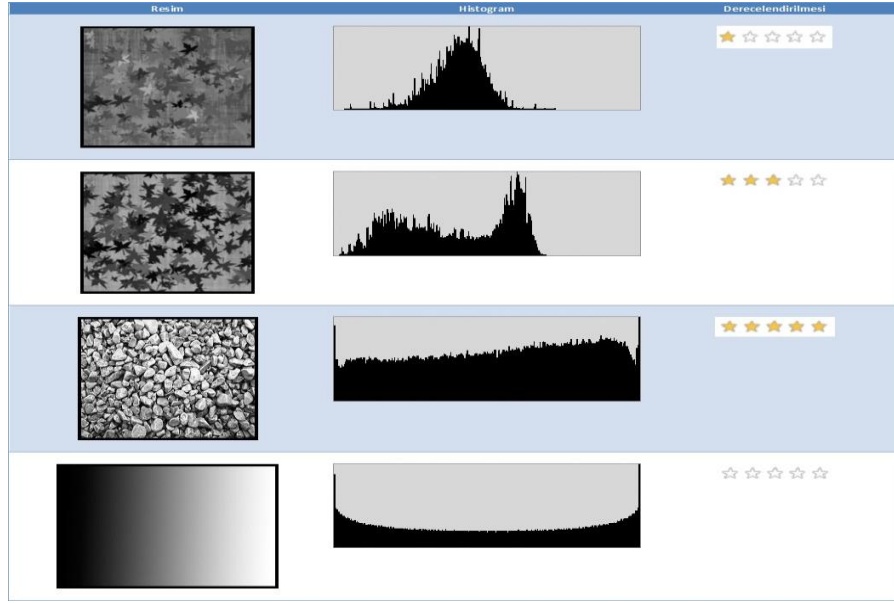
Görüntü sistemde daha iyi tanımlanabilmesi için görüntünün gri tonlamalı hali kullanılmaktadır. Bu nedenle gri tonlamalı resmin histogram verisi elde edilir. Histogram görüntünün yoğunluğunu ifade eden renk değerlerinin sayısını gösteren grafiğdir. Sistem verimli çalışabilmesi için görüntünün histogram verisinden yararlanmaktadır. Histogramın matematiksel gösterimi Formül 4.1 ile gösterilmiştir.

$$h(r_k) = n_k \quad (4.1)$$

r_k : k'inci gri deęerini

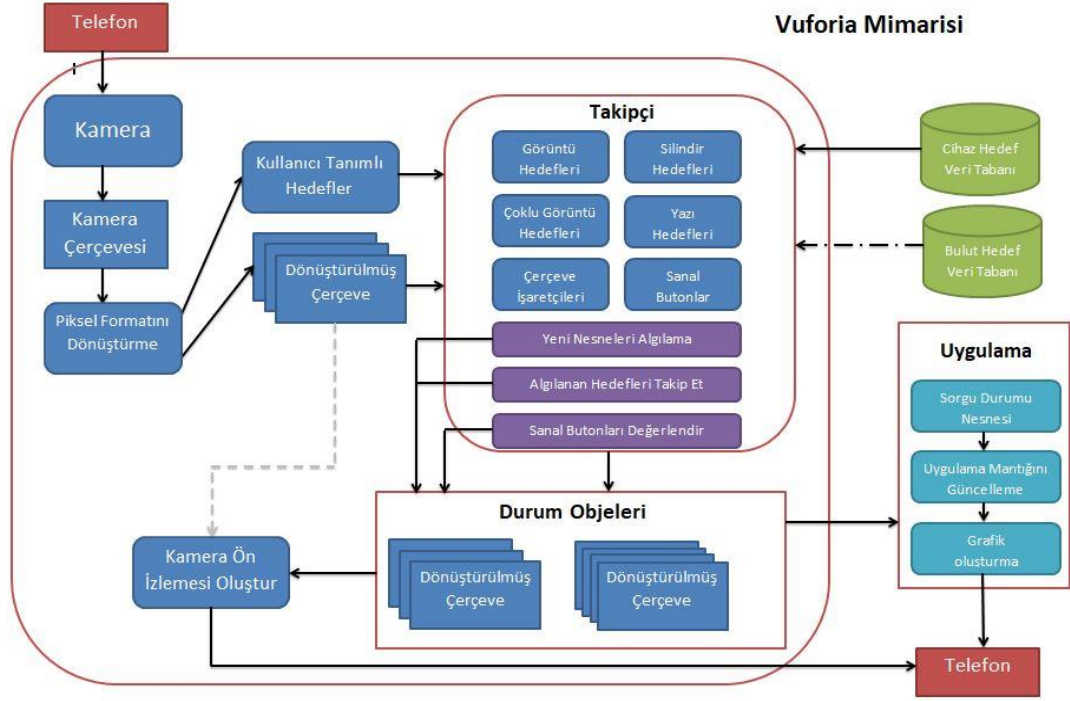
n_k : k'inci gri deęerinin grntdeki sayısı

Őekil 4.29 incelendięinde kullanılan grntnn histogram grafięinde tek bir ıkıŐ olmas veya dŐk seviyede dz ilerleme yapmas grntnn tanıma ve izlemede kalitesiz olduęunu gstermektedir. Veri daęılım yksek olmas ve geniŐ bir alana yayılması tanıma ve izlemede kaliteli olduęu gstermektedir [80].



Őekil 4.29. Histogramlara gre grnt kalitesi [80].

Unity zerinde hazırlanacak uygulamada Őekil 4.30'da gsterilen Vuforia mimarisinden yararlanarak uygulama ierisinde artırılmıŐ gereklik teknolojisinin kullanılması saęlanmıŐtır. Őekil 4.30 incelendięinde artırılmıŐ gereklik teknolojisinin iŐlem adımları ortaya konulmuŐtur.



Şekil 4.30. Vuforia mimarisi [81].

AG teknolojisinin şekli tanımlayıp izleyebilme özelliği incelendiğinde yapı ilk olarak kameradaki nesnenin pozunu almaktadır. Bir nesne pozunun altı adet bağımsız parametresi bulunur. Bunlar ise nesnenin konumu ve doğrultusunu temsil eder. Konum, koordinat düzleminde (x, y, z) ile ifade edilirken doğrultu ise koordinat düzlemindeki dönüş açıları olan (α, β, γ) ile ifade edilir.

Kameranın fiziksel özelliklerine bağlı olarak kameralardaki sensörün elde ettiği görüntünün koordinatları ile ideal görüntüde ki koordinatlar birbirlerinden farklıdır. Bu kapsamda görüntünün sistem tarafından tanımlanması için öncelikle kamera dönüşümü için işaretçi ile kamera arasında dönüşüm B ile ifade edilir. $s = B.S$ formülü ile s yerine ideal görüntü koordinatları, S yerine ise görüntüdeki koordinatlar yerleştirilir. B transformasyonu ise çeviri vektörü olan b ve 3×3 rotasyon matrisi olan N 'den oluşmaktadır. Bu kapsamda formül $s = [N \mid b]S$ olarak ifade edilebilir. Formüldeki ifadeleri yerlerine yazılması ile Formül 4.2 elde edilir [82].

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} n_1 & n_2 & n_3 & b_x \\ n_4 & n_5 & n_6 & b_y \\ n_7 & n_8 & n_9 & b_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.2)$$

Sistem işaretçiyi algıladığında her bir kare için matrisin çözülmesi gerekmektedir.

Poz hesaplama işlemlerinde işaretçi algılama, resim koordinatlarında dört köşe noktası belirler. Her köşe için s_i $i = 1,2,3,4$ olacak şekilde yazıldığında $s_i = QBS_i$ ifadesine ulaşılır. Q ifadesi kamera kalibrasyon matrisini ifade etmektedir ve günümüzdeki çoğu geliştirilmiş özellikli modern kamera için bozulmanın kamera modelinden ayrılacağı güvenli bir var sayıdır. Formüldeki ifadeleri yerlerine yazılması ile Formül 4.3 elde edilir [82].

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} c & 0 & g_x & 0 \\ 0 & c & g_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} n_1 & n_2 & n_3 & b_x \\ n_4 & n_5 & n_6 & b_y \\ n_7 & n_8 & n_9 & b_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (4.3)$$

QT = W formülü ile dünyadaki köşe noktalarının gerçek konumları hesaplanmaktadır. Formüldeki ifadeleri yerlerine yazılması ile Formül 4.4 elde edilir [82].

$$\begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_5 & w_6 & w_7 & w_8 \\ w_9 & w_{10} & w_{11} & w_{12} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (4.4)$$

Sistemde her köşedeki iki koordinattan sekiz denklem ve altı bağımsız parametre bulunmaktadır. Bu nedenle eldeki verilerle dönüşüm matrisini çözmek mümkündür.

Başlangıç tahmini için sık kullanılan yenilemeli olmayan bir yaklaşım vardır. Tahmin işleminden sonra asıl pozun hesaplama işlemleri için yinelemeli optimizasyondan yararlanır.

Tahmini dönüşüm matrisi W için hesaplama yapılırsa, tasarlama noktasına s ve dünyayı S olarak belirtirsek formül $\hat{s} = WS$ olarak gösterilir.

$\|s - \hat{s}\|$ formülü ile Formül 4.5'ten yararlanarak hataları minimize eden matris bulunarak dönüşüm matrisi hesaplanır. [82]

$$err = \frac{1}{4} \sum_{i=1,2,3,4} \{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\} \quad (4.5)$$

Sistem sürekli izleme işleminde işaretleyici verileri algılayıp çözdükten sonra içeriğin kodunu tekrardan analiz edip çözmeden işaretleyiciyi sonraki karelerde tespit etmesi yeterlidir [82].

Artırılmış gerçeklikteki sistemin amacı, bilgisayar ortamında oluşturulmuş sanal nesneyi gerçek dünya ortamının bir parçası gibi kullanıcıya sunmaktır. Sistem yakaladığı poz ile bilgisayar ortamında oluşturulan sanal nesneyi ayarlanmış fiziksel özelliklerine göre oluşturmakta kullanır.

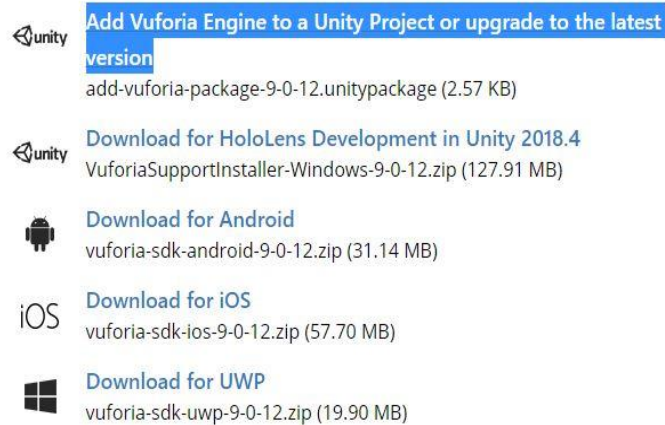
Oluşturulacak sanal nesnede herhangi bir konumsal ayarlama yapılmadıysa, sanal nesne başlangıç noktasının koordinatlarını temel alınarak gösterilir. Eğer sanal nesneye konumsal ayarlama yapılmışsa sanal nesne gösteriminde yapılan ayarlamalar dikkate alınarak gösterimi yapılır [82].

4.6. ARTIRILMIŞ GERÇEKLIK UYGULAMASININ GELİŞTİRİLMESİ

Unity ile artırılmış gerçeklik üzerine geliştirme yapabilmek için Vuforia üzerinden hazırlanmış olan işaretçiler paketi ile yazılım geliştirme kiti indirilmesi gerekmektedir. Kite ulaşılabilmek için <https://developer.vuforia.com/downloads/sdk> adresi üzerinden Şekil 4.31'deki gösterilen Add Vuforia Engine to a Unity Project or upgrade to the latest version seçeneği seçilerek gerekli dosyaların indirme işlemleri gerçekleştirilir.

Vuforia Engine 9.0

Use Vuforia Engine to build Augmented Reality Android, iOS, and UWP applications for mobile devices and AR glasses. Apps can be built with Unity, Android Studio, Xcode, and Visual Studio. Vuforia Engine can also be accessed through the Unity Package Manager by adding Vuforia's package repository with the script below.



add-vuforia-package-9-0-12.unitypackage (2.57 KB)

Download for HoloLens Development in Unity 2018.4
VuforiaSupportInstaller-Windows-9-0-12.zip (127.91 MB)

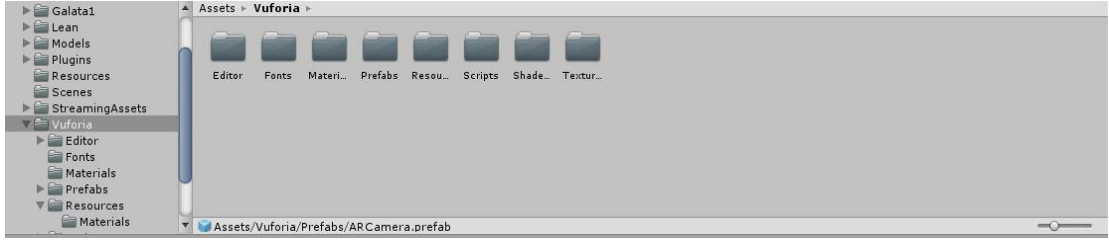
Download for Android
vuforia-sdk-android-9-0-12.zip (31.14 MB)

Download for iOS
vuforia-sdk-ios-9-0-12.zip (57.70 MB)

Download for UWP
vuforia-sdk-uwp-9-0-12.zip (19.90 MB)

Şekil 4.31. Vuforia kütüphanesini indirme ekranı.

İndirilen dosyaları Assets>ImportPackage>CustomPackage yoluyla oluşturulan projeye entegre edilmesi ile Vuforia kütüphanesinin sağlamış olduğu araçlar Assets>Vuforia klasörü altında Şekil 4.32'deki gibi eklenecektir.



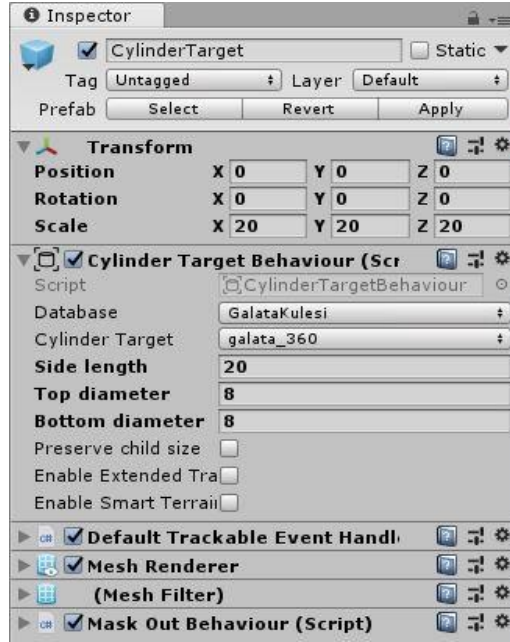
Şekil 4.32. Unity üzerinde Vuforia nesneleri.

Eklene dosyalarda bulunan Prefabs klasörü altındaki ARCamera nesnesi sahneye eklenmesi gerekmektedir. Eklene ARCamera nesnesine tıklanarak Inspector sekmesinde bulunan Open Vuforia configuration butonu seçilmesiyle Vuforia dosyalarının kullanılabilmesi için yapılması gereken ayarlar kısmı açılacaktır. Açılan ekranda Vuforia başlığı altındaki App License Key alanı Vuforia sisteminde üretilen lisans anahtarı ile doldurulmalıdır. Eklene veri tabanını aktifleştirmek için datasets başlığı altında bulunan veri tabanı isminin yanında tik işareti ile altındaki Activate isminin yanındaki tik işareti aktifleştirilmelidir. Bu işlemlerle ARCamera nesnesinin ayarları tamamlanmış olunur. Şekil 4.33'te artırılmış gerçeklik kamera ayarları gösterilmektedir.



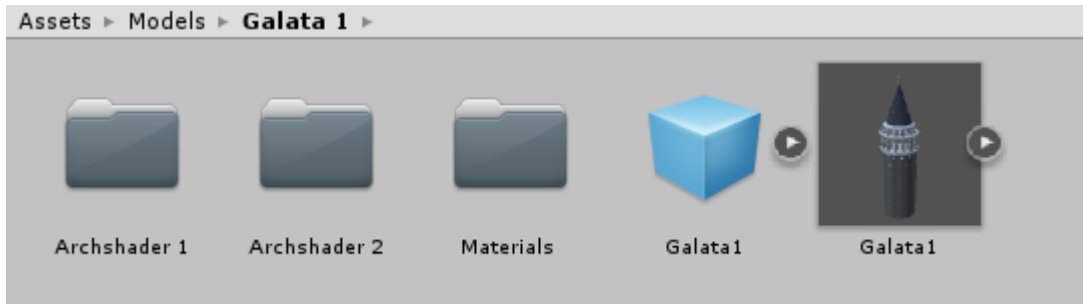
Şekil 4.33. Artırılmış kamera ayarları.

Biblonun verilerini program içerisinde kullanılması için Assets>Vuforia>Prefabs klasörü içerisindeki CylinderTarget nesnesi sahneye eklenmesi gerekmektedir. Eklenen nesneye tıklanarak Inspector sekmesinde Cylinder Target Behaviour başlığı altındaki Database kısmında eklenen veri tabanı seçilmeli ve Cylinder Target kısmında ise tanımlanan görselin ismi seçilmelidir. Şekil 4.34'te Cylinder Target nesnesinin ayarları gösterilmektedir.



Şekil 4.34. CylinderTarget nesnesinin ayarları.

Üzerindeki işlemleri tamamlanan CylinderTarget nesnesine sanal ortamda hazırlanmış olan Galata Kulelerinin modelleri entegre edilmesi gerekmektedir. Hazırlanan modeller proje içerisinde yeni oluşturulan Models klasörüne yüklenmelidir. Şekil 4.35'te Unity üzerine eklenen 1967 yılındaki Galata Kulesi'nin model dosyaları gösterilmektedir.



Şekil 4.35. 1967 yılındaki Galata Kulesi'nin model dosyaları.

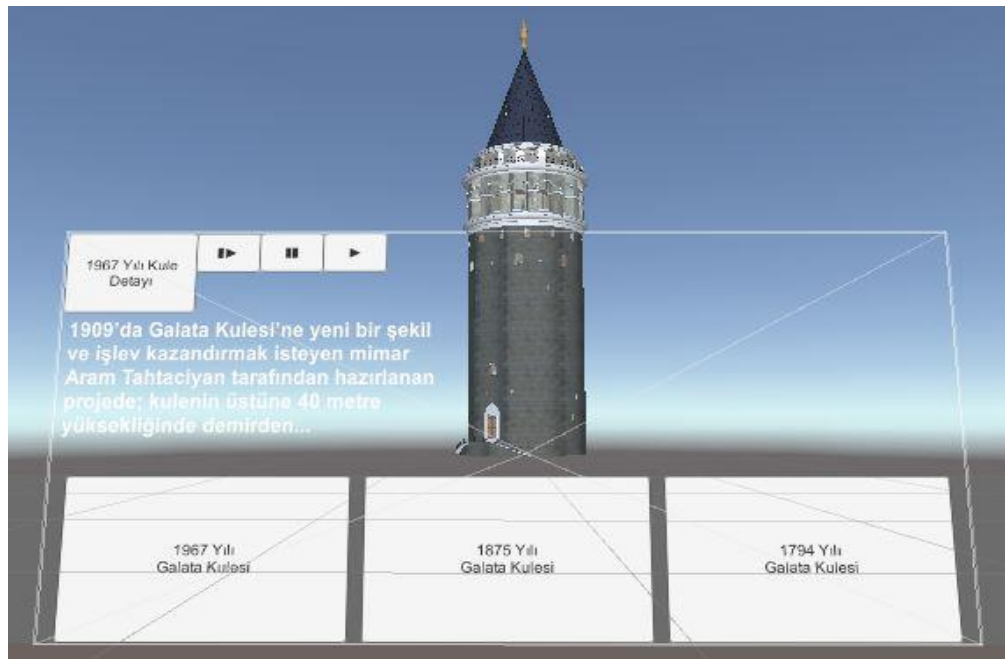
Kullanıma hazır hale getirilen modeller CylinderTarget nesnesine bağlanması gerekmektedir. Eklenen modellerden birinci model dışındaki diğer modellerin Inspector

sekmesinden aktiflik tiki kaldırılmalıdır. Bu sayede uygulama çalıştırıldığında kullanıcı modelleri tek tek inceleyebilecektir. Modellerin eklenmesi ile sahnede Şekil 4.36'daki görüntü elde edilmektedir.



Şekil 4.36. Uygulamaya eklenen Galata Kulesi modeli.

Uygulama içerisinde sahneye modeller arasında geçiş yapılabilmesi için üç adet buton, bilgilendirme kısmının sesli olarak kullanımı için üç adet buton, bilgilendirme metninin açılıp kapanması için bir adet buton ve bilgilendirme metni için bir adet metin alanı eklenmelidir. Ekleme işlemleri sonucunda Şekil 4.47'deki görüntü elde edilmektedir.



Şekil 4.37. Butonlar eklenmiş uygulama ekranı.

Kameranın odağının aktif olarak netleştirilmesi için projeye focus adında C# scripti oluşturulmalıdır. Oluşturulan script içerisine Şekil 4.38'deki kodlar eklenmelidir.

```
using UnityEngine;
using System.Collections;
using Vuforia;

public class focus : MonoBehaviour
{
    //Kamerayı odaklama için kullanılacak nesne
    private bool mVuforiaStarted = false;

    //Başlangıçta kullanılacak nesnelere
    void Start()
    {
        VuforiaARController vuforia = VuforiaARController.Instance;

        if (vuforia != null)
            vuforia.RegisterVuforiaStartedCallback(StartAfterVuforia);
    }

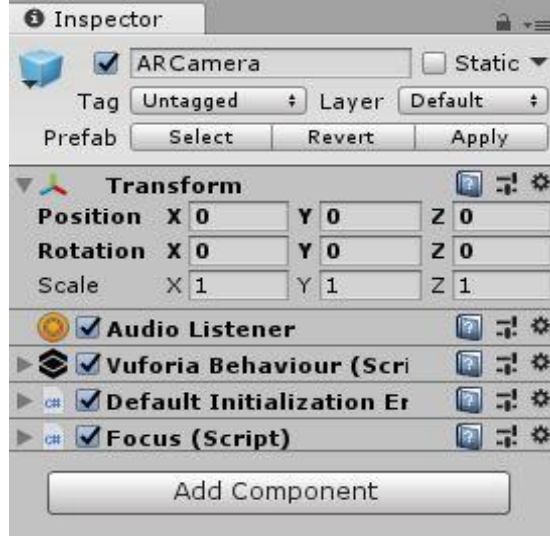
    //Eğer kamera başlatılmışsa kamerayı otomatik odakla
    private void StartAfterVuforia()
    {
        mVuforiaStarted = true;
        SetAutofocus();
    }

    //Uygulama durdurulmadıysa otomatik odaklamanın bakıldığı alan
    void OnApplicationPause(bool pause)
    {
        if (!pause)
        {
            if (mVuforiaStarted)
            {
                SetAutofocus();
            }
        }
    }

    //Kameranın odaklamayı destekleyip desteklemediğine bakıldığı alan
    private void SetAutofocus()
    {
        if (CameraDevice.Instance.SetFocusMode(CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO))
        {
            Debug.Log("Autofocus set");
        }
        else
        {
            Debug.Log("this device doesn't support auto focus");
        }
    }
}
```

Şekil 4.38. Focus scriptinin kodu.

Oluşturulan script ArCamera nesnesine tanımlanarak uygulamanın aktif olarak odağının netleştirilmesi sağlanmaktadır. Şekil 4.39'da ArCamera'nın inspector ekranı gösterilmektedir.



Şekil 4.39. ArCamera'nın inspector ekranı.

Modeller arasında geçiş yapılabilmesi için projeye GatalaSelector adında C# scripti oluşturulmalıdır. Oluşturulan script içerisine Şekil 4.40'daki kodlar eklenmelidir.

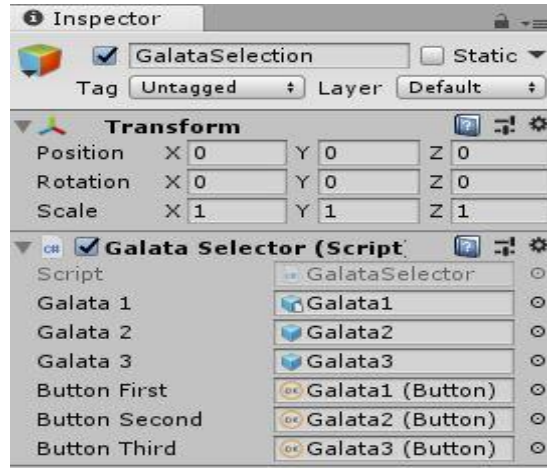
```

using UnityEngine;
using System.Collections;
using UnityEngine.UI;
public class GalataSelector : MonoBehaviour
{
    //Galata kulesi modellerinin tanımlanması için kullanılan nesneler
    public GameObject Galata1;
    public GameObject Galata2;
    public GameObject Galata3;
    //İlgili ekranda kullanılacak butonlar
    public Button buttonFirst;
    public Button buttonSecond;
    public Button buttonThird;
    //Başlangıçta 2. ve 3. modelleri aktifliğini kapat
    void Start()
    {
        //Başlangıçta modellerin aktifliğini kapatma
        Galata2.SetActive(false);
        Galata3.SetActive(false);
        //Butonlara tıklanma olayı
        buttonFirst.onClick.AddListener(() => LoadGalata1());
        buttonSecond.onClick.AddListener(() => LoadGalata2());
        buttonThird.onClick.AddListener(() => LoadGalata3());
    }
    //1. butona tıklanmasıyla 1. modeli aktif et, 2. ve 3. modelleri kapat
    public void LoadGalata1()
    {
        Galata1.SetActive(true); Galata2.SetActive(false); Galata3.SetActive(false);
    }
    //2. butona tıklanmasıyla 2. modeli aktif et, 1. ve 3. modelleri kapat
    public void LoadGalata2()
    {
        Galata1.SetActive(false); Galata2.SetActive(true); Galata3.SetActive(false);
    }
    //3. butona tıklanmasıyla 3. modeli aktif et, 1. ve 2. modelleri kapat
    public void LoadGalata3()
    {
        Galata1.SetActive(false); Galata2.SetActive(false); Galata3.SetActive(true);
    }
}

```

Şekil 4.40. GalataSelector scriptinin kodu.

Script projede kullanılabilmesi için uygulamaya GalataSelection adında GameObject eklenmeli ve eklenen nesneye hazırlanan script tanımlanmalıdır. Eklenen GameObject nesnesine tıklanarak Inspector sekmesinde hazırlanan scriptte ilgili modeller ve butonlar tanımlanmalıdır. Şekil 4.41’de GalataSelection’ın inspector ekranı gösterilmektedir.



Şekil 4.41. GalataSelection’ın inspector ekranı.

Bilgilendirme alanının her modele göre çalışabilmesi için InfoButton adında C# scripti oluşturulmalıdır. Oluşturulan script içerisine Şekil 4.42, Şekil 4.43 ve Şekil 4.44’teki kodlar eklenmelidir.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class InfoButton : MonoBehaviour
{
    //Buton üzerinde yazı ve bilgilendirme alanı için kullanılacak nesnelere
    public Text InfoText;
    public Text InfoButtonText;

    //Butonların tıklanma durumunu takip için kullanılacak nesnelere
    public ButtonType currentButtonType;

    //İlgili ekranda kullanılacak butonlar
    public Button buttonFirst;
    public Button buttonSecond;
    public Button buttonThird;
    public Button buttonInfo;

    //Veri tekrarını engellemek için kullanılacak yazıların belirlenmesi
    private const string galata1InfoText = "1909'da Galata Kulesi'ne yeni bir şekil ve işlev kazandırmak isteyen mimar Aram Tahtacıyan tarafından hazırlanan projede; kulenin üstüne 40 metre yüksekliğinde demirden yapılmış büyük bir parça ilave edilmesi öngörülmüş, ilave edilen parçanın küre şeklinde bir kısım ile onun üstünde silindirik olarak yükselen ikinci bir kısımdan meydana gelmesi, silindirik şeklindeki kısmın telefon ve telsiz telgrafla donatılarak, telgrafhane, rasathane ve yangın gözetleme amaçlı kullanılması, küre şeklindeki kısmın ise projeyi sunan işletmeciyeye tahsis edilmesi teklif edilmiştir. İleriki yıllarda ülkede radyo yayıncılığının gelişmesiyle birlikte bu uygulamadan vazgeçilmiş, 1967'de de kulenin tepesine yeni bir kule geçirilerek bugünkü halini almıştır.";

    private const string galata2InfoText = "Sultan II. Mahmut tarafından yeniden yaptırılan sivri kulehının 1875'de fırtınadan uçup gitmesinden sonra tepesine iki küçük katçık ve bunların ortasına dikilen uzun bir bayrak direğiyle şekillenen Galata Kulesi uzun yıllar bu haliyle kalmıştır.";
}
```

Şekil 4.42. InfoButton scriptinin kodu- 1.

private const string galata3InfoText = "Yapısal deęişikliklerini incelediđimizde ise Galata kulesi 1794 yılında restorasyon işlemleri yapılmıştır. III. Selim zamanında çıkan yangın nedeniyle büyük bölümü zarar görmüştür. Bu yangında kulenin tepesindeki külahı tamamıyla yanmıştır. Yangın sonrası kule tekrardan tadilata alınarak kullanılır hale getirilmiştir. Tadilat sonrasında kulenin tepesine tekrardan bir külah konulmuştur ve en üst kısmına dört tarafa çıkıntı şeklinde alanlar eklenmiştir.";

```
//Başlangıçtaki ayarlamalar
private void Start()
{
    //Bilgilendirmeyi boşaltma
    InfoText.text = "";

    //Butonlara tıklanma olayı atama
    buttonFirst.onClick.AddListener(() => Galata1Event());
    buttonSecond.onClick.AddListener(() => Galata2Event());
    buttonThird.onClick.AddListener(() => Galata3Event());
    buttonInfo.onClick.AddListener(() => ButtonInfo());
}

//1. butona tıklanma ile gerçekleştirilecek olaylar
public void Galata1Event()
{
    //1. butona göre bilgilendirme butonu üzerindeki yazıyı
    InfoButtonText.text = "1967 Yılı Kule Detayı";

    //1. butona tıkladığını belirt
    currentButtonType = ButtonType.first;

    //Eğer bilgilendirme açıksa 1. butonun bilgilendirmesi ile doldur
    if (InfoText.text != "")
        InfoText.text = galata1InfoText;
}

//2. butona tıklanma ile gerçekleştirilecek olaylar
public void Galata2Event()
{
    //2. butona göre bilgilendirme butonu üzerindeki yazıyı
    InfoButtonText.text = "1875 Yılı Kule Detayı";

    //2. butona tıkladığını belirt
    currentButtonType = ButtonType.second;

    //Eğer bilgilendirme açıksa 2. butonun bilgilendirmesi ile doldur
    if (InfoText.text != "")
        InfoText.text = galata2InfoText;
}

//3. butona tıklanma ile gerçekleştirilecek olaylar
public void Galata3Event()
{
    //3. butona göre bilgilendirme butonu üzerindeki yazıyı
    InfoButtonText.text = "1794 Yılı Kule Detayı";

    //3. butona tıkladığını belirt
    currentButtonType = ButtonType.third;

    //Eğer bilgilendirme açıksa 3. butonun bilgilendirmesi ile doldur
    if (InfoText.text != "")
        InfoText.text = galata3InfoText;
}
```

Şekil 4.43. InfoButton scriptinin kodu- 2.

```

//Bilgilendirme alanının kontrolü
public void Info(ButtonType buttonType)
{
    switch (buttonType)
    {
        //1. butona tıklanarak bilgilendirme butonuna tıklanırsa çalışacak olaylar
        case ButtonType.first:

            //Eğer bilgilendirme yokken tıklanırsa bilgilendirme ile doldur
            if (InfoText.text == "")
                InfoText.text = galata1InfoText;

            //Eğer bilgilendirme var iken tıklanırsa alanı boşalt
            else
                InfoText.text = "";
            break;

        //2. butona tıklanarak bilgilendirme butonuna tıklanırsa çalışacak olaylar
        case ButtonType.second:

            //Eğer bilgilendirme yokken tıklanırsa bilgilendirme ile
            if (InfoText.text == "")
                InfoText.text = galata2InfoText;

            //Eğer bilgilendirme var iken tıklanırsa alanı boşalt
            else
                InfoText.text = "";
            break;

        //3. butona tıklanarak bilgilendirme butonuna tıklanırsa çalışacak olaylar
        case ButtonType.third:

            //Eğer bilgilendirme yokken tıklanırsa bilgilendirme ile doldur
            if (InfoText.text == "")
                InfoText.text = galata3InfoText;

            //Eğer bilgilendirme var iken tıklanırsa alanı boşalt
            else
                InfoText.text = "";
            break;
    }
}

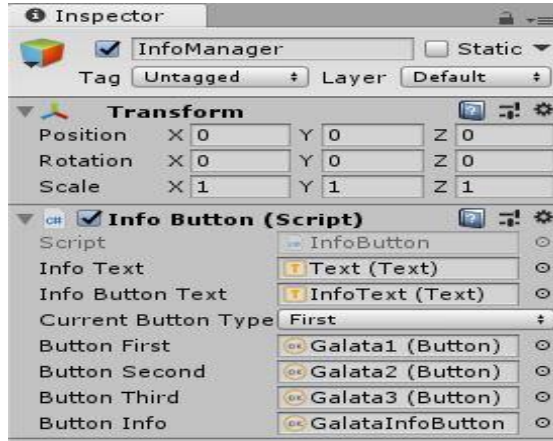
//Bilgilendirme butonuna tıklanma durumunda çalışacak olaylar
public void ButtonInfo()
{
    Info(currentButtonType);
}
}

//Buton değişikliğinin takibi için kullanılan yardımcı tanımlama
public enum ButtonType { first, second, third }

```

Şekil 4.44. InfoButton scriptinin kodu- 3.

Script projede kullanılabilmesi için uygulamaya InfoManager adında GameObject eklenmeli ve eklenen nesneye hazırlanan script tanımlanmalıdır. Hazırlanan InfoManager nesnesine tıklanarak Inspector sekmesinde hazırlanan scriptte ilgili butonlar ve metin alanları tanımlanmalıdır. Şekil 4.45'te Info Manager'ın inspector ekranı gösterilmektedir.



Şekil 4.45. InfoManager'ın inspector ekranı.

Sesli bilgilendirme alanının çalışabilmesi için kullanılacak ses dosyaları projeye eklenmeli ve SoundEffect adında C# scripti oluşturulmalıdır. Oluşturulan script içerisine Şekil 4.46 ve Şekil 4.47'deki kodlar eklenmelidir.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class SoundEffect : MonoBehaviour
{
    //Ses efekti için kullanılacak nesnelere
    public AudioSource audioSource;
    public List<AudioClip> audioClips;

    //Butonların tıklanma durumunu takip için kullanılacak nesnelere
    public ButtonType currentButtonType;

    //İlgili ekranda kullanılacak butonlar
    public Button buttonFirst;
    public Button buttonSecond;
    public Button buttonThird;
    public Button buttonStart;
    public Button buttonStop;
    public Button buttonPlay;

    //Başlangıçtaki ayarlamalar
    private void Start()
    {
        //Butonlara tıklanma olayı atama
        buttonFirst.onClick.AddListener(() => ButtonFirst());
        buttonSecond.onClick.AddListener(() => ButtonSecond());
        buttonThird.onClick.AddListener(() => ButtonThird());
        buttonStart.onClick.AddListener(() => ButtonStart());
        buttonStop.onClick.AddListener(() => ButtonStop());
        buttonPlay.onClick.AddListener(() => ButtonPlay());
    }

    //1. modelin butonuna tıklanma ile gerçekleştirilecek olaylar
    public void ButtonFirst()
    {
        //1. modelin butonuna tıkladığını belirt
        currentButtonType = ButtonType.first;
        //Çalan bir ses varsa durdur
        audioSource.Stop();
    }
}

```

Şekil 4.46. SoundEffect scriptinin kodu- 1.

```

//2. modelin butonuna tıklanma ile gerçekleştirilecek olaylar
public void ButtonSecond()
{
    //2. modelin butonuna tıkladığını belirt
    currentButtonType = ButtonType.second;
    //Çalan bir ses varsa durdur
    audioSource.Stop();
}

//3. modelin butonuna tıklanma ile gerçekleştirilecek olaylar
public void ButtonThird()
{
    //3. modelin butonuna tıkladığını belirt
    currentButtonType = ButtonType.third;
    //Çalan bir ses varsa durdur
    audioSource.Stop();
}

//Sesi başlatma işlemleri
public void PlaySound(ButtonType buttonType)
{
    //Çalan bir ses varsa durdur
    audioSource.Stop();

    //Eğer ses bekletilmeye alındıysa aktif et
    if (AudioListener.pause)
        AudioListener.pause = false;

    //Çalacak sesin kontrolü
    switch (buttonType)
    {
        //1. modelin butonuna tıklanarak başlat butonuna tıklanırsa çalışacak olaylar
        case ButtonType.first:
            //Atanmış ilk ses dosyasını çalıştır.
            audioSource.PlayOneShot(audioClips[0]);
            break;

        //2. modelin butonuna tıklanarak başlat butonuna tıklanırsa çalışacak olaylar
        case ButtonType.second:
            //Atanmış ilk ses dosyasını çalıştır.
            audioSource.PlayOneShot(audioClips[1]);
            break;

        //3. modelin butonuna tıklanarak başlat butonuna tıklanırsa çalışacak olaylar
        case ButtonType.third:
            //Atanmış ilk ses dosyasını çalıştır.
            audioSource.PlayOneShot(audioClips[2]);
            break;
    }
}

//Çalan sesi duraklatma işlemleri
public void ButtonStop()
{
    AudioListener.pause=true;
}

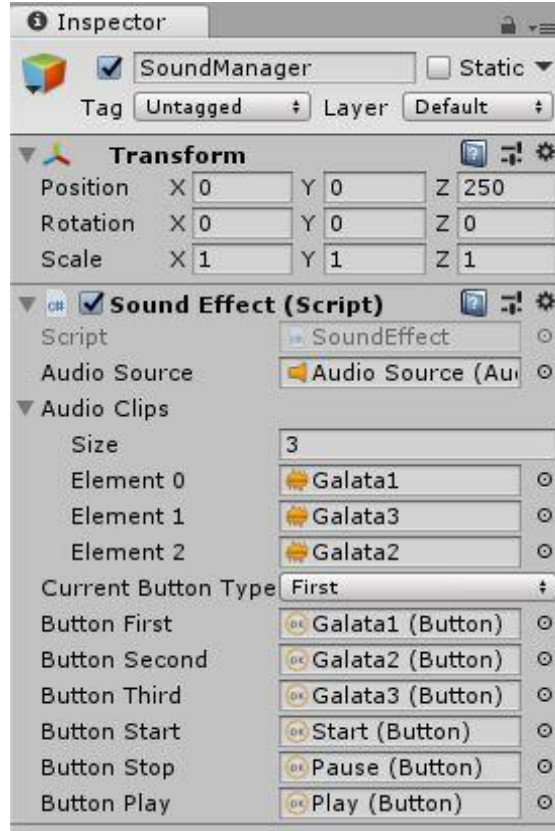
//Duraklatılan sesi devam ettirme işlemleri
public void ButtonPlay()
{
    AudioListener.pause = false;
}

//Sesi başlatma işlemleri
public void ButtonStart()
{
    PlaySound(currentButtonType);
}
}
//Buton değişikliğinin takibi için kullanılan yardımcı tanımlama
public enum ButtonType { first,second,third}

```

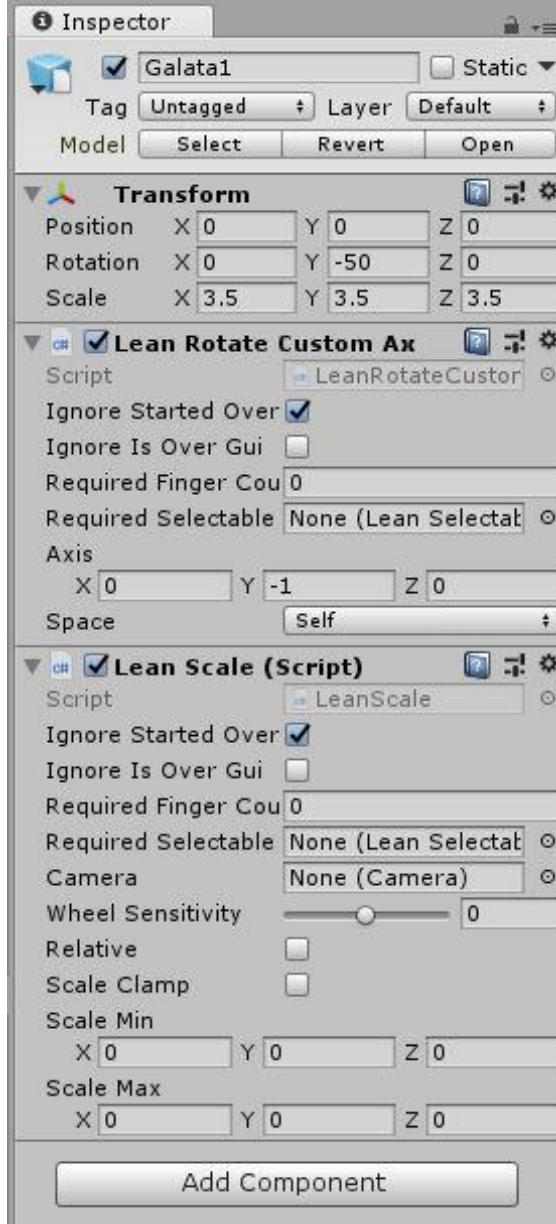
Şekil 4.47. SoundEffect scriptinin kodu- 2.

Script projede kullanılabilmesi için uygulamaya SoundManager adında GameObject nesnesi ve Audio Source nesnesi eklenmelidir. Hazırlanan SoundManager nesnesine script tanımlanmalıdır. SoundManager nesnesine tıklanarak Inspector sekmesinde hazırlanan scriptte ilgili butonlar, eklenen ses kaynağı ve kullanılacak sesleri ilgili alanlara tanımlanmalıdır. Şekil 4.48’de Sound Manager’ın inspector ekranı gösterilmektedir.



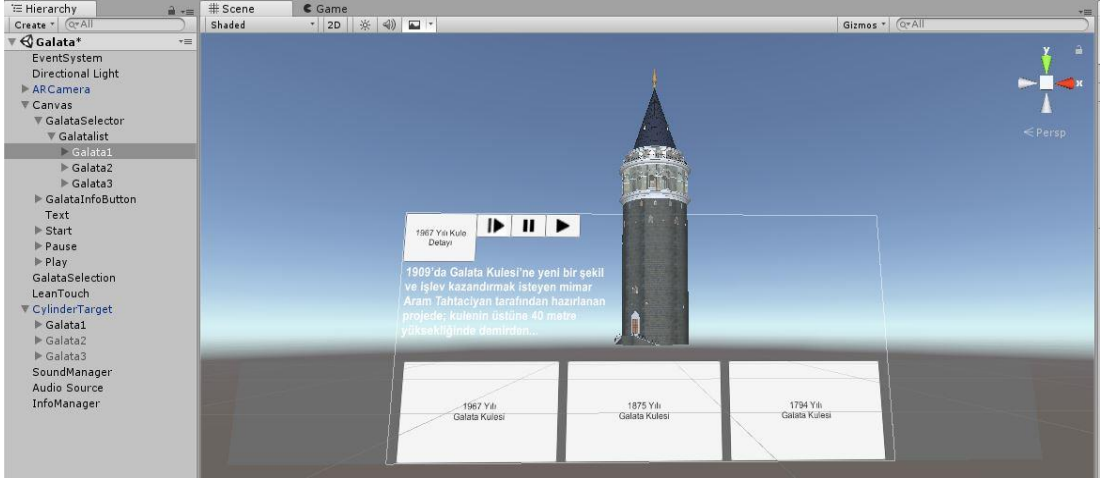
Şekil 4.48. SoundManager’ın inspector ekranı.

Oluşturulan Galata Kulesi modellerinin daha rahat incelenebilmesi için Unity içerisinde Window>General>Asset Store giriş yapılarak LeanTouch eklentisi eklenmelidir. Eklentinin sağlamış olduğu LeanTouch nesnesi sahneye eklenerek modellerin yakınlaştırılma ve kendi etrafında döndürülme özellikleri eklenilmesine olanak sağlanılmıştır. Projeye eklenen Galata Kulesi modellerinin Inspector sekmesi altındaki Add Component butonuna tıklanarak Lean Rotate ve Lean Scale scriptleri modellere eklenir. Şekil 4.49’da Galata1’in inspector ekranı gösterilmektedir.



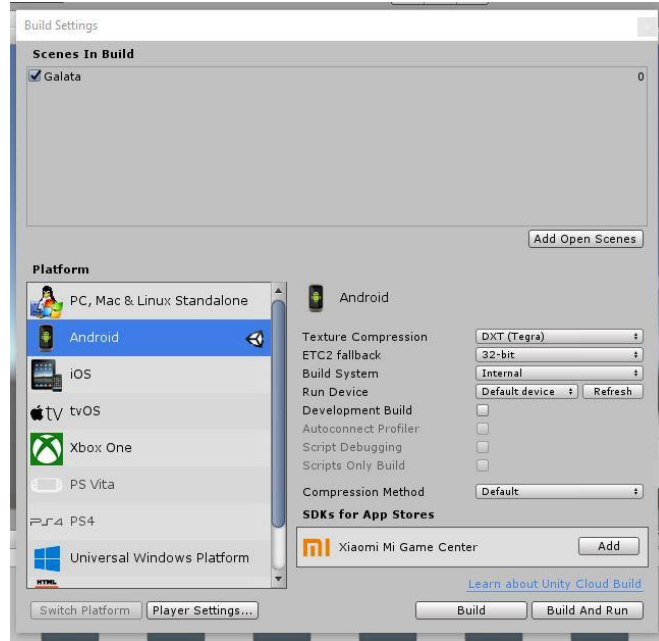
Şekil 4.49. Galata1'in inspector ekranı.

Yapılan adımlar sonrasında uygulama geliştirme işlemleri tamamlanmıştır. Şekil 4.50'de uygulamanın son hali gösterilmektedir.



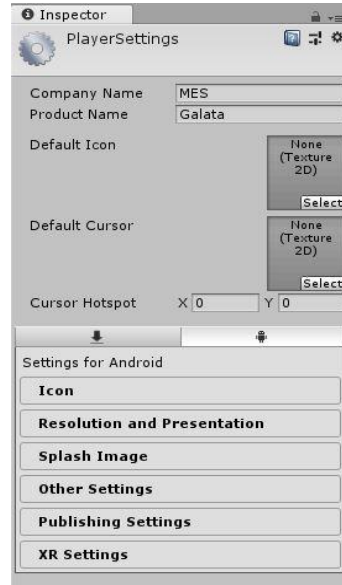
Şekil 4.50. Uygulamanın son hali.

Uygulamayı tamamlamak için son adım olan proje yapılandırılması gerekmektedir. File>BuildSettings seçeneği ile açılan pencerede gerekli Android ayarlamaları yapılmalıdır. Şekil 4.51’de Buildg Settings ayarları gösterilmektedir.



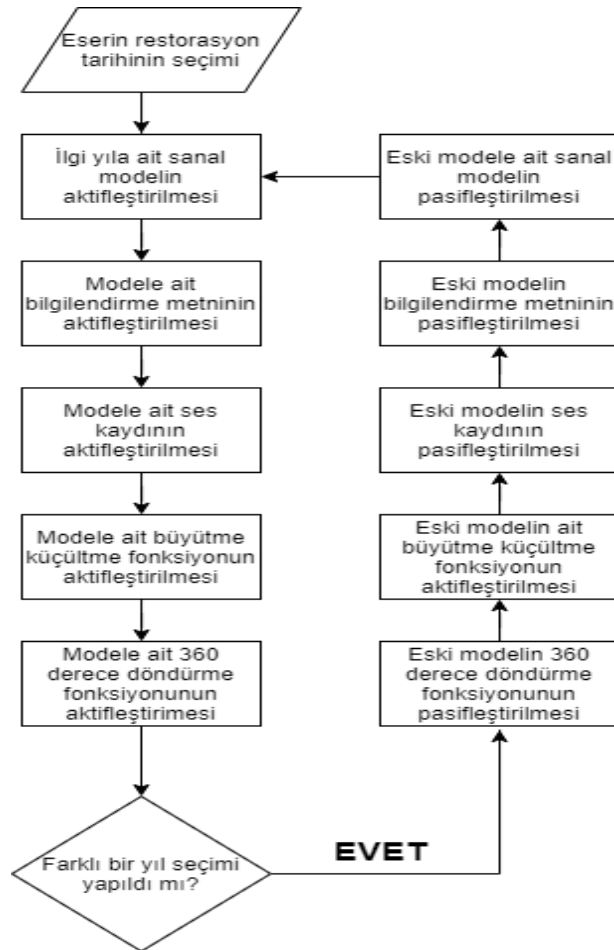
Şekil 4.51. Build Settings ayarları.

Player Settings butonu aracılığı ile Inspector sekmesinde açılan alanda Company Name, Product Name gibi zorunlu alanlar doldurularak Build Settings penceresinde Build butonu ile uygulamanın apk’sı elde edilir. Şekil 4.52’de Player Settings ayarları gösterilmektedir.



Şekil 4.52. Player Settings ayarları.

Hazırlanan uygulama incelendiğinde AG tarafının akış diyagramı Şekil 4.53 üzerinde ortaya konulmuştur.



Şekil 4.53. AG tarafının akış diyagramı.

4.7. UYGULAMAYA YENİ MODEL EKLENMESİ

Bilgisayar ortamında hazırlanmış olan yeni model uygulamaya eklenmelidir. Eklenen model hazırlanan yapıda CylinderTarget nesnesine bağlanmalıdır. Bağlanan modele tıklanarak Inspector sekmesinden aktifliği kapatılmalıdır. Uygulama içerisinde modelin kullanılabilmesi için sahneye yeni bir buton eklenmelidir.

Yeni modelin butonlarla değişimini sağlamak için GalataSelector scripti içerisinde, eklenen modelin tanımlanması için bir adet buton ve gameobject tanımlaması yapılmalıdır. Diğer modeller için hazırlanan metotların işlevine uygun olarak yeni model metotlara eklenmelidir. Uygulamadaki modellere hazırlanan metotlar gibi yeni eklenen modelin aktifliğini düzenleyen metot eklenmelidir. Oluşturulan butonun tıklanma durumuna yeni eklenen metot tanımlanmalıdır. Scripte eklenen yeniliklerin kullanılabilmesi için GalataSelection'a tıklanarak inspector sekmesinde sahneye yeni eklenen buton ve model tanımlanmalıdır.

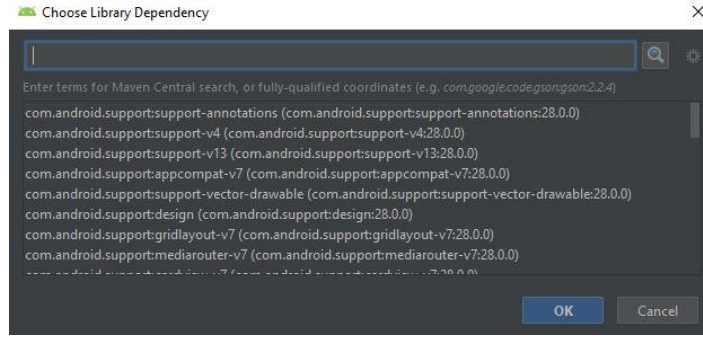
Yeni modelin bilgilendirme metni ile bağlantısını sağlamak için InfoButton scriptine bir adet buton tanımlanması yapılmalıdır. Script içerisinde diğer modeller için hazırlanan bilgilendirme metni formatını temel alarak yeni model için de bilgilendirme metni eklenmelidir. Uygulamadaki modellere hazırlanan metotlar gibi yeni eklenen modelin bilgilendirmesini yapan metot eklenmelidir ve bilgilendirme alanının kontrolünü sağlayan metot düzenlenmelidir. Oluşturulan butonun tıklanma durumuna yeni eklenen metot tanımlanmalıdır. Scripte eklenen yeniliklerin kullanılabilmesi için InfoManager'e tıklanarak inspector sekmesinde sahneye yeni eklenen buton tanımlanmalıdır.

Yeni modelin ses özelliğini oluşturmak için hazırlanmış olan ses kaydı uygulamaya eklenmelidir. Ses kaydının kullanılabilmesi için SoundEffect scriptine bir adet buton tanımlanması yapılmalıdır. Script içerisinde diğer modeller için hazırlanan metotlar gibi yeni eklenen modelin ses kaydını düzenleyen metot eklenmelidir ve çalan sesin kontrolü sağlayan metot yeni ses için düzenlenmelidir. Oluşturulan butonun tıklanma durumuna yeni eklenen metot tanımlanmalıdır. Scripte eklenen yeniliklerin kullanılabilmesi için SoundManager'e tıklanarak inspector sekmesinde sahneye yeni eklenen butonun tanımlanması ve AudioClip altındaki Size özelliği 3'ten 4'e çekilerek Element 3 yerine uygulamaya eklenen yeni ses tanımlanmalıdır.

Son olarak sahneye eklenen modele tıklanarak Inspector sekmesi altındaki Add Component butonu ile Lean Rotate ve Lean Scale scriptleri modelle eklenmelidir.

4.8. HARİTA UYGULAMASININ GELİŞTİRİLMESİ

Uygulamanın başlangıç noktası olan harita sisteminin kullanılabilmesi için Android Studio üzerinden Google harita projesi açılmalıdır. Uygulama içerisinde konum bilgilerinden yararlanabilmek için File>Project Structure seçeneği ile açılan pencereden Dependencies kısmından Library Dependency seçeneği ile com.google.android.gms:play-services-location servisi Şekil 4.54'te gösterilen alandan projeye eklenmelidir.



Şekil 4.54. Library Dependency ekranı.

Harita uygulamasında sistemin yapısını oluşturmak için destekleyici yapılar kurulmalıdır. Birden fazla yerde kullanılması için yardımcı metotların yazılacağı TaskLoadedCallback.java adında Java dosyası oluşturularak dosya içerisine Şekil 4.55'teki kodlar eklenmelidir.

```
package com.galatakulesi.mes;

public interface TaskLoadedCallback {
    void onTaskDone(Object... values);
}
```

Şekil 4.55. Task Loaded Callback kodu.

Uygulamada oluşturulan yolların güzergâhını belirlemek ve verilerin işlenecek hale getirmesi için DataParser.java adında Java dosyası oluşturularak dosya içerisine Şekil 4.56 ve Şekil 4.57'deki kodlar eklenmelidir.

```

package com.galatakulesi.mes;
import com.google.android.gms.maps.model.LatLng;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class DataParser {

    public List<List<HashMap<String, String>>> parse(JSONObject jObject) {

        List<List<HashMap<String, String>>> routes = new ArrayList<>();
        JSONArray jRoutes;
        JSONArray jLegs;
        JSONArray jSteps;

        try {

            jRoutes = jObject.getJSONArray("routes");

            for (int i = 0; i < jRoutes.length(); i++) {

                jLegs = ((JSONObject) jRoutes.get(i)).getJSONArray("legs");
                List path = new ArrayList<>();

                for (int j = 0; j < jLegs.length(); j++) {

                    jSteps = ((JSONObject) jLegs.get(j)).getJSONArray("steps");

                    for (int k = 0; k < jSteps.length(); k++) {

                        String polyline = "";
                        polyline = (String) ((JSONObject) ((JSONObject) jSteps.get(k)).get("polyline")).get("points");
                        List<LatLng> list = decodePoly(polyline);

                        for (int l = 0; l < list.size(); l++) {

                            HashMap<String, String> hm = new HashMap<>();
                            hm.put("lat", Double.toString(list.get(l).latitude));
                            hm.put("lng", Double.toString(list.get(l).longitude));
                            path.add(hm);

                        }

                    }

                }

                routes.add(path);

            }

        } catch (JSONException e) {
            e.printStackTrace();
        } catch (Exception e) {
        }

        return routes;
    }
}

```

Şekil 4.56. Data Parser kodu- 1.

```

private List<LatLng> decodePoly(String encoded) {

    List<LatLng> poly = new ArrayList<>();
    int index = 0, len = encoded.length();
    int lat = 0, lng = 0;

    while (index < len) {
        int b, shift = 0, result = 0;

        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);

        int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lat += dlat;
        shift = 0;
        result = 0;
        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lng += dlng;

        LatLng p = new LatLng((((double) lat / 1E5)),
                               (((double) lng / 1E5)));
        poly.add(p);
    }
    return poly;
}
}

```

Şekil 4.57. Data Parser kodu- 2.

Oluşturulacak güzergâhın başlangıç ve bitiş noktalarının düzenlenmesi ve çizilen yolun ayarlarının yapılması için PointsParser.java adında Java dosyası oluşturularak dosya içerisine Şekil 4.58'deki kodlar eklenmelidir.

```

package com.galatakulesi.mes;
import android.content.Context;
import android.graphics.Color;
import android.os.AsyncTask;
import android.util.Log;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.PolylineOptions;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class PointsParser extends AsyncTask<String, Integer, List<List<HashMap<String, String>>>> {
    TaskLoadedCallback taskCallback;
    String directionMode = "driving";

    public PointsParser(Context mContext, String directionMode) {
        this.taskCallback = (TaskLoadedCallback) mContext;
        this.directionMode = directionMode;
    }

    @Override
    protected List<List<HashMap<String, String>>> doInBackground(String... jsonData) {

        JSONObject jObject;
        List<List<HashMap<String, String>>> routes = null;

        try {
            jObject = new JSONObject(jsonData[0]);
            DataParser parser = new DataParser();
            routes = parser.parse(jObject);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return routes;
    }

    @Override
    protected void onPostExecute(List<List<HashMap<String, String>>> result) {
        ArrayList<LatLng> points;
        PolylineOptions lineOptions = null;

        for (int i = 0; i < result.size(); i++) {
            points = new ArrayList<>();
            lineOptions = new PolylineOptions();
            List<HashMap<String, String>> path = result.get(i);

            for (int j = 0; j < path.size(); j++) {
                HashMap<String, String> point = path.get(j);
                double lat = Double.parseDouble(point.get("lat"));
                double lng = Double.parseDouble(point.get("lng"));
                LatLng position = new LatLng(lat, lng);
                points.add(position);
            }
            lineOptions.addAll(points);
            if (directionMode.equalsIgnoreCase("walking")) {
                lineOptions.width(10);
                lineOptions.color(Color.MAGENTA);
            } else {
                lineOptions.width(20);
                lineOptions.color(Color.BLUE);
            }
        }

        if (lineOptions != null) {
            taskCallback.onTaskDone(lineOptions);
        }
    }
}

```

Şekil 4.58. Points Parser kodu.

Güzergâh için hazırlanan verilerin servisler aracılığı ile gönderilerek gerekli verileri almak için FetchURL.java adında Java dosyası oluşturularak dosya içerisine Şekil 4.59'daki kodlar eklenmelidir.

```
package com.galatakulesi.mes;
import android.content.Context;
import android.os.AsyncTask;
import android.util.Log;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class FetchURL extends AsyncTask<String, Void, String> {
    Context mContext;
    String directionMode = "driving";

    public FetchURL(Context mContext) {
        this.mContext = mContext;
    }

    @Override
    protected String doInBackground(String... strings) {
        String data = "";
        directionMode = strings[1];
        try {
            data = downloadUrl(strings[0]);
        }
        return data;
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        PointsParser parserTask = new PointsParser(mContext, directionMode);
        parserTask.execute(s);
    }

    private String downloadUrl(String strUrl) throws IOException {
        String data = "";
        InputStream iStream = null;
        HttpURLConnection urlConnection = null;

        try {
            URL url = new URL(strUrl);
            urlConnection = (HttpURLConnection) url.openConnection();
            urlConnection.connect();
            iStream = urlConnection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(iStream));
            StringBuffer sb = new StringBuffer();
            String line = "";
            while ((line = br.readLine()) != null) {
                sb.append(line);
            }

            data = sb.toString();
            br.close();
        } catch (Exception e) {
            {
            } finally
            {
                iStream.close();
                urlConnection.disconnect();
            }
        }
        return data;
    }
}
```

Şekil 4.59. Fetch URL kodu.

Sistemi tamamlayan yapıyı oluşturmak için projede MapsActivity.java adında Java dosyası oluşturularak dosya içerisine Şekil 4.60, Şekil 4.61, Şekil 4.62, Şekil 4.63 ve Şekil 4.64'teki kodlar eklenmelidir. Eklenen kodlar ile Galata Kulesi'nin konumu ve kişinin konumu üzerinde işlemler yapılarak artırılmış gerçeklik uygulamasının tetiklenmesine imkân sağlanmaktadır.

```
package com.galatakulesi.mes;
import androidx.annotation.DrawableRes;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.drawable.Drawable;
import android.location.Location;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptor;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;

public class MapsActivity extends AppCompatActivity
    implements OnMapReadyCallback,
        GoogleApiClient.ConnectionCallbacks,
        GoogleApiClient.OnConnectionFailedListener,
        LocationListener, TaskLoadedCallback {
    private Polyline currentPolyline;
    GoogleMap mGoogleMap;
    SupportMapFragment mapFrag;
    LocationRequest mLocationRequest;
    GoogleApiClient mGoogleApiClient;
    Location mLastLocation;
    LatLng latLng, girilen_lokasyon;
    Marker mCurrLocationMarker;
    Context context = this;
    Location l1,l2;
    float aradaki_mesafe;
    Boolean girilene_göre_hesapla = false;
    String packageName = "com.MES.Galata";
    private MarkerOptions place1, place2,galata_konum,anlık_konum;
    Button yurume, arac;
    boolean yurume_arac = false;
```

Şekil 4.60. Maps Activity kodu- 1.

```

@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);
    yurume = findViewById(R.id.yurume);
    arac = findViewById(R.id.arac);

    yurume.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            yurume_arac = true;
            yurume.setBackgroundResource(R.drawable.yurume_aktif);
            arac.setBackgroundResource(R.drawable.arac_pasif);
            Toast.makeText(MapsActivity.this, "Yüreme rotası gösterilecek...", Toast.LENGTH_LONG).show();
        }
    });

    arac.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            yurume_arac = false;
            yurume.setBackgroundResource(R.drawable.yurume_pasif);
            arac.setBackgroundResource(R.drawable.arac_aktif);
            Toast.makeText(MapsActivity.this, "Araç rotası gösterilecek...", Toast.LENGTH_LONG).show();
        }
    });

    galata_konum = new MarkerOptions().position(new LatLng(41.0256541, 28.9741433)).title("Galata
Kulesi").icon(bitmapDescriptorFromVector(context, R.drawable.galata_tower));
    mapFrag = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
    mapFrag.getMapAsync(this);
}

@Override
public void onPause() {
    super.onPause();
    if (mGoogleApiClient != null) {
        LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,
(com.google.android.gms.location.LocationListener) this);
    }
}

@Override
public void onMapReady(GoogleMap googleMap)
{
    mGoogleMap=googleMap;
    mGoogleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    mGoogleMap.addMarker(galata_konum);

    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
== PackageManager.PERMISSION_GRANTED) {
            buildGoogleApiClient();
            mGoogleMap.setMyLocationEnabled(true);
        } else {
            checkLocationPermission();
        }
    }
    else {
        buildGoogleApiClient();
        mGoogleMap.setMyLocationEnabled(true);
    }
}
}

```

Şekil 4.61. Maps Activity kodu- 2.

```

protected synchronized void buildGoogleApiClient() {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
    mGoogleApiClient.connect();
}

@Override
public void onConnected(Bundle bundle) {

    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(5000);
    mLocationRequest.setFastestInterval(5000);
    mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) ==
    PackageManager.PERMISSION_GRANTED) {

        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest,
        (com.google.android.gms.location.LocationListener) this);
    }
}

@Override
public void onConnectionSuspended(int i) {}

@Override
public void onConnectionFailed(ConnectionResult connectionResult) {}

@Override
public void onLocationChanged(Location location)
{
    mLastLocation = location;
    if (mCurrLocationMarker != null) {
        mCurrLocationMarker.remove();
    }

    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions();
    markerOptions.position(latLng);

    markerOptions.title("Buradasınız!");
    markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN));
    mCurrLocationMarker = mGoogleMap.addMarker(markerOptions);

    l2 = new Location("");
    l2.setLatitude(mLastLocation.getLatitude());
    l2.setLongitude(mLastLocation.getLongitude());

    l1 = new Location("");
    l1.setLatitude(41.0256541);
    l1.setLongitude(28.9741433);

    anlık_konum = new MarkerOptions().position(new LatLng(mLastLocation.getLatitude(),
    mLastLocation.getLongitude())).title("Location 2");
    aradaki_mesafe = l1.distanceTo(mLastLocation);

    if (yurume_arac == false){
        new FetchURL(MapsActivity.this).execute(getUrl(galata_konum.getPosition(), anlık_konum.getPosition(), "driving"),
        "driving");
    }

    else if (yurume_arac == true){
        new FetchURL(MapsActivity.this).execute(getUrl(galata_konum.getPosition(), anlık_konum.getPosition(), "walking"),
        "walking");
    }
}

```

Şekil 4.62. Maps Activity kodu- 3.

```

if (aradaki_mesafe <= 500.0){
    Intent intent = getPackageManager().getLaunchIntentForPackage(packageName);

    if(intent == null) {
        try {
            intent = new Intent(Intent.ACTION_VIEW, Uri.parse("market://details?id=" + packageName));
        } catch (Exception e) {
            intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://play.google.com/store/apps/details?id=" +
packageName));
        }
        startActivity(intent);
    }
}

public static final int MY_PERMISSIONS_REQUEST_LOCATION = 99;
private void checkLocationPermission() {
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {

        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
            Manifest.permission.ACCESS_FINE_LOCATION)) {

            new AlertDialog.Builder(this)
                .setTitle("Konum bilginize erişim gerekiyor!")
                .setMessage("Bu uygulamanın fonksiyonlarını kullanabilmeniz için konum bilginize ihtiyaç duyulmaktadır.
")
                .setPositiveButton("İzin ver", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {
                        ActivityCompat.requestPermissions(MapsActivity.this,
                            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                            MY_PERMISSIONS_REQUEST_LOCATION );
                    }
                })
                .create()
                .show();
        } else {
            ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                MY_PERMISSIONS_REQUEST_LOCATION );
        }
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_LOCATION: {
            if (grantResults.length > 0
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

                if (ContextCompat.checkSelfPermission(this,
                    Manifest.permission.ACCESS_FINE_LOCATION)
                    == PackageManager.PERMISSION_GRANTED) {

                    if (mGoogleApiClient == null) {
                        buildGoogleApiClient();
                    }
                    mGoogleMap.setMyLocationEnabled(true);
                }
            } else {
                Toast.makeText(this, "İzin verilmedi...", Toast.LENGTH_LONG).show();
            }
            return;
        }
    }
}
}

```

Şekil 4.63. Maps Activity kodu- 4.

```

private BitmapDescriptor bitmapDescriptorFromVector(Context context, @DrawableRes int vectorDrawableResourceId) {
    Drawable background = ContextCompat.getDrawable(context, R.drawable.galata_tower);
    background.setBounds(0, 0, background.getIntrinsicWidth(), background.getIntrinsicHeight());
    Drawable vectorDrawable = ContextCompat.getDrawable(context, vectorDrawableResourceId);
    vectorDrawable.setBounds(40, 20, vectorDrawable.getIntrinsicWidth() - 40, vectorDrawable.getIntrinsicHeight() - 20);
    Bitmap bitmap = Bitmap.createBitmap(background.getIntrinsicWidth(), background.getIntrinsicHeight(),
    Bitmap.Config.ARGB_8888);
    Canvas canvas = new Canvas(bitmap);
    background.draw(canvas);
    vectorDrawable.draw(canvas);
    return BitmapDescriptorFactory.fromBitmap(bitmap);
}

private String getUrl(LatLng origin, LatLng dest, String directionMode) {
    String str_origin = "origin=" + origin.latitude + "," + origin.longitude;
    String str_dest = "destination=" + dest.latitude + "," + dest.longitude;
    String mode = "mode=" + directionMode;
    String parameters = str_origin + "&" + str_dest + "&" + mode;
    String output = "json";
    String url = "https://maps.googleapis.com/maps/api/directions/" + output + "?" + parameters + "&key=" +
    getString(R.string.google_maps_key);
    return url;
}

@Override
public void onTaskDone(Object... values) {
    if (currentPolyline != null)
        currentPolyline.remove();
    currentPolyline = mGoogleMap.addPolyline((PolylineOptions) values[0]);
}
}

```

Şekil 4.64. Maps Activity kodu- 5.

MapsActivity içinde hazırlanan sistemin ekran üzerinde deneyimleyebilmek için activity_maps.xml içerisine Şekil 4.65’teki kodlar eklenmelidir.

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MapsActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <fragment
            xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:map="http://schemas.android.com/apk/res-auto"
            xmlns:tools="http://schemas.android.com/tools"
            android:id="@+id/map"
            android:layout_weight="1"
            android:name="com.google.android.gms.maps.SupportMapFragment"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context=".MapsActivity" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:layout_marginTop="5dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/arac"
            android:layout_width="120dp"
            android:layout_height="36dp"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="15dp"
            android:background="@drawable/arac_aktif">
        </Button>

        <Button
            android:id="@+id/yurume"
            android:layout_width="120dp"
            android:layout_height="36dp"
            android:layout_marginRight="55dp"
            android:layout_marginTop="10dp"
            android:background="@drawable/yurume_pasif">
        </Button>
    </LinearLayout>
</FrameLayout>
</LinearLayout>

```

Şekil 4.65. Activity Maps kodu.

Hazırlanan projenin tamamlanması için console.developers.google.com üzerinden alınan harita anahtarını kullanarak google_maps_api.xml içerisine Şekil 4.66'daki kodlar eklenmelidir.

```

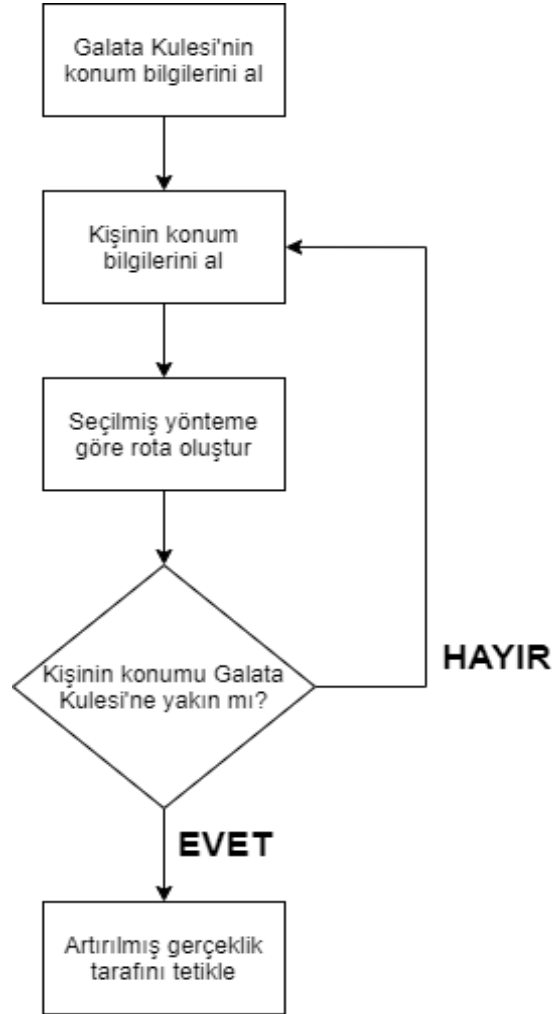
<resources>
    <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">KEY</string>
</resources>

```

Şekil 4.66. Google Maps Api kodu.

Build>Build APK(s) seçeneği ile hazırlanan yapının APK'sı elde edilir.

Hazırlanan uygulama incelendiğinde harita tarafının akış diyagramı Şekil 4.67 üzerinde ortaya konulmuştur.



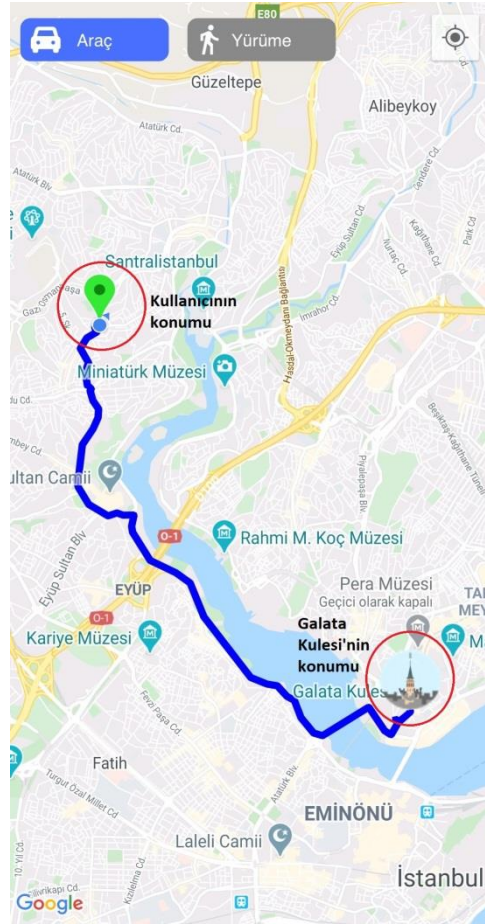
Şekil 4.67. Harita tarafının akış diyagramı.

4.9. HAZIRLANAN UYGULAMANIN ÇALIŞMA ŞEKLİ VE DENEYSEL ÇALIŞMALAR

Bu bölümde geliştirilen uygulama iki farklı şekilde test edilmiştir. İlk olarak Galata Kulesi'nin gerçek görüntüsü için test yapılmış ardından da Galata Kulesi biblosu kullanılarak test yapılmıştır.

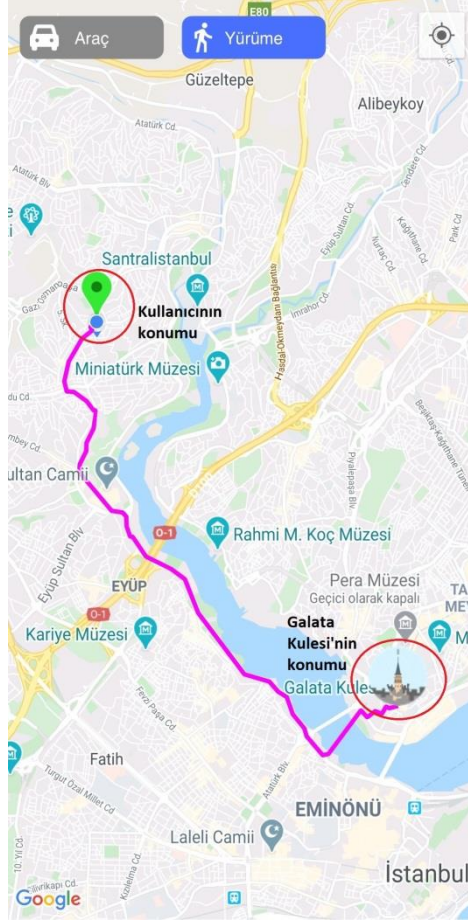
Gerçek kule görüntüsü üzerinde test işlemi için kullanıcının Galata Kulesi'nden uzakta olduğu varsayılarak başlanmıştır. Bunun nedeni uygulamaya eklenen CBS'nin etkisini göstermektir. Buna göre; Çalışma incelendiğinde başlangıç ekranı (Şekil 4.68) olarak

kişinin ve Galata Kulesi'nin konumlarını gösteren bir harita ekranı görülmektedir. Ekranın üst kısmında kişi ulaşımını sağlaması için araç ve yürüme olarak seçebileceği iki farklı yöntem bulunmaktadır. Uygulama başladığında varsayılan olarak araç için güzergâh gösterimi yapılmaktadır.



Şekil 4.68. Araç için güzergâh gösterimi.

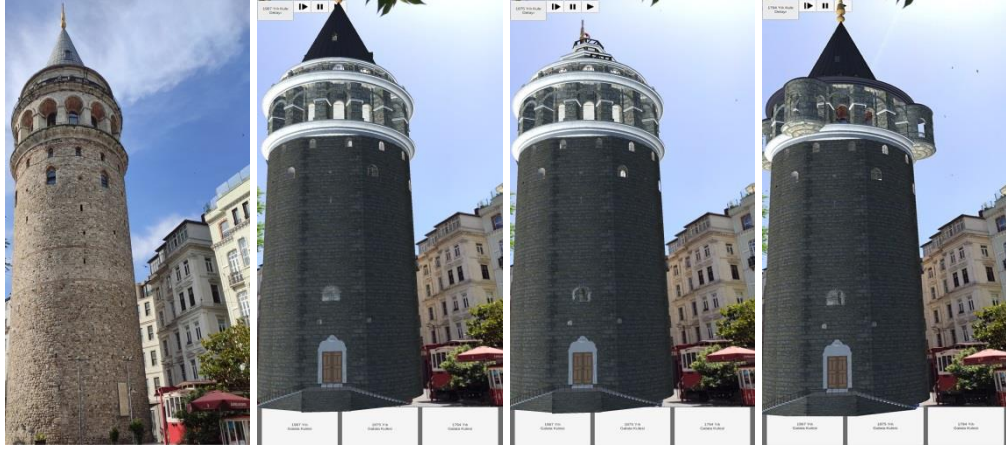
Uygulama içerisinde yürüme yönteminin aktifleştirilmesi ile çizilen rota yürümeye uygun olarak değişmektedir. Rota yürüme için güncellendiğinde araç için çizilen kalın mavi güzergâh yerine ince mor renkte (Şekil 4.69) bir güzergâh gösterimi yapılmaktadır.



Şekil 4.69. Yürüme için güzergâh gösterimi.

Kullanıcının konumu Galata Kulesi'ne yaklaştığında çalışmanın ikinci kısmı olan AG tarafı aktifleşmektedir. Test işlemleri sırasında hazırlanan uygulama içerisinde kullanılan tetikleyici için 360 derecelik çekim yerine 90 derecelik çekim ile Başlık 4.9'da anlatılan geliştirme adımları temel alınarak yapılmıştır. Şekil 4.8 incelendiğinde Galata Kulesi'ne yakın yapılan yapılar, yüksek ağaçlandırma ve kule boyunun 60 metre üstü olması nedeniyle uygulama kadrājına kulenin sığdırılabilmesi için en ideal kısım olan kulenin kuzey cephesi üzerinde uygulama test edilmiştir.

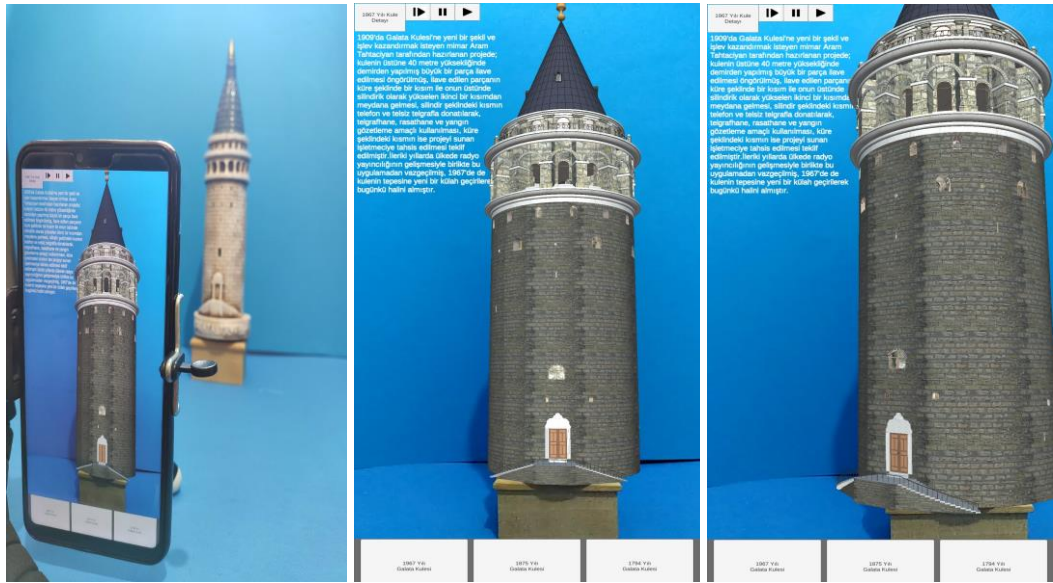
Uygulamanın Galata Kulesi üzerindeki test işlemleri sonucunda hazırlanmış olan sanal nesnelerin gösterimi Şekil 4.70'deki gibi gerçekleştirilmiştir.



a) b) c) d)

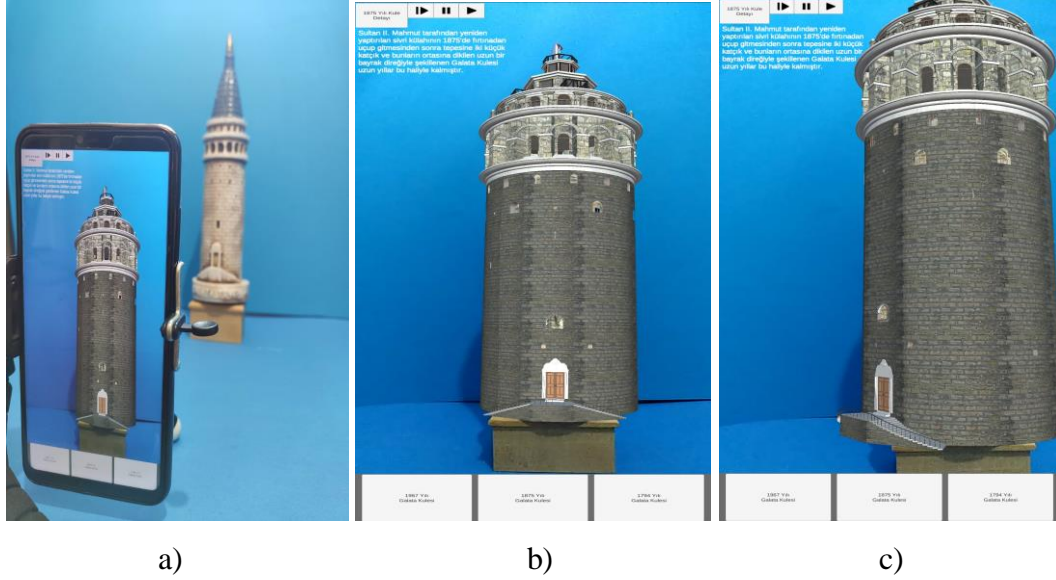
Şekil 4.70. a) Gerçek Galata Kulesi b) Galata Kulesi üzerinde 1967 yılındaki modelin gösterimi c) Galata Kulesi üzerinde 1875 yılındaki modelin gösterimi d) Galata Kulesi üzerinde 1794 yılındaki modelin gösterimi.

İlk test sonucunda uygulamanın gerçek kule görüntüsü ile tetiklenebildiği gösterilmiştir. İkinci test için ise daha önceden hazırlanan Galata Kulesi biblosu kullanılmıştır. Bu testte uygulamaya gösterilen biblo yardımı ile AG tetiklenmekte ve kullanıcıya Galata Kulesi ile ilgili bilgiler Şekil 4.71, Şekil 4.72 ve Şekil 4.73'teki gibi sunulmaktadır.

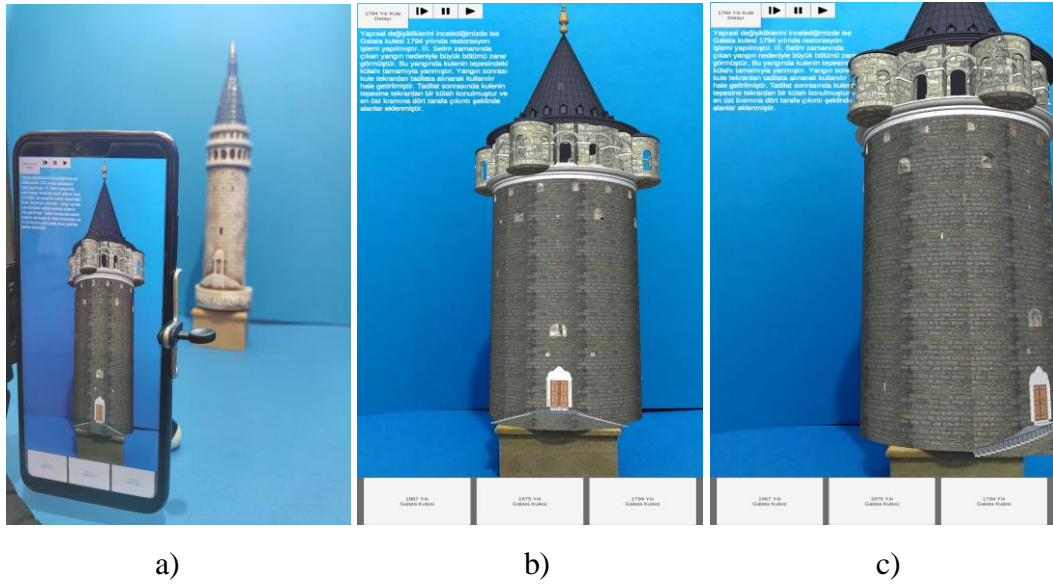


a) b) c)

Şekil 4.71. 1967 yılındaki Galata Kulesi'nin uygulamada a) uzaktan gösterimi b) ekran görüntüsü c) modelin döndürülüp büyütülmüş hali.



Şekil 4.72. 1875 yılındaki Galata Kulesi'nin uygulamada a) uzaktan gösterimi b) ekran görüntüsü c) modelin döndürülüp büyütülmüş hali.



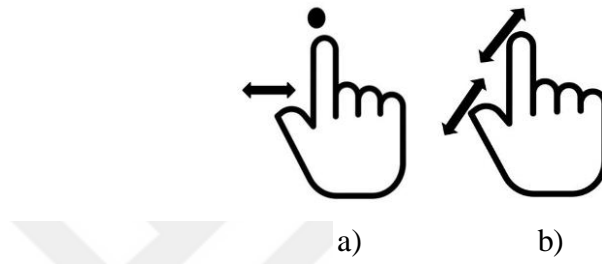
Şekil 4.73. 1794 yılındaki Galata Kulesi'nin uygulamada a) uzaktan gösterimi b) ekran görüntüsü c) modelin döndürülüp büyütülmüş hali.

Uygulama içeriği incelendiğinde alt kısmında bulunan butonlar Galata Kulesi'nin belirtilen yıllardaki restorasyon işlemi sonrası durumunu göstermektedir. Bu sayede kullanıcıya gerçek dünya üzerinde Galata Kulesi'nin tarihi akışını görüp deneyimleme imkânı verilmektedir.

Uygulamanın üst kısmında bulunan bilgilendirme butonu ise restorasyon işleminin neden yapıldığı hakkında bilgi vermektedir. Bu sayede tarihi akışı inceleyen kişi yapılan restorasyonların nedenlerini ve sonuçlarını net bir şekilde görebilecektir.

Uygulamada bulunan ses butonları ile kullanıcı modelleri incelerken bilgilendirme kısmında yer alan metni sesli olarak dinleyebilme imkânı sunulmaktadır. Her model için özel olarak hazırlanan ses kayıtları butonlar aracılığı ile tekrar dinleme, durdurma, başlatma gibi fonksiyonları barındırmaktadır.

Kullanıcı ekran üzerinde 2 parmağını kullanarak döndürme hareketi (Şekil 4.74.a) ile uygulamada bulunan modelleri kendi etrafında 360 derece döndürebilme ve kullanıcı yine 2 parmağını kullanarak büyütme-küçültme hareketi (Şekil 4.74.b) ile modeli büyütüp-küçülterek modelin detaylarını inceleyebilme imkânı kullanıcıya verilmiştir. Bu kapsamda uygulamanın işlevselliği artırılarak kullanım kolaylığı sağlanmıştır.



Şekil 4.74. Uygulamada modeli a) 360 derece döndürme hareketi b) büyütme-küçültme hareketi.

Hazırlanan çalışmanın gerçek dünya da kullanılması ile olası restorasyon işlemlerinin sonuçları doğru bir şekilde ortaya konulacaktır. İşlemler tamamlanmadan kullanıcı tarihi akıştaki halini net bir şekilde görebileceği için daha doğru sonuçlar elde edecektir.

Her iki testin sonuçları birlikte değerlendirildiğinde biblo için hazırlanan tetikleyicinin performansı gerçek Galata Kulesi için hazırlanan tetikleyicinin performansından daha iyi olduğu gözlemlenmiştir. Performans farkının nedeni ise biblo üzerinde yapılan çekimlerin profesyonel ortam ve ekipmanlar ile yapılması, Galata Kulesi'nin çekimlerinin ise kulenin önünde bulunan direkler, asılmış olan süsler, kule önünde bulunan portatif kafeler, kule önüne park eden araçlar, yoğun insan popülasyonu ve amatör kamera kullanımı gibi etkenler nedeniyle amatör olarak yapılabilmesidir.

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, artırılmış gerçeklik ve coğrafi bilgi sistemleri yaklaşımları kullanılarak tarihi eser restorasyonu için mobil bir uygulama geliştirilmiştir.

Çalışmada tarihi geçmişi ve bulunduğu konum nedeniyle İstanbul'un simgelerinden birisi olan Galata Kulesi model olarak seçilmiştir. Gerçekleştirilen birçok restorasyon işlemi ile milletlerin tarihlerini ve kültürlerini günümüze kadar taşıyabilmiştir.

Artırılmış gerçeklik teknolojisi hakkında yapılan çalışmaların ortaya koyduğu sonuçlar doğrultusunda, artırılmış gerçeklik teknolojisi insanın algısını artırmaktadır. Bu doğrultuda AG teknolojisi kullanılarak yapılan işler incelendiğinde alınan sonuçların doğruluk oranları artmaktadır.

Bu çalışma ile birlikte restorasyon işlemleri sırasında kullanıcının algısının artırılması ve en doğru sonuca ulaşılması hedeflenmiştir. Çalışma içerisinde yer alan farklı tarihlere ait restorasyon modelleri ile kullanıcı net bir şekilde Galata Kulesi'nin tarihi akışını gerçek dünya üzerinde deneyimleyebilecek ve bilgilendirme kısmı ile daha önceden yapılan çalışmaların sebep-sonuç ilişkilerini geniş bir açıdan görebilecektir.

Hazırlanan çalışmayla yapılacak restorasyon işlemlerinin gerçekleştirilmeden gerçek dünyada nasıl olacağını kullanıcıya sunabileceği için kullanıcı önceden ön göremediği hataları işlemlere başlamadan müdahale edebilme imkânı olacaktır. Bu çalışma ile birlikte restorasyon işlemlerinde daha doğru adımlar izleneceği ön görülmektedir.

Uygulamayı geliştirme sırasında, hazırlanan çalışma yapılacak restorasyon işlemlerinde bir yol haritası olarak düşünülmüş olmasına rağmen, uygulama yapının tarihi akışını net bir biçimde verebildiği için yerli ve yabancı turistlerinde oldukça dikkatini çekmiştir.

Hazırlanan çalışma Galata Kulesi üzerine yapılmış olmasına rağmen diğer tarihi eserler içinde uygulanması oldukça kolaydır. Hazırlanan çalışmanın restorasyon işlemlerinde yeni bir yaklaşımın oluşmasında katkı yapması beklenmektedir.

Yapılan çalışma, üretimleri sınırlı olan artırılmış gerçeklik gözlüklerine entegre edilmesiyle kullanıcıya daha kolay bir kullanım imkânı verecektir ve alınacak sonuçların doğruluk oranlarını arttıracığı düşünülmektedir.

6. KAYNAKLAR

- [1] G. Demirel, "Artırılmış gerçeklik uygulamaları ile işlenen fen bilimleri dersinin 7.sınıf öğrencilerinin akademik başarılarına ve artırılmış gerçeklik uygulamalarına karşı tutumlarına etkisi," Yüksek lisans tezi, Matematik ve Fen Bilimleri Eğitimi, Eğitim Bilimleri Enstitüsü, Gazi Üniversitesi, Ankara, Türkiye, 2019.
- [2] D. Perez-Lopez ve M. Contero, "Delivering educational multimedia contents through an augmented reality application: A case study on its impact on knowledge acquisition and retention," *The Turkish Online Journal of Educational Technology*, c. 12, sayı 4, ss. 19-28, 2013.
- [3] A. DiSerio, M. B. Ibanez ve C. D. Kloos, "Impact of an augmented reality system on students motivation for a visual art course," *Computers & Education*, c. 68, sayı 1, ss. 586-596, 2012.
- [4] P. Toledo-Morales ve J. M. Sanchez-Garcia, "Use of augmented reality in social sciences as educational resource," *Turkish Online Journal of Distance Education*, c. 19, sayı 3, ss. 38-42, 2018.
- [5] Türk Coğrafya Kurumu. (2019, 10 Eylül). "Türkiye'nin coğrafi bölgeleri" [Online]. Erişim: <http://www.tck.org.tr/makale/tr/fiziki-cografya/turkiyenin-cografibolgeleri>.
- [6] H. Biçer. (2019, 10 Eylül). "Anadolu'da kurulan medeniyetler (uygarlıklar)" [Online]. Erişim: <https://bilgihanem.com/anadoluda-kurulan-medeniyetler-uygarliklar>.
- [7] Kültür ve Turizm Bakanlığı. (2019, 10 Eylül). "Son 3 yılda Restorasyon Daire Başkanlığı görev ve sorumluluk alanında tamamlanan restorasyon çalışmaları" [Online]. Erişim: <http://www.kulturvarliklari.gov.tr/TR-180795/son-3-yilda-restorasyon-daire-baskanligi-gorev-ve-sorum-.html>.
- [8] A. Bursalı. (2019, 12 Eylül). "Yeni Hatay Arkeoloji Müzesi'nde roma mozaikleri restorasyon skandalı" [Online]. Erişim: <https://arkeofili.com/yeni-hatay-arkeoloji-muzesinde-roma-mozaikleri-restorasyon-skandalı>.
- [9] İstanbul İl Kültür ve Turizm Müdürlüğü. (2019, 15 Eylül). "Galata Kulesi" [Online]. Erişim: <https://istanbul.ktb.gov.tr/TR-165615/galata-kulesi.html>.
- [10] E. Artvinli, "Coğrafi Bilgi Sistemleri'nin (CBS) coğrafya öğretimine katkısı ve ortaöğretim öğrencilerinin CBS'ye ilişkin tutumları," *Kurmadan Uygulamaya Eğitim Bilimleri Dergisi*, c. 10, sayı 3, ss. 1255-1292, 2010.
- [11] W. Chen, "Historical Oslo on a handheld device - a mobile augmented reality application," 18th international conference on knowledge-based and intelligent information & engineering systems, Gdynia, Polonya, 2014, ss. 979-985.
- [12] I. Efstathiou, E. A. Kyza ve Y. Georgiou, "An inquiry-based augmented reality mobile learning approach to fostering primary school students' historical reasoning in non-formal settings", *Interactive Learning Environments*, c. 26, sayı 1, ss. 22-41, 2018.

- [13] C. H. Wang, N. H. Tsai, J. M. Lu ve M. J. J. Wang, "Usability evaluation of an instructional application based on Google Glass for mobile phone disassembly tasks," *Applied Ergonomics*, c. 77, ss. 58-69, 2019.
- [14] T. Oleksy ve A. Wnuk, "Augmented places: An impact of embodied historical experience on attitudes towards places," *Computers in Human Behavior*, c. 57, ss. 11-16, 2016.
- [15] Y. L. Chang, H. T. Hou, C. Y. Pan, Y. T. Sung ve K. E. Chang, "Apply an augmented reality in a mobile guidance to increase sense of place for heritage places," *Educational Technology & Society*, c. 18, sayı 2, ss. 166-178, 2015.
- [16] F. Girbacia, S. Butnariu, A. P. Orman ve C. C. Postelnicu, "Virtual restoration of deteriorated religious heritage objects using augmented reality technologies," *European Journal of Science and Theology*, c. 9, sayı 2, ss. 223-231, 2013.
- [17] L. Sabah ve M. Şimşek, "Artırılmış gerçeklik yöntemleri ile konumsal mobil kampüs bilgi sistemi," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 6, sayı 3, ss. 637-649, 2018.
- [18] V. Demirel ve Ç. Erbaş, "Mobil artırılmış gerçeklik uygulamalarının incelenmesi ve eğitimsel açıdan değerlendirilmesi," *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, c. 11, sayı 3, ss. 802-813, 2015.
- [19] S. Kıvrak ve G. Arslan, "İnşaat proje imalatlarında artırılmış gerçeklik teknolojisi uygulamaları," *Politeknik Dergisi*, c. 21, sayı 2, ss. 379-385, 2017.
- [20] M. Polat, "Coğrafi bilgi sistemleri yaklaşımı ile tasarlanan turistik amaçlı artırılmış gerçeklik uygulaması: Safranbolu örneği," Yüksek lisans tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Karabük Üniversitesi, Karabük, Türkiye, 2017.
- [21] M. Albayrak ve V. Altıntaş, "Artırılmış gerçeklik teknolojisinin veritabanı dersinde kullanımı," *İstanbul Eğitimde Yenilikçilik Dergisi*, c. 3, sayı 1, ss. 13-23, 2017.
- [22] V. Vlahakis, N. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne ve L. Alameda, "Archeoguide: An augmented reality guide for archaeological sites," *IEEE Computer Graphics and Applications*, c. 22, sayı 5, ss. 52-59, 2002.
- [23] A. C. Haugstvedt ve J. Krogstie, "Mobile augmented reality for cultural heritage: a technology acceptance study," *IEEE international symposium on mixed and augmented reality*, Atlanta, Amerika Birleşik Devletleri, 2012, ss. 247-255.
- [24] G. A. Lee, A. Duenser, S. Kim ve M. Billingham, "CityViewAR: A mobile outdoor AR application for city visualization," *IEEE international symposium on mixed and augmented reality*, Atlanta, Amerika Birleşik Devletleri, 2012, ss. 57-64.
- [25] G. Papagiannakis, S. Schertenleib, B. O'Kennedy, M. Arevalo-Poizat, N. Magnenat-Thalmann, A. Stoddart ve D. Thalmann, "Mixing virtual and real scenes in the site of ancient Pompeii," *Computer Animation and Virtual Worlds*, c. 16, sayı 1, ss. 11-24, 2005.
- [26] A. F. Abate, S. Barra, G. Galeotafiore, C. Diaz, E. Aura, M. Sanchez, X. Mas ve E. Vendrell, "An augmented reality mobile app for museums: virtual restoration of a

- plate of glass,” 7th international conference, Lefkoşa, Kıbrıs, 2018, ss. 539-547.
- [27] V. C. Ak, “Artırılmış gerçeklik ortamı kullanılarak bir uygulama geliştirilmesi,” Yüksek lisans tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Yaşar Üniversitesi, İzmir, Türkiye, 2018.
- [28] G. Sucaklı, “Turizmde modern müzecilik anlayışı ve artırılmış gerçeklik uygulamalarının müzelerde kullanımı,” Yüksek lisans tezi, Turizm İşletmeciliği, Sosyal Bilimler Enstitüsü, Çanakkale Onsekiz Mart Üniversitesi, Çanakkale, Türkiye, 2019.
- [29] O. Balas, “Uzamsal artırılmış gerçeklik tekniklerinin sergileme tasarımında kullanımı,” Yüksek lisans tezi, Grafik, Güzel Sanatlar Enstitüsü, Hacettepe Üniversitesi, Ankara, Türkiye, 2019.
- [30] İ. Kazan, “Turizmde artırılmış gerçeklik kullanımı,” Yüksek lisans tezi, Seyahat İşletmeciliği ve Turist Rehberliği, Sosyal Bilimler Enstitüsü, Selçuk Üniversitesi, Konya, Türkiye, 2019.
- [31] E. Durand, F. Merienne, C. Pere ve P. Callet, “Ray-on, an on-site photometric augmented reality device,” *Journal on Computing and Cultural Heritage*, c. 7, sayı 2, ss. 1-13, 2014.
- [32] F. C. Ünal, “Mimari tasarımda artırılmış gerçeklik aracılığıyla yerel veri kullanımı,” Doktora tezi, Bilişim, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2019.
- [33] G. Serbest, “Kültürel mirasın sanal ve artırılmış gerçeklikle temsili üzerine bir çalışma,” Yüksek lisans tezi, Elektronik Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Hasan Kalyoncu Üniversitesi, Gaziantep, Türkiye, 2019.
- [34] F. Demircioğlu, “Altyapı bilgi sistemlerinde artırılmış gerçeklik ve coğrafi bilgi teknolojileri entegrasyonu,” Yüksek lisans tezi, Uzaktan Algılama ve Coğrafi Bilgi Sistemleri, Lisansüstü Eğitim Enstitüsü, Eskişehir Teknik Üniversitesi, Eskişehir, Türkiye, 2019.
- [35] A. Doğan, “Artırılmış gerçeklik teknolojileriyle desteklenmiş hikaye kitabı okuma deneyimi,” *Medeniyet Sanat Dergisi*, c. 2, sayı 2, ss. 121-137, 2016.
- [36] R.E. Poelman ve D.W.F. Krevelen, “A survey of augmented reality technologies, applications and limitations,” *The International Journal of Virtual Reality*, c. 9, sayı 2, ss. 1-20, 2010.
- [37] Ü. Tunalı, “Bilgisayar mühendisliği eğitiminde artırılmış gerçeklik uygulaması,” Yüksek lisans tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Selçuk Üniversitesi, Konya, Türkiye, 2015.
- [38] S. Somyürek, “Öğrenme sürecinde z kuşağının dikkatini çekme: Artırılmış gerçeklik,” *Eğitim Teknolojisi Kuram ve Uygulama Dergisi*, c. 4, sayı 1, ss. 63-80, 2014.
- [39] T. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier ve B. Macintyre, “Recent advances in augmented reality,” *IEEE Computer Graphics and Applications*, c. 21, sayı 1, ss. 34-47, 2001.
- [40] BBC. (2019, 18 Eylül). “Google glass to go on sale for one day” [Online]. Erişim: <https://www.bbc.com/news/technology-26987972>.

- [41] F.C. Ünal, "Artırılmış gerçeklik teknolojisinin kullanımıyla mimarlık rehberi; Eindhoven kenti üzerinden değerlendirilmesi," Yüksek lisans tezi, Bilişim, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2013.
- [42] M. F. Akbaş ve C. Güngör, "Artırılmış gerçeklikte işaretçi tabanlı takip sistemleri üzerine bir literatür çalışması ve tasarlanan çok katmanlı işaretçi modeli," *Dokuz Eylül Üniversitesi-Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, c. 19, sayı 56, ss. 559-619, 2017.
- [43] E. İbili, "Geometri dersi için artırılmış gerçeklik materyallerinin geliştirilmesi, uygulanması ve etkisinin değerlendirilmesi," Doktora tezi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Eğitim Bilimleri Enstitüsü, Gazi Üniversitesi, Ankara, 2013.
- [44] T. İçten ve G. Bal, "Artırılmış gerçeklik teknolojisi üzerine yapılan akademik çalışmaların içerik analizi," *Bilişim Teknolojileri Dergisi*, c. 10, sayı 4, ss. 401-415, 2017.
- [45] P. Theodorou, P. Kydonakis, M. Botzori ve C. Skanavis, "Augmented reality proves to be a breakthrough in environmental education," Protection and restoration of the environment XIV'de sunuldu, Yunanistan, 2018.
- [46] A. Todoran. (2019, 3 Ekim). "Augmented reality in android apps using ARCore" [Online]. Erişim: <https://medium.com/corebuild-software/augmented-reality-in-android-apps-using-arcore-c6fba0897ce1>.
- [47] H. Altınpulluk, "Artırılmış gerçekliği anlamak: Kavramlar ve uygulamalar," *Açıköğretim Uygulamaları ve Araştırmaları Dergisi*, c. 1, sayı 4, ss. 123-131, 2015.
- [48] E. K. Gür, "Artırılmış gerçeklik teknolojisinin mimari tasarım uygulama sürecinde değerlendirilmesi üzerine bir model," Yüksek lisans tezi, Bilişim, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2014.
- [49] T. Yöndem ve G. H. Karadağ, "Artırılmış gerçeklikle değişen haber sunumu," *Yeni Medya Elektronik Dergisi*, c. 3, sayı 1, ss. 22-44, 2018.
- [50] J. Corso ve G. Hager, "Planar surface tracking using direct stereo," The Johns Hopkins University - Computational Interaction and Robotics Laboratory, Amerika Birleşik Devletleri, 2002.
- [51]. R. Azuma, "Predictive tracking for augmented reality," Kuzey Karolina Üniversitesi, Amerika Birleşik Devletleri, TR95-007, 1995.
- [52] Vuzix. (2019, 5 Ekim). "Introducing the augmented reality smart glasses" [Online]. Erişim: <https://www.vuzix.eu/augmentedreality>.
- [53] B. A. Parviz. (2020, 21 Ocak). "Augmented reality in a contact lens" [Online]. Erişim: <https://spectrum.ieee.org/biomedical/bionics/augmented-reality-in-a-contact-lens>.
- [54] Engineers Garage. (2019, 9 Ekim). Sixth sense technology. https://www.engineersgarage.com/article_page/sixth-sense-technology.
- [55] U. Albrecht, K. Folta-Schoofs ve U. Von Jan, "Effects of mobile augmented reality learning compared to textbook learning on medical students: Randomized controlled pilot study," *Journal of Medical Internet Research*, c. 15, sayı 8, böl 182, 2013.

- [56] M. R. Marnier, R. T. Smith, J. A. Walsh ve B. H. Thomas, "Spatial user interfaces for large-scale projector-based augmented reality," *IEEE Computer Graphics and Applications*, c. 34, sayı 1, ss. 74-82, 2014.
- [57] İ. Yiğit, M. Ataoğlu ve A. Dinç, "Coğrafya bölümlerindeki CBS eğitimi ve CBS'nin gerekliliği," *Marmara Coğrafya Dergisi*, c. 1, sayı 24, ss. 312-331, 2011.
- [58] Maden Tetkik Ve Arama Genel Müdürlüğü. (2019, 9 Ekim). "Coğrafi bilgi sistemleri" [Online]. Erişim: <https://www.mta.gov.tr/v3.0/arastirmalar/cografibilgi-sistemleri>.
- [59] Aselsan. (2019, 20 Ekim). "Coğrafi bilgi sistemleri" [Online]. Erişim: <https://www.aselsan.com.tr/tr/cozumlerimiz/bilgi-teknolojileri-sistemleri/edevlet/cografibilgi-sistemleri>.
- [60] E. Kapluhan, "Coğrafi Bilgi Sistemleri'nin (CBS) coğrafya öğretiminde kullanımının önemi ve gerekliliği," *Marmara Coğrafya Dergisi*, c. 1, sayı 29, ss. 34-59, 2014.
- [61] M. Çakır, "Yerel yönetimlerde coğrafi bilginin kullanımı," Yüksek lisans tezi, Coğrafya, Sosyal Bilimler Enstitüsü, İstanbul Üniversitesi, İstanbul, Türkiye, 2019.
- [62] M. Gümrükçüoğlu, "Coğrafi bilgi sistemleri: Anlamı, yararları, sorunları ve geleceği," *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, c. 7, sayı 1, ss. 67-72, 2003.
- [63] İstanbul Büyük Şehir Belediyesi CBS Akademi. (2019, 20 Ekim). "CBS yazılımları geliştirme amaçları ve kullanım alanları" [Online]. Erişim: <https://cbsakademi.ibb.istanbul/cbs-yazilimlari-gelistirme-amaclari-ve-kullanim-alanlari>.
- [64] S. Saraç, "Android işletim sistemi üzerinde çalışan bir harita paket programının geliştirilmesi," Yüksek lisans tezi, Geomatik Mühendisliği, Fen Bilimleri Enstitüsü, Zonguldak Bülent Ecevit Üniversitesi, Zonguldak, Türkiye, 2019.
- [65] G. Mercan, "Android grafiksel kullanıcı arayüzleri için model tabanlı test senaryosu üretimi," Yüksek lisans tezi, Uluslararası Bilgisayar, Fen Bilimleri Enstitüsü, Ege Üniversitesi, İzmir, Türkiye, 2019.
- [66] A. Işingör. (2019, 21 Ekim). "Android studio'yu tanıyalım" [Online]. Erişim: <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/android-studioyu-taniyalim>.
- [67] S. Südor, "Üç boyutlu modelleme bilgisinin unity programı öğrenimine katkısı," *Ankara Üniversitesi Sosyal Bilimler Dergisi*, c. 10, sayı 2, ss. 126-134, 2019.
- [68] F. Sarı, A. Erdi ve O. Kırtıoğlu, "İnternet tabanlı coğrafi bilgi sistemi uygulamalarında geoserver, arcgis server, google maps api ve openlayers entegrasyon," *Jeoinformasyon ve Arazi Yönetimi Dergisi*, c. 1, sayı 1, ss. 140-144, 2011.
- [69] M. Akbulut ve B. Çare, "Beytepe kampüsü haritası mashup uygulaması," *Bilgi Dünyası Dergisi*, c. 12, sayı 2, ss. 334-346, 2011.
- [70] Sabah . (2019, 2 Kasım). "İstanbul'un tarihi gözdeleleri" [Online]. Erişim: <https://www.sabah.com.tr/galeri/kultursanat/istanbulun-tarihi-gozdeleri-1416581838>.

- [71] Akşam. (2019, 2 Kasım). “Galata Kulesi'nin bilinmeyen kardeşleri” [Online]. Erişim: <https://www.aksam.com.tr/foto-galeri/kultursanat/galata-kulesinin-bilinmeyen-kardesleri/34054>.
- [72] M. Kara, “Matrakçı Nasuh'un İstanbul minyatürü,” Lisans tezi, Sanat Tarihi, Sosyal Bilimler Enstitüsü, Anadolu Üniversitesi, Eskişehir, Türkiye, 2016.
- [73] İslam Ansiklopedisi. (2019, 5 Kasım). Galata Kulesi. <https://islamansiklopedisi.org.tr/galata-kulesi>.
- [74] D. Aral. (2019, 5 Kasım). “Galata Kulesi tarihi” [Online]. Erişim: <https://www.restorasyonforum.com/sanat-tarihi/galata-kulesi-tarihi>.
- [75] R. E. Koçu, “Galata Kulesi,” *İstanbul Ansiklopedisi*, İstanbul, Türkiye: Koçu Yayınları, 1972, c. 11, sayı 165, ss. 5901-5910.
- [76] R. Salik. (2019, 2 Aralık). “Galata Kulesi'nin gizemli hikayesi: Bu kule neden burada?” [Online]. Erişim: <http://www.milliyet.com.tr/galata-kulesi-nin-gizemli-hikayesi--bu-kule-neden-burada--molatik-8798>.
- [77] Beltur. (2019, 2 Aralık). “Galata Kulesi” [Online]. Erişim: <https://beltur.istanbul/tr/mekan/galata-kulesi>.
- [78] E. Yiğiter, “Sivas'taki Anadolu Selçuklu medreselerinde restorasyon çalışmaları (19. - 21. yüzyıl),” Yüksek lisans tezi, Sanat Tarihi, Sosyal Bilimler Enstitüsü, Selçuk Üniversitesi, Konya, Türkiye, 2018.
- [79] E. Önal, “Diyarbakır ili, Çermik ilçesi kültür varlıklarını koruma önerileri,” Yüksek lisans tezi, Mimarlık, Fen Bilimleri Enstitüsü, Maltepe Üniversitesi, İstanbul, Türkiye, 2018.
- [80] Vuforia. (2020, 25 Mayıs). “Optimizing target detection and tracking stability” [Online]. Erişim: <https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability.html>.
- [81] C. H. Tsai ve J. C. Yen, “The augmented reality application of multimedia technology in aquatic organisms instruction,” *Journal of Software Engineering and Applications*, c. 7, sayı 1, ss. 745-755, 2014.
- [82] S. Siltanen, “Theory and applications of marker-based augmented reality,” VTT Science 3, Espoo, Finlandiya. 2012.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Murat Eren Sarı
Doğum Tarihi ve Yeri : 26.06.1995 Eminönü, İstanbul
Yabancı Dili : İngilizce
E-posta : muraterensari@hotmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2020
Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2017
Lise	Bilişim Teknolojileri	İnönü Teknik Meslek Lisesi	2013

YAYINLAR

Sarı M. E., Toz M, Artırılmış Gerçeklik ve Coğrafi Bilgi Sistemleri Yaklaşımları Kullanılarak Tarihi Eser Restorasyonu İçin Mobil Uygulama Geliştirme (BILMES 2019)

Kalkışım A. E., Sarı M. E., Yücedağ İ., Application Of Augmented Reality Technology In Civil Engineering And Architecture Areas (ICEARC 2019)