



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



SANAL VE ARTIRILMIŞ GERÇEKLİK TEKNOLOJİSİYLE TÜMLEŞİK KUANTUM BİLGİSAYAR DERLEYİCİSİ KULLANIM PLATFORMUNUN GERÇEKLEŞTİRİLMESİ

HASAN HAKAN GENÇ

DOKTORA TEZİ

Elektrik ve Elektronik
Mühendisliği Ana Bilim Dalı
Elektrik ve Elektronik
Mühendisliği Programı

DANIŞMAN

Prof. Dr. Hasan ERDAL

EŞ-DANIŞMAN

Dr. Öğr. Üyesi Serkan AYDIN

İSTANBUL, 2022

TEŞEKKÜR

2014 yılında başladığım Doktora eğitimimi yatay geçiş ile Marmara Üniversitesi'ne aldığım 2017 yılında, öğrencisi olarak beni kabul eden, bu süreçte böyle bir çalışmanın yürütülmesi noktasında şahsıma duyduğu güven ve sağladığı destek için Marmara Üniversitesi Teknoloji Fakültesi Dekanı sayın Prof.Dr. Hasan ERDAL hocama teşekkürlerimi sunarım.

Güncel gelişmeleri çok iyi takip eden, çok yüksek analitik düşünme ve tahmin gücüne sahip olan, bu çalışmanın temellerinin atılmasını sağlayan ve çalışmanın eş danışmanlığını yürüten, kıymetli dostum Dr.Öğr. Üyesi Serkan AYDIN'a en derin sevgilerle teşekkür ederim.

Çalışma sırasında ilgi ve desteğini esirgemeyen, akademik çalışma hayatım süresince hep yanımda olan sevgili eşim Hatice'ye, çalışmalarım sırasında bana en ufak bir rahatsızlık yaşatmayan biricik kızlarım Deniz Zülal ve Şevval'e, dualarıyla her zaman yanımda olan annem Süheyla GENÇ'e ve babam İrfan GENÇ'e çok teşekkür ederim.

Yatay geçiş ile birlikte yeniden ders almak durumunda kaldığım süreçte vermiş oldukları derslerle yardımlarını ve eğitim hayatım süresince olumlu katkılarını gördüğüm kıymetli hocalarım sayın Prof.Dr. Hayriye KORKMAZ'a ve Dr.Öğrt. Üyesi Ulvi BAŞPINAR'a, tez izleme çalışmalarımı takip eden ve hazırladığım raporları okuyan, kıymetli fikirlerini benimle paylaşan sayın Doç.Dr. Önder DEMİR'e ve İstanbul Ticaret Üniversitesinden Doç.Dr. Cem KASAPBAŞI'na, İstanbul'da bulunduğum süre zarfında her zaman kapısını çaldığım ve evini bana açan kıymetli dostum Dr.Öğrt. Üyesi Barış DOĞAN'a ayrı ayrı en derin saygılarımı ve teşekkürlerimi sunarım.

Çalışmamda kullandığım donanımların tedarikini sağlayan ve hazırladığımız projeyi kabul eden Marmara Üniversitesi Bilimsel Araştırma Projeleri Birimine de teşekkürlerimi bir borç bilirim.

Mart 2022

H.Hakan GENÇ

İÇİNDEKİLER

SAYFA

TEŞEKKÜR	i
İÇİNDEKİLER.....	ii
ÖZET	iv
ABSTRACT	v
YENİLİK BEYANI.....	vi
SEMBOLLER	vii
KISALTMALAR	viii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	xiii
1. GİRİŞ	1
1.1. Araştırmanın Amacı.....	4
1.2. Literatür Araştırması.....	5
1.2.1. İnsan Bilgisayar Etkileşimi (HCI)	6
1.2.1.1 Etkileşim Uzayının Tasarımı.....	6
1.2.1.2 Etkileşim Tasarımında Dikkat Edilmesi Gereken Diğer Noktalar.....	8
1.2.2. Sanal Gerçeklik ile İlgili Genel Bilgiler	9
1.2.3. VR/AR Tasarımında Kullanılan Platformlar	15
1.2.3.1. Unity Programının Kurulumu	16
1.2.3.2. Unity Programı Hakkında	18
1.2.4. Kuantum Bilişimi ile İlgili Genel Bilgiler	19
1.2.4.1. Kuantum Bilgisayarı	20
1.2.4.2. Kübit Kavramı ve Bloch Küresi ile Temsili.....	23
1.2.4.3. Kuantum Kapıları	25
1.2.4.4. Kuantum Bilişimi ve Kuantum Kapılarında İşlemler	28
1.2.4.5. Kuantum Bulut Programlama Platformları	34
1.2.4.6. IBM Kuantum Besteleyicisi	37
2. MATERYAL ve YÖNTEM	43

2.1. Tasarım Konseptinin Planlanması	44
2.2. Eğitim İçeriğinin Belirlenmesi	47
2.2.1. Temel Kapılar	48
2.2.2. Ölçme İşlemleri	49
2.2.3. Çok Kübitli Kapılar	50
2.2.4. Süperpozisyon.....	51
2.2.5. Grover Algoritması.....	53
2.2.6. Işınlanma (Teleportation)	56
3. VR KUANTUM PROGRAMLAMA PLATFORMU	63
3.1. Sahne Yönetimi Tasarımı	63
3.1.1. Sahnedeki Nesnelerin Kontrolü.....	67
3.1.2. Sahnedeki Nesnelerde Fizik Yasaları	70
3.1.3. Dinamik Değişken Bilgi Mesajlarının Tasarımı.....	72
3.2. Temel Kuantum Bilişimi Uygulamalarının Tasarımı	77
3.2.1. Temel Kuantum Bilişimi Uygulamalarında Görsel Tasarım.....	77
3.2.2. Temel Kuantum Bilişimi Uygulamalarında Matematik	87
3.3. QASM Dosyasının Üretilmesi	95
4. SONUÇ VE TARTIŞMALAR.....	99
5. ÖNERİLER	102
KAYNAKLAR.....	103
EKLER	110
EK 1: Unity Programının Kurulumu Hakkında.....	111
EK 2: Oculus Rift ve XR Management Araçlarının Unity 3D'ye Kurulumu	115

ÖZET

SANAL VE ARTIRILMIŞ GERÇEKLIK TEKNOLOJİSİYLE TÜMLEŞİK KUANTUM BİLGİSAYAR DERLEYİCİSİ KULLANIM PLATFORMUNUN GERÇEKLEŞTİRİLMESİ

Süperpozisyon durumlarının, tutarlılık süresinin, aslına uygunluk, hata düzeltme, güç tüketimi ve gürültü açısından mükemmelleşmesiyle her geçen gün daha güçlü ve pratik hale gelen kuantum bilgisayarlarının, artan bir hızla mevcut sistemlere göre pek çok alanda daha yüksek bir problem çözme performansı gösterecekleri öngörülmektedir. Bazı yüksek teknoloji şirketlerinin araştırma ve geliştirme aşamasından problem çözme aşamasına kadar aktif olarak kullanıma sundukları kuantum bilgisayar tasarımları vardır ve bu bilgisayarlarda farklı mimariler kullanılmaktadır.

Çalışmaya kaynaklık eden “IBM Quantum Experience” laboratuvarı, sahip olduğu bulut tabanlı kuantum bilişimi platformu ile herkese açık erişimli ve çok alternatifli bir seçenek sunmaktadır. Bu platformda, programlama kuantum besteleyicisi adı verilen bir web arayüzü aracılığıyla kuantum kapıları kullanılarak yapılabilmektedir.

Sanal gerçeklik (Virtual Reality – VR) ve artırılmış gerçeklik (Augmented Reality – AR) teknolojilerinde de önemli ilerlemeler kaydedilmiş ve sanal gerçeklik tabanlı tarayıcı platformları geliştirilmiştir. Bu çalışmada, IBM kuantum besteleyicisinden ve Microsoft’un Q# eğitimi materyallerinden esinlenerek tasarlanmış kuantum kapılarının matematik modelleri yazılım yordamıyla sanal gerçeklik ortamına aktarılmış ve daha sonra oluşturulan kodların bulut bilişim sisteminde derlenebilmesi için gerekli olan düzenlemeler gerçekleştirilerek sanal gerçeklik temelli kuantum bilişimi programlama platformunun tasarım süreci tez kapsamında sunulmuştur.

Çalışma temel olarak, insan-bilgisayar arayüzü etkileşimi çalışmalarının bir örneği olan sanal gerçeklik yaklaşımı ile kuantum bilişimi ilkelerinin birleştirilmesini içermektedir. Tasarım ortamında, hem üç boyutlu bir programlama platformu hem de sanal gerçeklik temelli olarak gerçek zamanlı çalışan bir web tarayıcısı yer almaktadır. Kullanıcılar sanal gerçeklik ortamında prosedürlere uygun şekilde işlemleri gerçekleştirmekte ve VR temelli web tarayıcısı yardımıyla kuantum bilgisayarları üzerinden derleme işlemi yapabilmektedir.

ABSTRACT

REALIZATION OF INTEGRATED QUANTUM COMPUTER COMPILER USER PLATFORM WITH VIRTUAL AND AUGMENTED REALITY TECHNOLOGY

It is predicted that quantum computers, which become more powerful and practical with each passing day, with the perfection of superposition states, coherence time and fidelity, error correction, power consumption and noise, will show a higher problem solving performance in many areas than existing systems with increasing speed. Some high-tech companies have quantum computer designs that they actively use from the research and development stage to the problem solving stage, and different architectures are used in these computers.

The "IBM Quantum Experience" laboratory, which is the source of the study, offers an open access and multi-alternative option with its cloud-based quantum computing platform. In this platform, programming can be done using quantum gates via a web interface called quantum composer.

Significant advances have been made in virtual reality and augmented reality technologies and virtual reality-based browser platforms have been developed. In this study, the mathematical models of quantum gates, inspired by IBM quantum composer and Microsoft's Q# training materials, were transferred to the virtual reality environment with a software procedure, then the necessary arrangements were made for compiling the generated codes in the cloud computing system, and the design of the virtual reality-based quantum computing programming platform was carried out.

The study basically consists of combining the principles of quantum computing and the virtual reality approach, which is an example of human computer interaction studies. In the design environment, there is both a comprehensive three-dimensional programming platform and a real-time web browser based on virtual reality. Users can perform operations in accordance with the procedures in the virtual reality environment and can compile over quantum computers via a VR-based web browser.

YENİLİK BEYANI

Çalışmanın sunduğu en önemli yenilik, iki farklı disiplinin birleştirilerek yeni nesil kuantum bilişimi programlama ilkelerinin sanal gerçeklik (Virtual Reality - VR) ortamı üzerinden eğitiminin sunulmasıdır. Çalışmanın yürütüldüğü dönem içinde yazın alanda yapılan incelemeler sonucu, bu ilkesel yaklaşımda bir çalışmanın başlatıldığı ve yürütüldüğü görülmemiştir. Bu nedenle tez çalışmasının tüm fikir unsurları ve bu unsurların sanal gerçeklik ortamında senaryolar dahilinde çalıştırılabilmesi ve sonrasında gerçek kuantum bilgisayarlarına derlenebilecek şekilde kodların sunulması yenilik bakımından ülkemizde ve dünyada yegâne özellik taşımaktadır.

Kullanılan bütün kuantum kapı fonksiyonlarının matematik modellerinin sanal gerçeklik platformunun kabul göreceği yazılımla koşturulması, çıktılarının görselleştirilmesi ve bulut bilişim sisteminde koşturulması için gerekli olan dönüşümlerin yapılması bu tezin bir ürünüdür.

Yalnız başına (standalone) olarak karşımıza çıkan VR web tarayıcılarının yerine, bağımsız olarak ve her VR ortamında kullanılabilen açık kaynak kodlu bir web tarayıcısı amaca uygun olarak revize edilmiştir. Bu web tarayıcısı VR eğitim platformu içinde IBM kuantum derleyicilerine bağlantı kurabilmek için kullanılmıştır. Kuantum besteleycisine benzer bir mantıkla tasarlanmış ortamda gerçekleştirilen işlemler, doğruluk durumlarına göre, VR web tarayıcısı üzerinden ilgili bağlantıya yönlendirilerek derleme işlemi gerçek zamanlı olarak ve gerçek kuantum bilgisayarları üzerinden yürütülmektedir.

Tez kapsamında kodlama işlemlerinin kolaylıkla yapılabilmesi, dezavantajlı bireyler dahil tüm bireylerin rahatlıkla kuantum bilgisayar programlama işlemlerini öğrenebilmesi için kendini işbaşında hissettiren uygun sanal gerçeklik platformu yeni bir programlama ortamı olarak gerçekleştirilmiştir.

Bu tez çalışması ülkemizde olmayan kuantum bilişim sistemlerine erişim sağlayarak hem kuantum bilgisayarların gerçek zamanlı programlanabilmesi için çevrimiçi bağlantı sağlamakta, hem de çevrimdışı kuantum kapılarının öğretilmesi için uygun senaryoları barındırmaktadır.

SEMBOLLER

Hz : Hertz

GHz : Giga hertz

t : Zaman (s)

T : Sıcaklık (°C)



KISALTMALAR

AR	: <u>A</u> ugmented <u>R</u> eality – Artırılmış gerçeklik
FOV	: <u>F</u> ield <u>O</u> f <u>V</u> iew – Görüş alanı
GUI	: <u>G</u> raphics <u>U</u> ser <u>I</u> nterface – Grafiksel kullanıcı arayüzü
HCI	: Human Computer Interaction – İnsan bilgisayar etkileşimi
HMD	: Head Mounted Device – Başa takılan sanal gerçeklik aygıtları
HQC	: <u>H</u> ybrid <u>Q</u> uantum- <u>C</u> lassical – Kuantum bilişiminde kullanılan bir dizi kuantum algoritması
HUD	: Head-Up Display – Bir çeşit görüntüleme aygıtı
IBM	: <u>I</u> nternational <u>B</u> usiness <u>M</u> achines Corporation
MR	: <u>M</u> ixed <u>R</u> eality – Karışık gerçeklik
QMI	: <u>Q</u> uantum <u>M</u> achine <u>I</u> mage – Kuantum yazılım uygulamalarının işletilmesini sağlayan sanallaştırılmış programlama ortamı
QPU	: <u>Q</u> uantum <u>P</u> rocessing <u>U</u> nit – Kuantum bilgisayarlarında kullanılan işlemci birimi
QVM	: <u>Q</u> uantum <u>V</u> irtual <u>M</u> achine – Kuantum komut seti mimarisini işleten bir simülatör
RSA	: <u>R</u> ivest, <u>S</u> hamir, and <u>A</u> dleman – Bir kriptoloji algoritması
SSL	: <u>S</u> ecure <u>S</u> ocket <u>L</u> ayer – Web sunucularına ve internet kaynaklarına güvenli erişimi sağlayan bir ağ güvenlik katmanı
URP	: <u>U</u> niversal <u>R</u> ender <u>P</u> ipeline – Unity programının tasarım şablonlarından
UX	: <u>U</u> ser e <u>X</u> perience – Kullanıcı ve makine arasındaki etkileşimin çıktısı
XR	: Sanal, artırılmış ve karışık gerçeklik yaklaşımlarının tümünün entegre edildiği teknolojiler bütünü
VR	: Virtual Reality – Sanal gerçeklik

ŞEKİL LİSTESİ

SAYFA

Şekil 1.1 Etkileşim tasarım kuramının beş boyutu	7
Şekil 1.2 UX Tasarımında bilişsel psikoloji	8
Şekil 1.3 Multimedya sistemlerinde mesajın alıcıya iletilmesi	9
Şekil 1.4 Karışık gerçeklikte süreklilik	11
Şekil 1.5 Farklı HMD aygıtları için FOV değerleri.....	11
Şekil 1.6 VR, teleprezans ve teleoperasyon sisteminin tipik öğeleri.....	13
Şekil 1.7 Yüksek çerçeve oranına sahip bir VR sistemde gecikme akışı	14
Şekil 1.8 Sanal retina ekranı	14
Şekil 1.9 Oculus Rift'in kullanımı	16
Şekil 1.10 Unity kurulum planı	17
Şekil 1.11 Örnek bir Unity Hub ekranı.....	17
Şekil 1.12 Unity tasarım şablonları	18
Şekil 1.13 Bilişim mimarileri ağacı	20
Şekil 1.14 Süperiletken kübitler üzerine inşa edilmiş bir IBM kuantum bilgisayar ilkesel modeli.....	21
Şekil 1.15 Süperiletken Josephson tünel bağlantısı.....	21
Şekil 1.16 5-kübitlik ibmqx4 rezonatör mimarisi (a) ve kübit bağlantı yapısı (b)	22
Şekil 1.17 Bloch küresi notasyonu	24
Şekil 1.18 Süperpozisyon evresinin temsili gösterimi.....	29
Şekil 1.19 Tek kübitlik Hadamard kapısının etkisinin Bloch küresinde gösterimi	31
Şekil 1.20 trans dönüşümü	31
Şekil 1.21 Bell evresini gerçekleştiren devre	34
Şekil 1.22 Web tabanlı kuantum programlamanın ilkesel işleyişi	36

Şekil 1.23 Kuantum bilgisayar programlama tercih menüsü	38
Şekil 1.24 İlk nesil IBM kuantum besteleycisinin görüntüsü	38
Şekil 1.25 Temmuz 2021 tarihi itibarıyla güncel kuantum besteleycisi arayüzü.....	39
Şekil 1.26 Örnek bir IBM Quantum Experience çalışması	39
Şekil 1.27 Devre besteleycisi üzerinde düzenleme işlemi gerçekleştirme	40
Şekil 1.28 Tasarım için histogram çıktısı	40
Şekil 1.29 Jupyter Notebook ile IBM kuantum bilgisayar programlaması	42
Şekil 2.1 “Burning Glass Technologies” firmasına göre 2021 yılında en hızlı gelişim göstereceği tahmin edilen yetkinlikler.....	44
Şekil 2.2 IBM Kuantum bilişimi eğitim videoları.....	45
Şekil 2.3 Tasarım için uygun platformun seçimi ve gerekli ayarların yapılması	46
Şekil 2.4 QU-VR eğitim aracının tasarım süreci.....	47
Şekil 2.5 Örnek bir temel kapı uygulaması	49
Şekil 2.6 Tek kübitlik bir süperpozisyon uygulaması	52
Şekil 2.7 Çift kübitlik bir süperpozisyon uygulaması	53
Şekil 2.8 Grover algoritmasının uygulama örneği	54
Şekil 2.9 Teleportasyon sürecinin temsili gösterimi	56
Şekil 2.10 Bell evresinin başlangıç durumuna döndürülmesi	58
Şekil 2.11 Örnek kodlama	58
Şekil 2.12 Buse ve Ali arasında teleportasyon	62
Şekil 3.1 Unity 3D ortamı için yapılan sahne tasarımları.....	63
Şekil 3.2 Unity’de genel tasarım mimarisi	64
Şekil 3.3 Örnek bir tasarım hiyerarşisi	65
Şekil 3.4 VR Eğitim platformu ana sayfası	65
Şekil 3.5 "SceneManager" nesnesindeki "script" dosyaları	68

Şekil 3.6 "XR Grab Interactable" bileşeninin "On Hover Entered" eylemine işletilecek bir yöntem eklenmesi	72
Şekil 3.7 Hakkında dinamik değişken bilgi mesajı verilen kuantum kapılarının tasarım hiyerarşisindeki konumları	73
Şekil 3.8 "StartingUI" tuvalinin ayrıntılı hiyerarşisi	74
Şekil 3.9 "Sphere" isimli alt nesnenin öğeleri	75
Şekil 3.10 VR platformunun örnek görünümü	77
Şekil 3.11 Temel kapı uygulamalarında örnek bir başlangıç ekranı	78
Şekil 3.12 Temel kapı uygulamalarında doğru işlemin gerçekleştirilmesi örneği	79
Şekil 3.13 Temel kapı uygulamalarında ölçme işlemi örneği	80
Şekil 3.14 İki kübitlik sistemde örnek bir başlangıç durumu	81
Şekil 3.15 Hadamard kapıları yerleştirildikten sonra elde edilen süperpoze durumu	82
Şekil 3.16 İki kübitlik bir sistemde ölçme işlemi ve süperpoze durumunun çökmesi ...	83
Şekil 3.17 Çok kübitli ölçme işleminde değişkenlerin ayarlanması.....	84
Şekil 3.18 Çok kapılı bir tasarımda işlem akışı	94
Şekil 3.19 Örnek bir besteleyici ekranı ve QASM çıktısı	96
Şekil 3.20 Kuantum kapılarının hiyerarşik olarak işleme sırası	97
Şekil 3.21 Örnek bir kod çıktısının (a) VR ortamda yazdırılması ve (b) besteleyiciye yüklenmesi.....	98
Şekil 4.1 Gömülü web tarayıcısı ile kuantum besteleyicisinde gezinti	100
EK 1-Şekil 1 Unity Hub Projeler sayfası.....	111
EK 1-Şekil 2 Unity Hub ile isteğe bağlı bir Unity versiyonu yüklemesi	112
EK 1-Şekil 3 Unity versiyonuna göre kurulabilecek modüllerin seçimi.....	112
EK 1-Şekil 4 Unity yükleme asistanı	113
EK 2-Şekil 1 XR Plugin yöneticisinin kurulumu	115
EK 2-Şekil 2 Oculus eklentilerinin yüklenmesi	116

EK 2-Şekil 3 Unity'de paket yöneticisi	116
--	-----



TABLO LİSTESİ

SAYFA

Tablo 1.1 5-kübitlik ibmqx4 işlemcisinin rezonatör ve kübit frekansları	23
Tablo 1.2 IBM kuantum besteleyicisi üzerinde kullanılan kapılar.....	25
Tablo 1.3 Bazı temel kuantum kapılarının matris gösterimleri	27
Tablo 2.1 İki bitlik mesaj iletimi	59
Tablo 2.2 İki kübitlik bir sistemde olasılık genliklerinin dağılımı.....	60
Tablo 3.1 Sahneler arasında gezinmeyi sağlayan kod bloğu.....	66
Tablo 3.2 İlgili kapıya göre bilgi ekranını gösteren kod bloğu	69
Tablo 3.3 Hadamard kapısının canlandırılmasını sağlayan C# kodu	75
Tablo 3.4 Çok kübitli işlemde kübitlerin rastgele olarak ayarlanması.....	84
Tablo 3.5 3 Boyutlu web aracının sahneye eklenmesi ve ön ayarlarının yapılması.....	86
Tablo 3.6 İki kübitlik sistem için rassal genlik değeri üretimi	87
Tablo 3.7 İki kübitlik sistem için kompleks rassal genlik değeri üretimi	88
Tablo 3.8 Kullanılan kapılar ve kübitlerin matris formunda hazırlanması.....	89
Tablo 3.9 Çok kübitli ölçme uygulamasında matris işlemleri.....	92
Tablo 3.10 QASM dosyası için başlık satırlarının oluşturulması.....	96
Tablo 3.11 QASM dosyasına seçili kapı etiketine göre ilgili satırın yazılması	97
Tablo 3.12 Hücrelere karşılık gelen kapı etiketlerinin kod karşılıklarının eklenmesi....	98

1. GİRİŞ

Sanal gerçeklik çalışmaları insan-bilgisayar arayüzü etkileşiminin (Human Computer Interaction – HCI) bir alt dalı olarak karşımıza çıkmaktadır. Bu çalışmalar, Google’ın ilk 100 bilimsel dergisi sıralamasında kendisine yer bulmuş olan (https://scholar.google.com/citations?view_op=top_venues&hl=en) “IEEE/CVF International Conference on Computer Vision” dergisinde de yayınlanmaktadır. IEEE sanal gerçeklik konferanslarını 1993 yılından günümüze ayrı bir alan olarak düzenlemekte ve bu alandaki çalışmaları <http://ieeervr.org/> sitesinden yayınlamaktadır.

1984 yılından günümüze gerçekleştirilen HCI uluslararası konferansı da sanal gerçeklik üzerine ayrı bir konferans düzenlemektedir. Bu alandaki ilk konferansını “1st International Conference on Virtual Reality” başlığıyla 2005 yılında gerçekleştirmiştir. Bu çalışmalar 2007 yılında ikincisi gerçekleştirilen “International Conference on Virtual Reality” isimli konferansta yayınlanmıştır. Bu tarihten sonra, her yıl VR üzerine yayınlanmış tüm makaleler tek bir kitap altında toplanmaya başlanmıştır. İlk iki konferans VR başlığı altında organize edilmişken, 3’üncü ve 4’üncü konferanslar VMR (Virtual and Mixed Reality) başlığı altında gerçekleştirilmiştir. 5’inci konferanstan sonra VAMR (Virtual-Augmented and Mixed Reality) adı altında yürütülmeye devam etmiştir. HCI tarafından organize edilen tüm bu çalışmalara ait bilgilere <http://www.hci.international/> sitesinden erişmek mümkündür. Ayrıca yayınlanmış makalelerin yıllara göre toplandıkları kitaplara <https://www.springer.com/gp> sitesinden erişilebilir.

Sanal gerçeklik konferansları “International Association of Computer Science and Information Technology” birliğince de 2015 yılından günümüze “International Conference on Virtual Reality” başlığıyla yürütülmekte olup, konferanslara ilişkin bilgilere <http://www.icvr.org/> sitesinden erişmek mümkündür.

Eğitimde sanal gerçeklik uygulamalarına ilk olarak 90’lı yıllarda karşılaşılmakta olup, somutlaştırılması zor konuların eğitiminde yaygın olarak kullanılmaktadır. İngilizcesiyle “immersion” olarak adlandırılan ve olayın içine girme hissiyatı kazandırılması süreci olarak tanımlayabileceğimiz yüksek etkileşimin, bir sanal kimya uygulamasında nasıl kullanılabileceğine ilişkin çalışmada (Byrne, 1996) VR’nin bir eğitim aracı olarak kullanımı sunulmuştur. Kavramsallaştırılması zor olan moleküler kimya konularının 3

boyutlu modellerle pekiştirilmesi sonucunda elde edilen başarı çıktıları da (Farias ve ark., 2011) sanal gerçeklik çalışmalarına güzel bir örnektir.

Azuma (1997) yapmış olduğu çalışmada, VR/AR'nin gelecekteki uygulamaları hakkında başarılı bir vizyon çizmiştir ve bugün bunların büyük ölçüde gerçekleştirildiği görülmektedir. Görüntü ekipmanlarının odak ve kontrast açısından sanallık ve gerçekliğin harmanlanması hakkında daha ayrıntılı bir araştırmanın sonuçları da yine Azuma ve ark.nın (2001) çalışmasında sunulmuştur.

İnsan arayüzü çalışmalarının bağımsız bir alt disiplini olarak ortaya çıkan VR/AR uygulamalarının, güvenlik personeli eğitiminden (Fragomeni ve ark., 2015; Bellemans ve ark., 2020) savunma sanayisine (Liu ve ark., 2018), eğlence sektöründen (Thöner, ve Kuiper, 2015), sağlık eğitimine (Andrews, 2011), anaokulu eğitiminden (Campos ve Pessanha, 2011) mesleki eğitime (Lindgren ve ark., 2016; Aviles ve Cruz, 2018) ve mühendislik öğrencilerinin eğitimine (Aviles-Cruz ve Villegas-Cortez, 2019), endüstri 4.0 üretim yönetimi ve planlamasından (Ma ve ark., 2011) otonom araç teknolojilerinin güvenliğine (Nascimento ve ark., 2019) ve iş makinelerinin simülasyonuna (Wu ve ark., 2013) kadar çok geniş bir yelpazede gerçekleştirildiği görülmektedir. Sanal gerçekliğin sahip olduğu bu potansiyel, her gün farklı uygulamalarla karşımıza çıkmakta ve bu disiplinle ilgili çalışmalar her geçen yıl etkisini artırarak devam etmektedir.

Son yıllarda kamuoyunda ve akademik çevrelerde geniş bir ilgi alanı bulan kuantum bilişimi ve kuantum bilgisayarlarında programlama da üzerinde durulması gereken bir eğitim disiplini olarak karşımıza çıkmaktadır.

Klasik bilgisayarların çekirdek frekans hızlarındaki sınırlı koşullar ve transistör sayısının artırılmaması, uygulanabilir kuantum bilgisayar çalışmalarının tasarımını daha da hızlandırmıştır.

Kuantum bilgisayarların belirsizlik ve uyumsuzluk problemleri (Shor, 1997; Bennett ve ark., 1997) kavramı içinde tartışıldığı zamanlardan beri, kuantum bilgisayarlarda ve dolayısıyla kuantum hesaplamada önemli ilerlemeler kaydedilmiştir. Deterministik polinom-zaman (P) veya deterministik olmayan polinom-zaman (NP) problemlerinin çözümünde, klasik bilgisayarların karşılaştığı logaritmik karmaşıklık (Nasar, 2016) problemleri ve alternatif algoritma geliştirme süreçleri uzun süredir tartışılmaktadır.

Bir kuantum bilgisayarda, tam sayılarda çarpanlara ayırma işlemini gerçekleştirmek

üzere geliştirilmiş algoritma (Shor, 1994), IBM tarafından geliştirilmiş 7-qubitlik bir kuantum bilgisayarı ilk kez 2001 yılında, 15 sayısını çarpanlarına ayırmada başarıyla kullanılmıştır (Ömer, 2009).

Her ne kadar deterministik klasik bilgisayarların günümüzde ulaştığı yüksek işlem gücü, bellek alanı kapasitesi ve yüksek doğruluk nedeniyle algoritmaların karmaşıklığının üstesinden gelinebilse de maksimum saat frekans oranlarına çok önceden erişilmiş olmasından dolayı (Kanamori ve ark., 2006), Grover (1996) ve Shor (1997) gibi araştırmacılar tarafından sunulmuş farklı kuantum algoritmalarıyla çok karmaşık süreçler için çok daha hızlı sonuçlar elde edilebildiği görülmüştür (Kanamori ve ark., 2006).

Deutsch ve Roza (1992) tarafından belirtildiği gibi, kuantum hesaplama, herhangi bir klasik deterministik hesaplama katlanarak daha kısa sürede problemi kesin olarak çözme gücüne sahiptir. Kuantum bilgisayarlar, çıktısı girdiler tarafından belirlenen bir fonksiyona ihtiyaç duymaz.

Yakın zamana kadar yüksek bitli kuantum algoritmaları gerektiren RSA şifreleme algoritmalarını simüle etmek mümkün değildi. Bugün kuantum algoritmalarını çalıştırmak için gerçek kuantum bilgisayarları kullanmak mümkündür.

Süperiletken kuantum işlemcilerinin tasarımında yaşanan rekabet beraberinde kuantum üstünlüğü kavramını da ortaya çıkarmıştır. Hesaplama güçlerinin üstel bir oranda artmaya devam edeceği belirtilmektedir (Arute ve ark., 2019). Harrow ve Montanaro (2017) tarafından belirtildiği gibi, Shor'un algoritmasını çalıştırabilecek büyük ölçekli bir kuantum bilgisayarı, klasik bir bilgisayar üzerinde kuantum üstünlüğünü kanıtlama noktasında 2048 bitlik bir RSA anahtarını çarpanlarına ayırabilir.

Kuantum bilgisayar tasarımının tüm mantıksal süreci, kuantum bilgisayarların olasılıksal gücü kullanılarak polinom zamanlı algoritmaların katlanarak daha kısa zamanlı çözümünün gerçekleştirilmesine dayanmaktadır. Belirli bir sorunu çözmek için bilgisayar, algoritma (Ekert ve Jozsa, 1996) adı verilen kesin bir talimat dizisini takip eder. Büyük bir tamsayıyı büyük- $O(\log n)$ ile çarpanlarına ayırmaya yönelik bir kuantum algoritmasının, klasik bilgisayarlar tarafından kullanılan bir algoritmadan katlanarak daha hızlı olduğu kanıtlanmıştır (Kanamori ve ark., 2006).

2020 yılında tüm Dünyayı etkisi altına almış olan Covid-19 virüsünün belirlenmesi için yapılan yakın zamanlı bir çalışmada da (Acar ve Yılmaz, 2021), kuantum transfer

öğrenmesi algoritması kullanılarak elde edilen çıktıların klasik yapay sinir ağları algoritmalarına karşın daha yüksek doğruluk oranıyla tahmin gücü ortaya koyduğu belirtilmiştir.

Kuantum hesaplama, bir hesaplamayı daha verimli bir şekilde yürütmek için süperpozisyon, dolanıklık ve kuantum tünelleme gibi kuantum mekanik etkilerden yararlanır. Geleneksel, dijital hesaplama ile karşılaştırıldığında, kuantum hesaplama, zaman ve enerji tüketimi açısından da çok önemli bir avantaj sunar (Abhijith ve ark., 2020).

Günümüzde IBM, D-Wave, Rigetti gibi şirketler, gelişen bulut programlama altyapısını kullanarak, kuantum bilgisayarlarını internet üzerinden deneysel kullanım amacıyla kullanıcıların programlayabilmesine açmışlardır. Eğitimde sınırsız fırsatlar sunan sanal gerçeklik teknolojileri, bulut temelli kuantum programlama platformlarının daha derinlemesine bir etkileşimle kullanılmasını sağlayabilir.

Klasik bilgisayar programlama dillerinde olduğu gibi kuantum programlamanın kodlanmasını daha okunaklı ve kullanıcı dostu hale getirmek için kuantum bilgisayarlarıyla programlama üzerine çeşitli çalışmalar devam etmektedir (Bichsel ve ark., 2020). Bu kapsamda, sanal gerçekliğin yeni bir uygulaması olarak, kuantum bilgisayarlarının VR temelli programlanması için özgün ve yeni bir tasarım gerçekleştirilmiştir. Bu çalışmada, klasik programlamadan oldukça farklı olan kuantum programlamanın ilkeleri de temel hatlarıyla anlatılmış, kuantum bilişimi matematiğinin VR temelli tasarım ortamına nasıl aktarıldığı programlama ilkeleri kapsamında uygulamalı olarak açıklanmıştır.

1.1. Araştırmanın Amacı

Bu tezin amaçları ve hedefleri;

- Kuantum bilgisayarların programlanmasıyla ilgili güncel teorik bilgileri araştırmak,
- Çevrimiçi bağlantıyla çıktılarına erişilebilen ortamdan bağımsız gerçek kuantum bilgisayarların bulunduğu bir platformda programlama işlemlerini gerçekleştirmek,

- Üniversite ve ülke kaynakları ile elde edilmesi zor olan kuantum bilgisayarların programlama işlemlerinin yapılabilmesi için temel kuantum kapılarıyla sanal gerçeklik ortamında işlem yapan yerli ve dünyada ilk programlama ortamını tasarlamak ve gerçekleştirmek,
- Temel kuantum kapılarının fiziksel ve matematiksel çıktılarının sanal gerçeklik ortamında simülasyonunu yapmak,
- Herhangi bir şekilde kuantum bilgisayarlara ve çevrimiçi programlama ortamlarına erişilemediği durumda eğitim altyapısının oluşması için gerekli olan kuantum besteleyicisinin ve kod dönüştürücüsünün gerçekleştirilmesini sağlamak,
- Gelecekte kuantum bilişimi dahil, oluşturulabilecek tüm çevrimiçi programlama ortamlarının benzetilmesi ve gerçekleşmesi için gerekli olan yerli sanal gerçeklik ortamını gerçekleştirmek,
- Dezavantajlı bireyler dahil tüm bireylerin vücudunu kullanarak kendisini işbaşındaymış gibi hissettirerek sanal gerçeklik ortamında kuantum bilgisayar programlama işlemini yaptırmak ve öğretmektir.

1.2. Literatür Araştırması

Bu tez hazırlanırken aşağıdaki adımlarda kaynak araştırması ve incelemesi yapılmıştır:

- İnsan bilgisayar etkileşimi ve arayüzü tasarımı ilkeleri araştırılmış ve incelenmiştir.
- Sanal ve artırılmış gerçeklik çalışmaları araştırılmış ve incelenmiştir.
- Mühendislik eğitiminde farklı disiplinler açısından sanal ve artırılmış gerçeklik uygulamaları araştırılmış ve incelenmiştir.
- Sanal gerçeklik ortamında bilgisayar programlama ilkeleri araştırılmıştır.
- Kuantum mekaniğinin temelleri araştırılmıştır.
- Kuantum bilişiminin kuantum mekaniği ile ilişkisi araştırılmıştır.
- Kuantum bilişimi konusunda yapılmış çalışmalar incelenmiştir.
- Bulut bilişim ağı üzerinden kuantum bilişimi desteği sağlayan firmaların hizmetleri incelenmiştir.
- Kuantum bilişiminin temelleri ve temel kuantum algoritmaları incelenmiştir.
- Kuantum kapılarıyla kuantum bilişimi uygulamaları incelenmiştir.

- 3B oyun tasarımı kapsamında kuantum bilişimi yaklaşımları araştırılmış ve incelenmiştir.

1.2.1. İnsan Bilgisayar Etkileşimi (HCI)

Kullanıcıların (insanlar) çeşitli ihtiyaçlara ve uygulamaların şekline göre birçok farklı cihaz ile etkileşime girmesi gerekmektedir. Araştırmacılar ve tasarımcılar, kullanıcıları memnun etmek için bir dizi yazılım ve donanım teknolojisinden yararlanmaktadır. Saatlerden akıllı telefonlara, televizyonlardan bilgisayarlara, tabletlerden oyun konsollarına, dijital içerik okuyuculardan spor aktivite monitörlerine, araba konsollarından sanal gerçeklik arayüz cihazlarına kadar karşımıza çıkan tüm interaktif cihazlar insan-bilgisayar etkileşiminin örnekleridir. Günlük hayatımızda son derece önemli bir yere sahip olan bu kadar çok cihazı kullanıcıların kullanmasını sağlayacak bir arayüz tasarımı uzmanlık gerektiren bir iştir. Bu noktadan hareketle, üç boyutlu bir sanal gerçeklik temelli eğitim platformu tasarımı, çalışma kapsamında tüm süreçleriyle ayrıntılı olarak sunulmuştur.

1.2.1.1 Etkileşim Uzayının Tasarımı

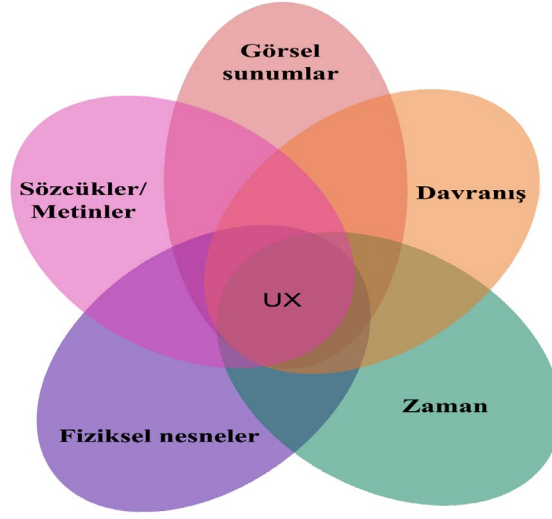
Uygun bir etkileşim tasarımının temel amacı, kullanıcının işi en pratik şekilde yapmasını sağlamaktır. Günlük kullanımımızda etkileşimde bulunduğumuz cihazların bir listesini yaparsak listenin bitmeyeceğini söyleyebiliriz. Coiera'nın (2003) işaret ettiği gibi, etkileşim tasarımı, insanların nesneler ve sistemlerle etkileşim biçimini yaratmaya odaklanır ve etkileşim tasarımının ürünü neredeyse tamamen kullanıcı deneyiminden elde edilen çıktıdır.

Sanal gerçeklik tasarımları da HCI çalışmalarının bir parçası olarak, etkileşim tasarım kuramının (interaction design theory) beş temel boyutunun göz önünde bulundurulmasını gerekli kılar. Bu ilkeleri şu şekilde listeleyebiliriz:

- Görsel sunumlar (Visual Representations),
- Sözcükler/Metinler (Words/Texts),
- Fiziksel nesneler (Physical Objects),
- Zaman (Time) ve
- Davranış (Behaviour).

Tüm bu beş ilke kullanıcı deneyimini (User eXperience – UX) biçimlendirir (Sharp ve

ark., 2019) ve Şekil 1.1’deki gibi gösterilebilir.

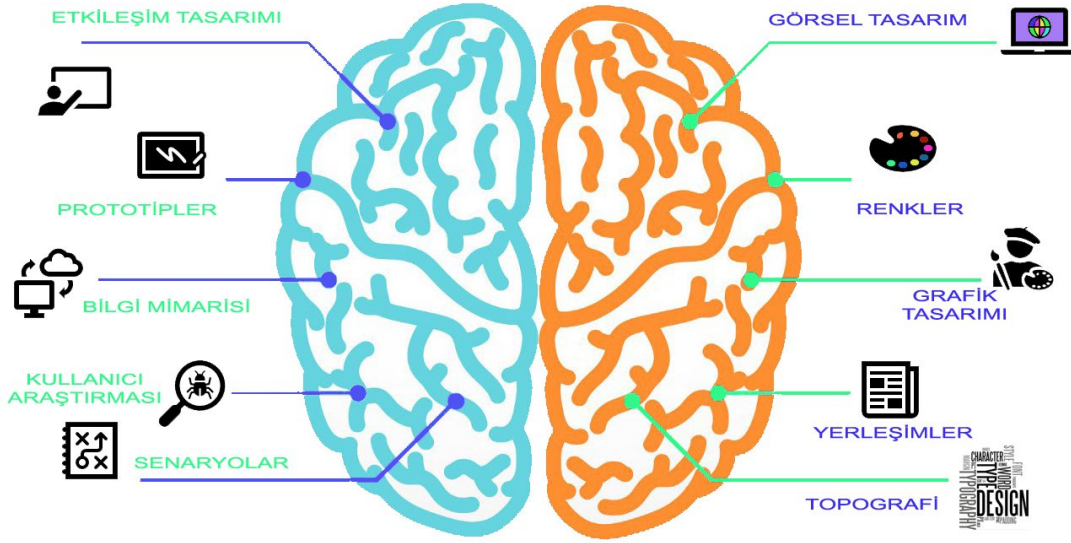


Şekil 1.1 Etkileşim tasarım kuramının beş boyutu

Kolko (2011) tarafından da ifade edildiği üzere, “Etkileşim tasarımı kişi ve ürün, sistem ya da hizmet arasında bir diyalog yaratma sürecidir. Bu diyalog doğası gereği hem fiziksel hem de duygusaldır ve zaman içinde deneyimlendiği şekliyle biçim, işlevler ve teknoloji arasındaki etkileşimde kendini gösterir.”

Etkileşim tasarımı, kullanıcı ve ürün arasındaki etkileşim/iletişim tasarımıdır. Bu ürün bir web sayfası veya tablet veya akıllı telefon gibi cihaz ekranları olabilir. Etkileşim tasarımı yalnızca bir kullanıcı arayüzü animasyonu değildir. Yukarıda listelendiği gibi, etkileşim tasarımı söz konusu olduğunda daha fazlası gereklidir. Bilişsel psikoloji ile İnsan Bilgisayar Etkileşimi arasında da bir ilişki vardır. Hewet'in (1999) belirttiği gibi, bilişsel psikoloji, yapay zekâ ve insan-bilgisayar iletişimi dahil olmak üzere bilgisayar biliminin dokuz temel alanından üçü için önemli bir destekleyici disiplindir.

Şekil 1.2'de gösterildiği gibi, verileri insan beyninin bilişsel aleminden alıp bilgisayarın dijital aleminden çıkarmak, fiziksel alemde tek bir tutarlı görsel yapıda somut hale getirmek, sizi insan beyninin hafıza sınırlarından ve teknolojinin yapay sınırlarından kurtarır.



Şekil 1.2 UX Tasarımında bilişsel psikoloji

Sözcükler kısaca, komut butonlarındaki metinler gibi kullanıcıyı yönlendirecek basit metinleri tanımlar. Görsel temsil; simgeler ve resimler gibi grafik öğelerle ilgilenir. Genellikle metin temelli bilgileri desteklemek için kullanılırlar. Fiziksel nesneler; kullanıcıların ürünle etkileşime girmesi için kullanılan giriş ve çıkış donanımına karşılık gelir. Bu donanım ürünleri fare, dokunmatik yüzey, dokunmatik ekran, el kumandaları, dokunsal sensörler vb. olabilir.

Animasyonlar, videolar, ses vb. gibi zamanla değişen medya, kullanıcının ürünle heyecan verici bir şekilde etkileşim kurmasına her zaman yardımcı olur. Zaman ilerleme animasyonları, kullanıcıya bu deneyim hakkında hoş bir his vermelidir. Beşinci boyut davranıştır. Davranış tüm eylemler, tepkiler, işlemler ve sunumlarla ilgilidir. Başka bir deyişle, uygulamadan elde edilecek geri bildirim ve çıktıyı ifade eder. Davranış boyutu, adapte edilebilir ve kullanıcılarca kolay anlaşılabilir şekilde tasarlanmalıdır.

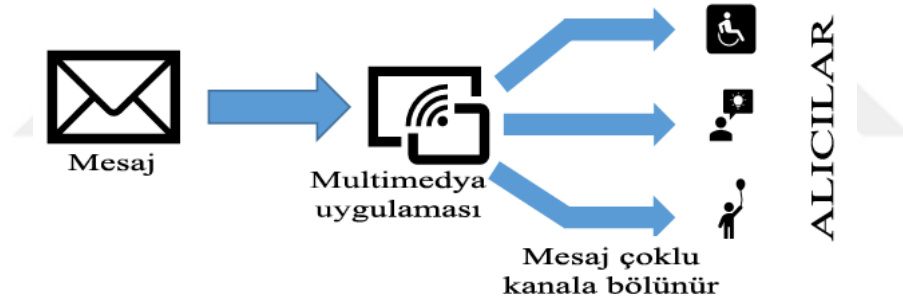
1.2.1.2 Etkileşim Tasarımında Dikkat Edilmesi Gereken Diğer Noktalar

HCI uygulaması tasarlarlarken bir diğer önemli nokta ise uygulamayı hemen hayata geçirmemenizdir. Öncelikle prototipler tasarlanmalıdır. Tasarım süreci birkaç dereceden aslına uygunluk (fidelity) içerir ve bunlara düşük, orta ve yüksek aslına uygunluk prototipleri denir (de Sá ve ark., 2008).

Tasarlanan ürünün ölçümü ve alan testlerini gerçekleştirmek te onu tasarlamak kadar

önemli bir husustur (Sharp ve ark., 2019). Bunun için HCI ürünlerinin değerlendirilmesinde yararlanılan Standart Kullanışlılık Anketleri'nden faydalanılabilir. <https://garyperلمان.com/quest/quest.cgi> sitesinden erişilebilen “Computer System Usability Questionnaire” anketi ve <https://garyperلمان.com/quest/quest.cgi?form=USE> sitesinden erişilebilen “Usefulness, Satisfaction, and Ease of use Questionnaire” anketi bu tarz ürünlerin değerlendirilmesinde kullanılabilir. <http://2017.hci.international/t13> sitesinde, geçerlilik ve güvenilirlik testleri yapılmış çok sayıda standart anket yer almaktadır.

Etkileşim alanını inşa etmek için, biri insan diğeri makine olmak üzere iki faktörün etkileşimini kurmak gerekir. Tasarım, bir etkileşimin diğeri üzerindeki etkisi dikkate alınarak yapılmalıdır (Coiera 2003). Johnson'ın (2003) dediği gibi, bir diğeri önemli konu da mesajın iletildiği kanaldır. Metin, ses, infografik, animasyon ve video gibi iletişim kanalları alıcı üzerinde duygusal bir etki yaratır. Şekil 1.3'te gösterildiği gibi, iletilecek mesaj farklı kanallara bölünebilir ve farklı alıcılar üzerinde farklı etkileri olabilir.



Şekil 1.3 Multimedya sistemlerinde mesajın alıcıya iletilmesi

Mehlenbacher'e (2002) göre, multimedia araçlarını tasarlarken dikkat edilmesi gerekenler şu şekilde listelenebilir:

- Etkileşimin en önemli amacı iletilecek mesajın güçlendirilmesidir.
- Kullanıcının yalnızca ihtiyaç duyduğunda kullanacağı bir uygulama iyi planlanmış bir etkileşim sunar.

Kısacası, insan bilgisayar etkileşimi ya da insan iletişim arayüzü tasarımı çok disiplinli bir yaklaşımı gerekli kılar.

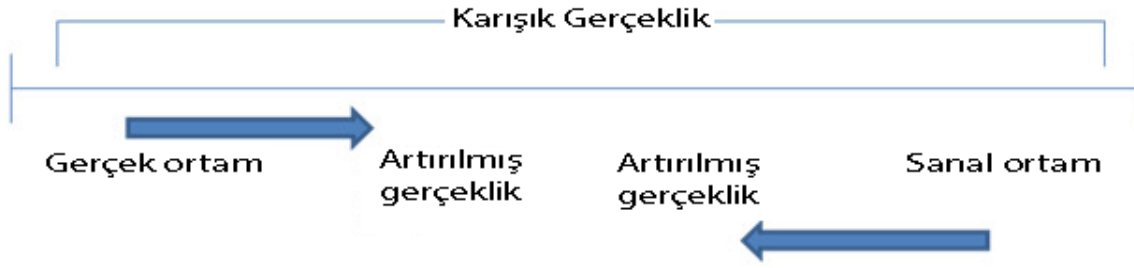
1.2.2. Sanal Gerçeklik ile İlgili Genel Bilgiler

Çalışma sürecinde, sanal ve artırılmış gerçeklik uygulamalarının gelişim süreci ve farklı

sektörlerde uygulama alanları incelenerek hangi ihtiyaçlardan kaynaklı olarak geliştirildikleri ortaya konmuş ve bu kapsamda, çeşitli terimlerin yaygın olarak kullanıldıkları, zaman zaman terimlerin kullanımında kavramsal karışıklıklar olduğu, bazı terimlerin birbirleriyle ya çok iç içe ya da birbirlerinin anlamlarını karşılayacak şekilde kullanıldıkları, bu nedenle çalışma sırasında çok sık olarak karşılaşılan terimlerin kısaca açıklanmalarında fayda olacağı değerlendirilmiştir.

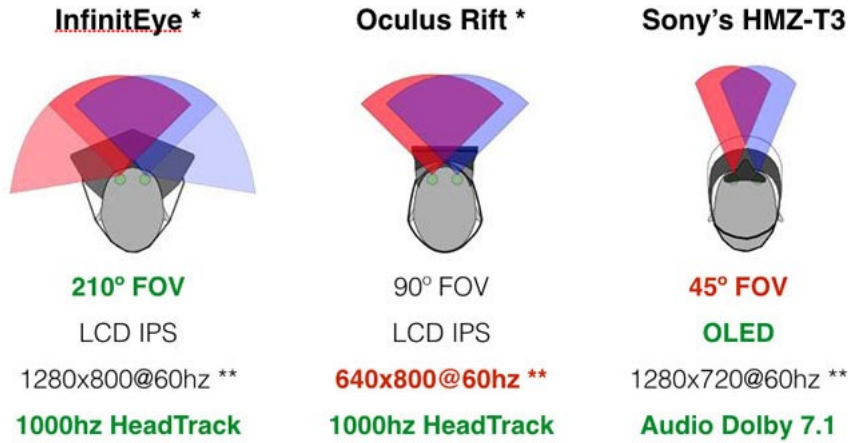
1. **Simulation – Benzeşim:** Gerçek dünyadaki bir işlem ya da sistemin taklit edilmesiyle oluşturulmuş yapay gerçeklemedir. Benzeşim teknolojilerinde öncelikle benzeşimi yapılacak sistemin modeli çıkartılır. Model seçili fiziksel ya da soyut sistemin-işlemin özelliklerini, davranışlarını ya da temel niteliklerini temsil eder. Model sistemi temsil ederken simülasyon ise sistemin davranışını temsil eder (CAD/CAM yazılımları, elektrik devre simülasyon yazılımları, vb.).
2. **Virtual Reality (VR) – Sanal Gerçeklik:** Bilgisayar destekli teknolojilerle 3 boyutlu sanal bir ortama dönüştürülen çevresel değişkenler, girdiler ve çıktıların, benzeşim ve canlandırma araçlarıyla kullanıcıya gerçek bir tecrübe hissini yaşatmak için kullanılmasıdır.
3. **Augmented Reality (AR) – Artırılmış Gerçeklik:** Kişilerin gerçek dünyadaki ortama ilişkin algısını ve tecrübesini, gerçek dünyadan alınan görüntülere çeşitli aygıtlardan gelen verileri (ses, görüntü, konum, vb.) bindirmek ve çeşitli sensörlerle desteklemek suretiyle, zenginleştirme işlemidir. Sanal gerçeklikte gerçek bir ortamın bilgisayar destekli benzetişimi ile yer değiştirilmiş olması durumu varken, artırılmış gerçeklikte ise dijital dünyanın nesnelerinin daha sürükleyici bir gerçeklik hissi oluşturmak için kişinin doğal çevre algısına eklenmesi durumu vardır.
4. **Mixed Reality (MR) – Karışık Gerçeklik:** Karışık gerçeklik ise VR ve AR teknolojilerinin birlikte harmanlandığı, gerçek bir fiziksel ortama 5 duyuya da hitap eden çeşitli sensörlerden oluşan aygıtlarla müdahil olduğu ve bu fiziksel ortamın bilgisayarlarca üretilen sanal gerçeklik nesneleriyle zenginleştirilerek kullanıcıya gerçek zamanlı olarak aktarılan hibrid bir sistemi tanımlar.

Karışık gerçekliğin sürekliliğini (continuum) kısaca aşağıdaki şekilde gösterebiliriz. Şekil 1.4'te görüldüğü üzere çift yönlü bir akış durumu söz konusudur.



Şekil 1.4 Karışık gerçeklikte süreklilik

5. **Başa Takılı Ekran (Head Mounted Display):** Gerçek dünyanın çeşitli ses efektleri, farklı hislere hitap eden aktüatörler ve 3-D görüntü yazılımlarıyla zenginleştirilmiş bir taklidinin yapıldığı görsel şölenin kişiye sunumunda kullanılan en önemli aracı teknolojidir.
6. **FOV (Field Of View):** Herhangi bir anda etkin olarak görülebilir alanı ifade eder. Optik izleme aygıtlarının belli bir konumdan (yatay, düşey, çapraz) üç boyutlu uzayda tek seferde alabildikleri görüntünün iki boyutlu açısıdır.



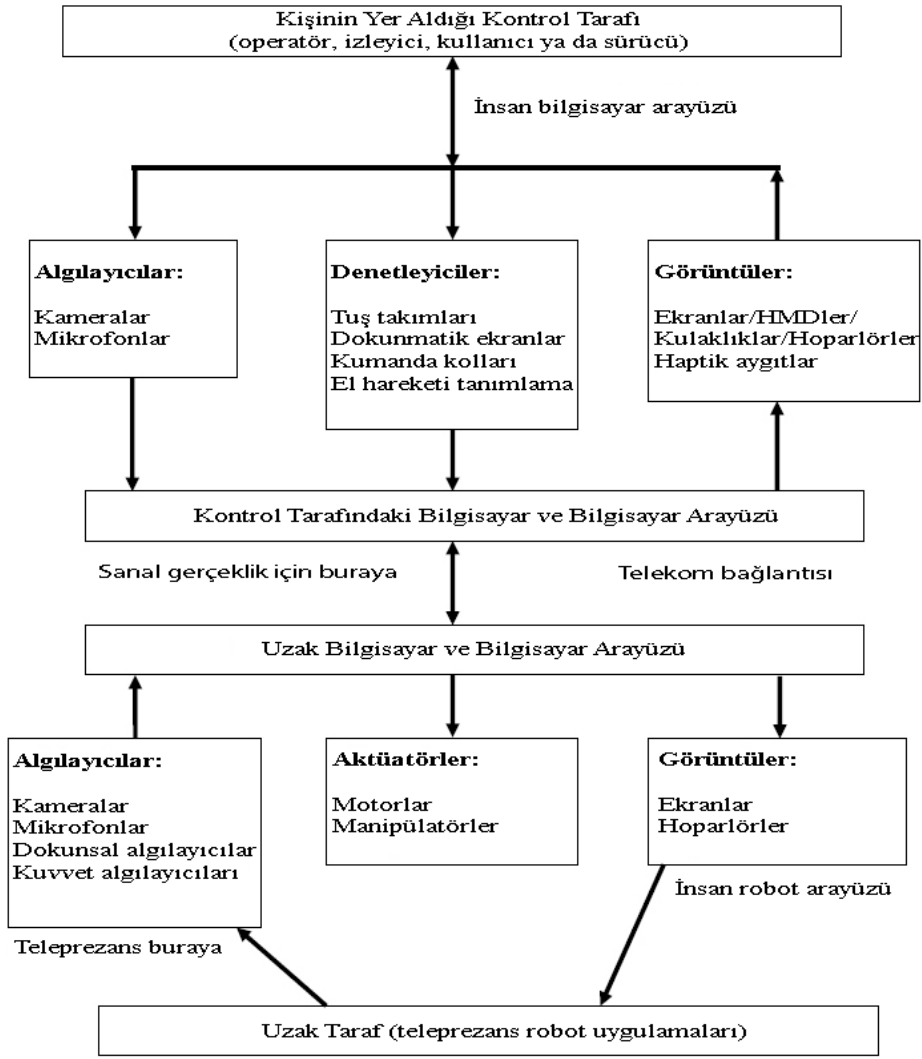
Şekil 1.5 Farklı HMD aygıtları için FOV değerleri

7. **HUD (Head-Up Display):** İlk olarak pek çok teknolojide olduğu üzere savaş uçakları için geliştirilmiş şeffaf bir görüntüleme ekranıdır. 1950'li yıllarda geliştirilmiştir. Pilotun gerekli bilgileri başını etrafa çevirmek zorunda kalmadan ekran üzerinden canlı olarak görmesini sağlar. Günümüzde ticari uçaklarda ve arabalarda da kullanılmaktadır.
8. **Dokunsal (Haptic) sistemler:** Kullanıcının vücut hareketlerini aktarıp ona göre işlem yapmasını sağlayan sistemlerdir. Günümüz gelişmiş sanal gerçeklik

aygıtlarında geri besleme sağlayan ve böylece dokunma hissini bize aktaran geri beslemeli dokunsal sistemler mevcuttur. Oyun konsolları bunlara örnek verilebilir.

9. Bilgisayar destekli senaryo: Sanal gerçekliğin en önemli ögesi bilgisayar destekli senaryo üretimidir. Kişilerin gerçek dünyaya benzer bir tecrübeyi yapay olarak yaşamasına imkân verecek bir görsel, işitsel ya da dokunsal ortam üretimini ifade eder.

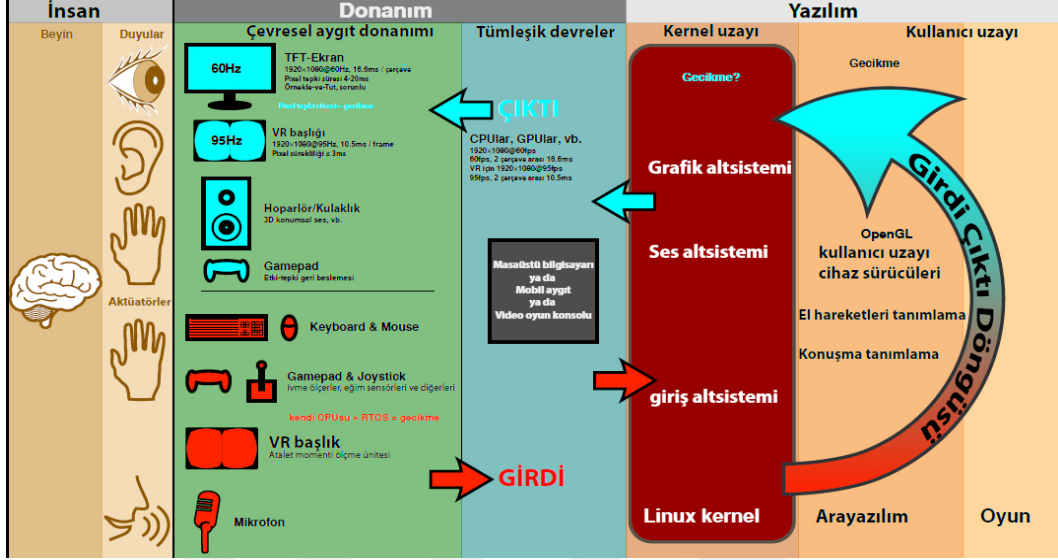
10. Teleprezans (telepresence): Televizyon kelimesine benzer olarak, kelime anlamı bakımından “uzakta varoluş” şeklinde ifade edebiliriz. Sanal gerçeklik ile uzak makine kontrolü ya da uzaktaki olaylara görünür katılım sağlama durumu için kullanılan bir terimdir. Özellikle tıp alanında, ameliyatı gerçekleştiren doktorun sanal gerçeklik aygıtlarıyla ameliyatı ameliyat odasına girmeden yapması durumu “teleprezans” olarak adlandırılabilir. Şekil 1.6’da bu durumun akış şeması gösterilmiştir.



Şekil 1.6 VR, teleprezans ve teleoperasyon sisteminin tipik öğeleri

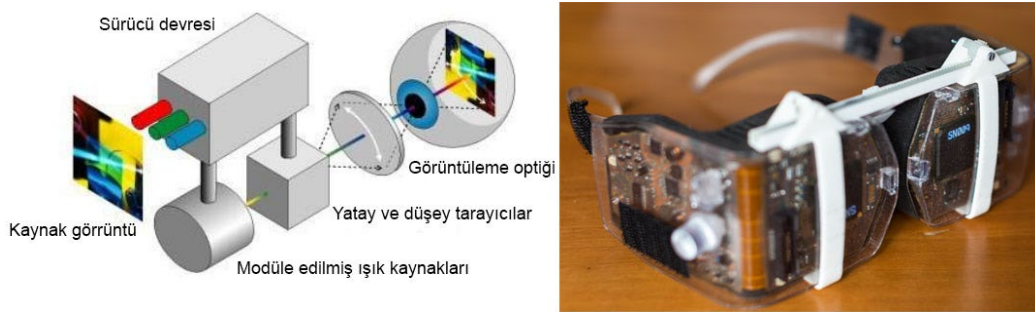
11. **İmage Registration:** Görüntü tescilleme, farklı veri kümelerini tek bir koordinat sistemine dönüştürme işlemidir. AR uygulamaları için son derece önemli bir işlemdir.
12. **İmmersiyon (immertion):** Bir olayın ya da bir senaryonun içinde yer alma durumunu ifade eder. VR bir sistemde, fiziksel olmayan bir dünyanın içinde fiziksel olarak varmışsınız algısının oluşturulması durumudur. Dokunsal, stratejik, öyküsel ya da uzaysal immersiyon gibi türleri vardır.
13. **Gecikme (latency):** Aksiyon ve reaksiyon arasındaki duraksama (etki-tepki) süresi olarak tanımlanabilir. VR-AR bir sistemde gecikme süresi sistemin kararlılığını etkileyen çok kritik bir unsurdur. HMD takılıyken başınızı

oylattığınızda meydana gelen değişim anında ekranda gösterilebilmelidir. Şekil 1.7’de, yüksek çerçeve oranına sahip bir VR sistemde Linux çekirdeğinin gecikme akışı gösterilmiştir.



Şekil 1.7 Yüksek çerçeve oranına sahip bir VR sistemde gecikme akışı

14. Virtual Retinal Display (VRD): VR ve AR teknolojileri içinde günümüzün belkide en yeni görüntüleme teknolojisidir. Washington Üniversitesi “Human Interface Technology Laboratory” laboratuvarınca geliştirme süreci devam etmektedir. Bu zamana kadar bildiğimiz tüm görüntüleme aygıtlarından farklı olarak görüntünün doğrudan kullanıcının retinasına taranması ilkesi söz konusudur.



Şekil 1.8 Sanal retina ekranı

15. SAR (Spatial Augmented Reality): Mekansal artırılmış gerçeklikte bildiğimiz görüntüleme aygıtları kullanılmaz. Dijital projektörler yardımıyla, fiziksel

nesnelerin üzerinde karmaşık görüntülerin yansımalarının oluşturulması durumu vardır.

16. İzleme (tracking): Modern mobil artırılmış gerçeklik sistemleri bir ya da daha fazla izleme teknolojisini (dijital kameralar, optik sensörler, ivmeölçerler, GPS, cayroskop, katı hal pusulaları, RFID, vb.) kullanır.

17. VIO (Visual Inertial Odometry): Görsel atalet odometrisi hareketli nesnelerin (ör: robot kolu) zaman içindeki muhtemel pozisyonunu tahmin etmede kullanılmaktadır.

1.2.3. VR/AR Tasarımında Kullanılan Platformlar

VR/AR temelli uygulama geliştirme sürecinde çok yaygın olarak kullanılan çeşitli platformlar vardır. Bunlardan en sık tercih edilenler, Epic Games firmasının geliştirdiği oyun motorlarından biri olan “Unreal Engines 4” ve Unity Technologies firması tarafından geliştirilmiş olan “Unity 3D” oyun motorlarıdır. Bu oyun motorları uzunca süredir 3D oyun tasarımında başarılı şekilde kullanılmakta olup, VR/AR temelli uygulamaların tasarımında da son yıllarda sıklıkla tercih edilmektedirler (Christopoulou ve Xinogalos, 2017). Çalışmada tanıtımı yapılan tasarım Unity 3D kullanılarak gerçekleştirilmiştir.

Üç boyutlu ortam tasarımları yüksek işlemci ve bellek gereksinimi gerektirdiğinden genel olarak bu tarz ortamların tasarımında oyun motorları tercih edilmektedir. Unity yazılım şirketi (Unity Technologies) 2D ya da 3D tasarım platformlarını bünyesinde bulunduran ve 2004 yılında Danimarka’nın Kopenhag şehrinde “Over the Edge Entertainment-(OTEE)” olarak kurulmuş bir oyun tasarım şirkettir (https://en.wikipedia.org/wiki/Unity_Technologies).

Unity şirketi ilk olarak Macintosh bilgisayarlar için geliştirdiği oyunlarla adını duyurmuştur. Ardından, 2007’li yıllarda Apple ve sonrasında Android temelli akıllı telefon ve diğer mobil cihazların yaygınlaşmasıyla birlikte, bu cihazlara yönelik olarak sunduğu geniş oyun geliştirme seçenekleriyle hızlıca yaygınlaşmıştır. Unity’nin popüler olmasındaki başlıca neden, mobil işletim sistemlerinden, masaüstü işletim sistemlerine kadar çok zengin bir platform desteği sunmasından ileri gelmektedir (Haas, 2014). 2015 yılından itibaren de birtakım sanal gerçeklik donanımına destek vererek platform

yelpazesini genişletmiştir.

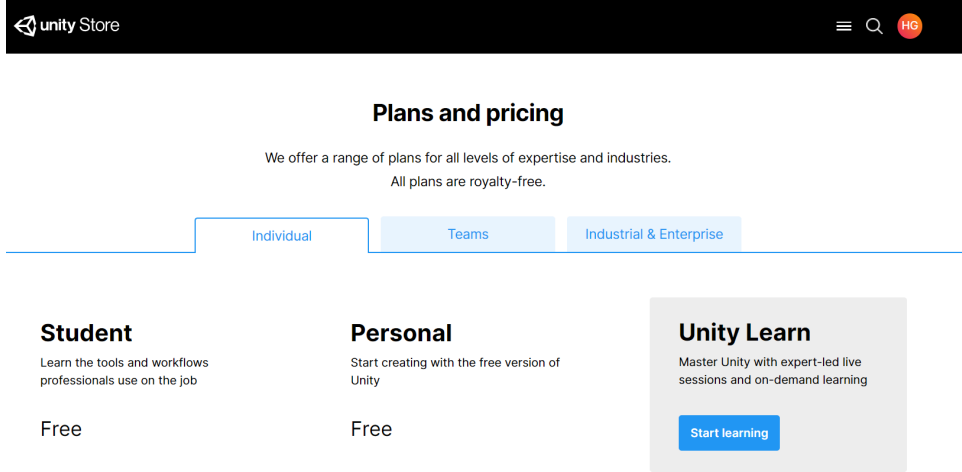
Çalışmada, sanal gerçeklik donanımları üreten firmaların başında gelen Oculus firmasının, Windows platformu için geliştirdiği Oculus Rift HMD seti kullanılmıştır (bkz. Şekil 1.9). Unity gelişmiş XR tasarım paketi ile çeşitli VR başlıkları için uygulama geliştirme desteği sunar. Uygulama geliştirme desteği sunulan başlıklardan birisi de 2016 yılında tüketici pazarına sunulan Oculus Rift başlığıdır. Bu başlığın iki adet el kumandası ve iki adet lazer sensörü bulunur. Lazer sensörler yardımıyla kullanıcı için sınır alanı oluşturulur.



Şekil 1.9 Oculus Rift'in kullanımı

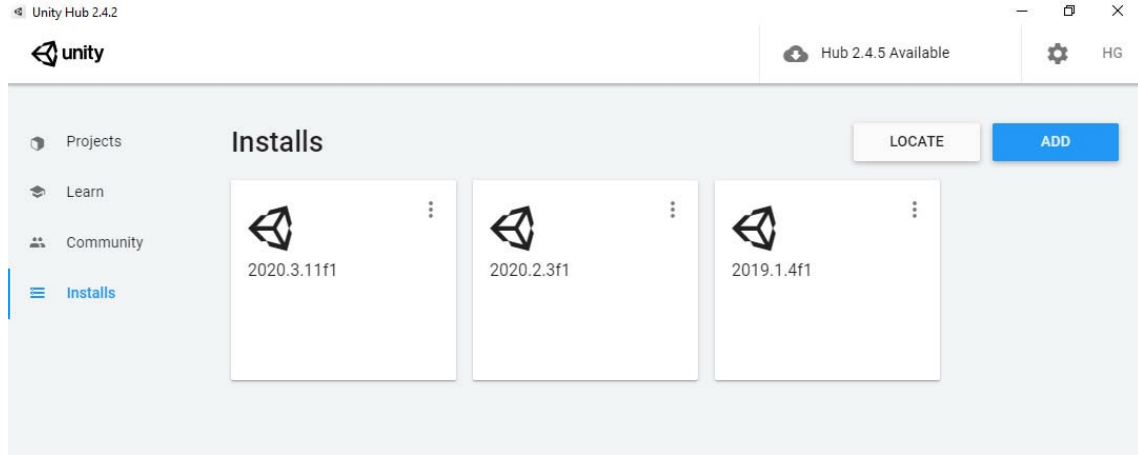
1.2.3.1. Unity Programının Kurulumu

Unity her kesim için geniş bir kurulum seçeneği sunar. Şekil 1.10'da gösterildiği üzere, "Student" ve "Personal" kurulum seçenekleri ücretsiz (free) olup, tasarım takımları, endüstriyel ve ticari tasarım kuruluşları için geliştirilmiş profesyonel sürümler ücretlidir. Unity hizmetlerinden sağlıklı şekilde yararlanabilmek ve uygulamayı etkin kullanabilmek için <https://id.unity.com/> sitesinden kullanıcı hesabı oluşturulması gerekir. Unity'nin zengin bir ürün mağazası bulunmaktadır ve bu mağazadan ücretsiz ya da ücretli çeşitli paketleri projenize indirmeniz mümkündür. Ancak bu işlemin yapılabilmesi için kullanıcı hesabı oluşturulması gerekmektedir.



Şekil 1.10 Unity kurulum planı

Unity geniş bir sürüm havuzuna sahiptir. Bu sürüm havuzundan istediğiniz bir sürümün kurulabilmesi için, Unity firması Unity Hub uygulamasının kurulmasını tavsiye etmektedir. İşletim sistemi ve bir takım grafik yazılımı güncelleştirmelerine ya da kullanıcı dosyalarının senkronizasyonuna bağlı olarak Unity Hub ile sürüm kurulumu sorunları yaşanabilmektedir. Bu tür durumlarda Unity programını kendi başına (standalone) kurulum dosyalarıyla da kurmak mümkündür. Şekil 1.11’de, çalışmada kullanılan yüklü sürümleri gösteren Unity Hub ekranı gösterilmiştir.



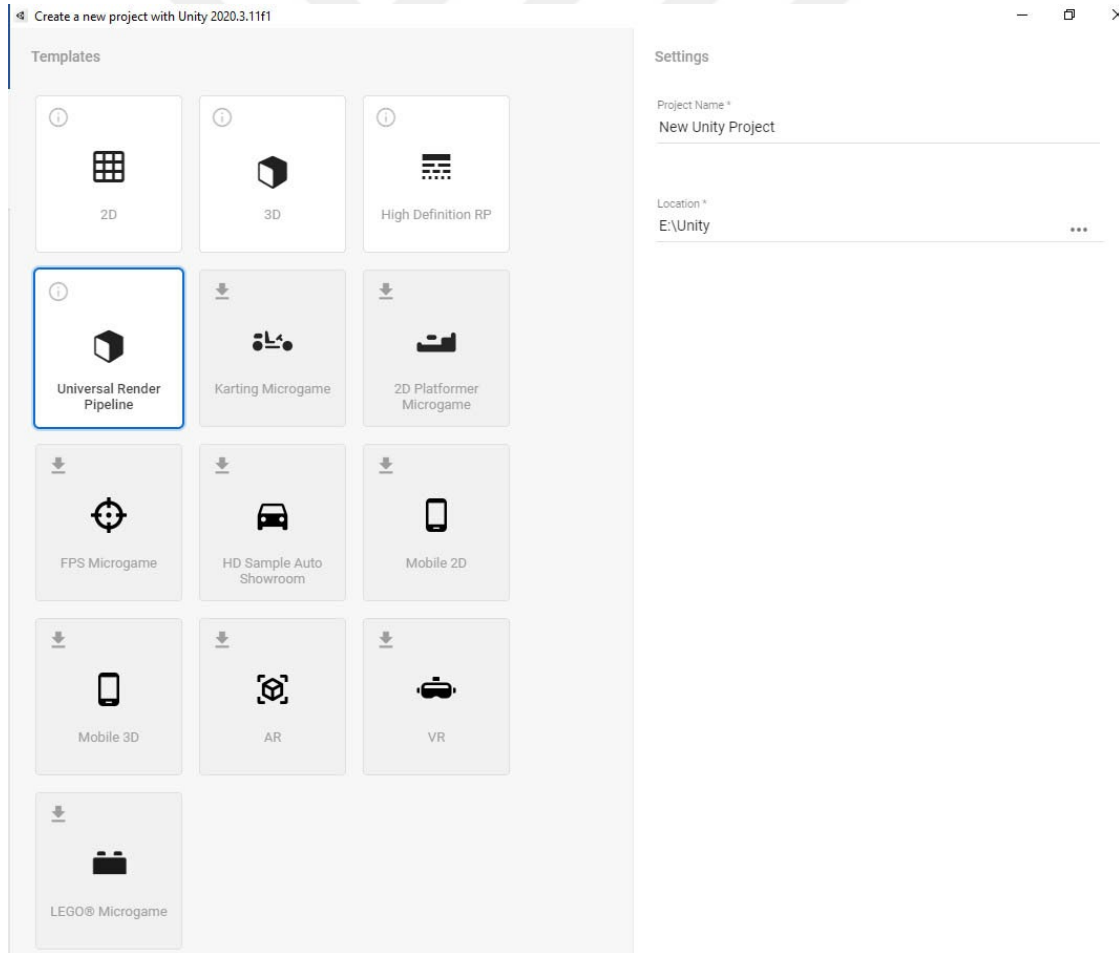
Şekil 1.11 Örnek bir Unity Hub ekranı

Kullanıcılar, üzerinde çalıştıkları oyun motorlarının ilgili sürümünü güncellerken ya da çalışma dosyalarını bir sürümden başka bir sürüme geçirirken çok dikkatli olmalıdır. Unity firması çok sık olarak güncelleştirme yayınlamakta olup, bu güncelleştirmelerle birlikte bazı paketlerde değişiklikler olabilmektedir. Bu tür durumlarda, üzerinde çalıştığınız proje derleme hataları verebilir. Dolayısıyla farklı sürümlere geçişlerden

önce, sürüm ile birlikte gelen değişiklikler hakkında <https://unity3d.com/get-unity/download/archive> sitesinden bilgi alınmalıdır. Bu sitede Unity'nin geçmişten günümüze tüm sürümleri yer almaktadır. Tam olarak emin olunmayan durumlarda mutlaka mevcut proje klasörü yedeklenmelidir. Unity yazılımının kurulumu hakkında daha ayrıntılı bilgi için Ek-1'e bakılabilir.

1.2.3.2. Unity Programı Hakkında

Unity, 2 ya da 3 boyutlu olarak farklı tasarım şablonları sunar. Tanıtımı yapılan çalışma "Universal Render Pipeline – URP" şablonu kullanılarak gerçekleştirilmiştir. Bu şablon da özünde 3D bir tasarım platformu olup, gelişmiş gölgelendirme tercihleri, farklı donanım platformları için zengin grafik seçenekleri sunar. Şekil 1.12'de çalışmanın yapıldığı bilgisayarda yüklü olan ve kurulum için bekleyen mevcut şablonlar görülmektedir.



Şekil 1.12 Unity tasarım şablonları

Her ne kadar VR ve AR uygulamaları için ayrı şablon kurulumları gözüксе de URP

şablonuyla da gelişmiş 3D VR ve AR uygulamaları tasarlanabilmektedir. Şablonlar kullanıcıya birtakım kütüphaneleri daha pratik şekilde kullanma fırsatı sağlar.

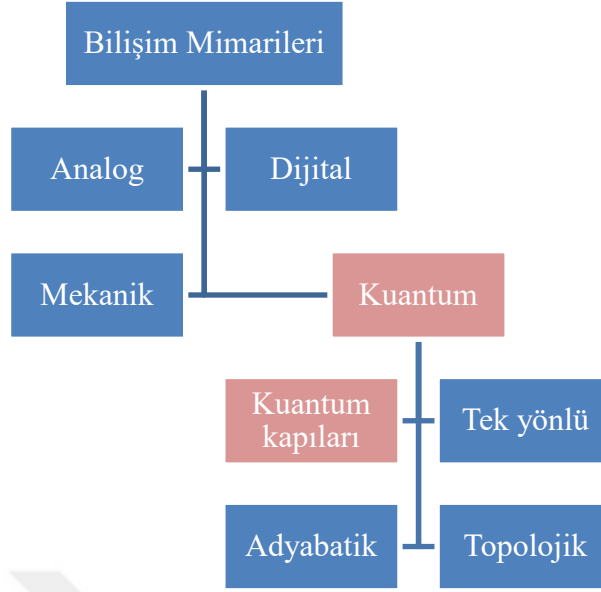
Unity bünyesinde gelişmiş bir betik dili desteğine sahiptir. Yaygın olarak tercih edilen dil C#'tır. Unity'de C# dilini kullanabilmek için Microsoft Visual Studio paket programının kurulumu gerekmektedir. Unity C# ile, standart yüksek seviyeli ve nesneye dayalı kodlama yanında, Unity'nin sahip olduğu tüm fizik motorlarını, vektör uzayı işlemlerini, mobil cihaz eklentilerini, sanal gerçeklik başlığı donanımlarını, hareket sensörlerini, dokunsal sensörleri ve diğer pek çok donanımı kullanmak mümkündür.

1.2.4. Kuantum Bilişimi ile İlgili Genel Bilgiler

Kuantum mekaniği ilkelerine sahip bilgisayarların kuantum teorisi ve tasarımı uzun bir geçmişe sahiptir. Doğayı simüle etme ilkesi (Feynman, 1982; Feynman, 1986) ve deterministik mantıkla çalışan klasik bilgisayar teknolojilerinin hız sınırı (Kanamori, 2006) araştırmacıları bu konuda çalışmalara yöneltmiştir.

Günümüzde kuantum bilgisayarların üretimi ve geliştirilmesi, IBM, D-wave sistemleri, Rigetti ve Bell laboratuvarları gibi çeşitli kurumlar tarafından hem bilimsel hem de ticari temelde gerçekleştirilmektedir. IBM, kuantum bilgisayarlarını araştırmacılar tarafından uzaktan erişimle programlamak için bulut teknolojisi aracılığıyla kullanılabilen web tabanlı bir besteleyici (composer) geliştirmiştir.

Genel olarak, mevcut bilgisayarlar zamanı veya rastgele durumları görselleştirmek için ayrık zamanlı olarak çalışır. Ayrık zamanlı sistemler, kuantum mekaniğini klasik bilgisayarlarla görselleştirmeyi imkânsız hale getirmektedir (Feynman, 1982). Bilişim mimarileri temel olarak Şekil 1.13'te (Padilha, 2014) gösterildiği gibi dört kategoride sınıflandırılır ve kuantum bilgisayarlar gerçek kuantum hesaplamayı mümkün kılar. Diğer üç yol “klasik” hesaplama olarak adlandırılır. Kuantum hesaplamanın da alt kategorileri vardır ve bu çalışma IBM'in kuantum kapıları modelini kullanan mimarisine dayanmaktadır. Ayrıca, Leap-2 adlı bir platform ile web sunucuları üzerinden programlanabilen ve adyabatik kuantum hesaplaması mimarisini (Ciocîrlan, 2014) kullanan D-Wave 2000Q kuantum bilgisayarı da bulunmaktadır.



Şekil 1.13 Bilişim mimarileri ağacı

Kuantum programlamasında sık kullanılan bazı kavramları ve terminolojiyi alt başlıklar halinde açıklamak konu bütünlüğü açısından yerinde olacaktır.

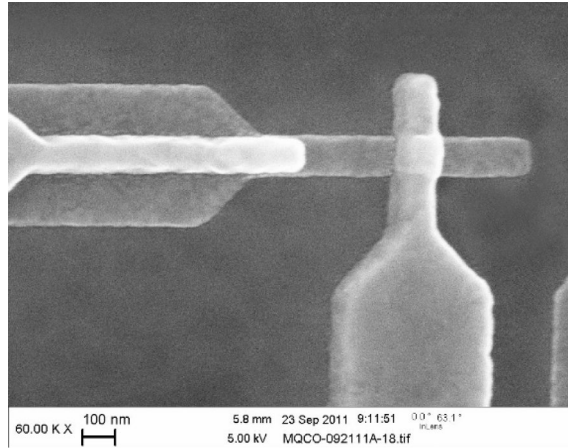
1.2.4.1. Kuantum Bilgisayarı

Atom ve atom altı seviyelerde madde ve enerjiyi açıklamaya çalışan kuantum teorisinin temel prensipleri üzerine geliştirilmiş, deterministik bilgisayar teknolojisinden farklı bir çalışma prensibine sahip bilgisayar mimarisidir. Bu bilgisayarlar çalışırken; kuantum bitleri (qubit), süperpozisyon ve dolanıklık (quantum entanglement) teorisi gibi kuantum fenomenlerini kullanırlar. Bu sayede kuantum bilgisayarlar, sıradan bilgisayarların tek başına gerçekleştiremeyeceği zorlu işlemleri kolaylıkla gerçekleştirebilir (Williams, 2011). Şekil 1.14'te IBM'in geliştirmiş olduğu süperiletken mimarisine dayalı kuantum bilgisayarlarının ilkesel bir modeli gösterilmiştir.



Şekil 1.14 Süperiletken kubitler üzerine inşa edilmiş bir IBM kuantum bilgisayar ilkesel modeli

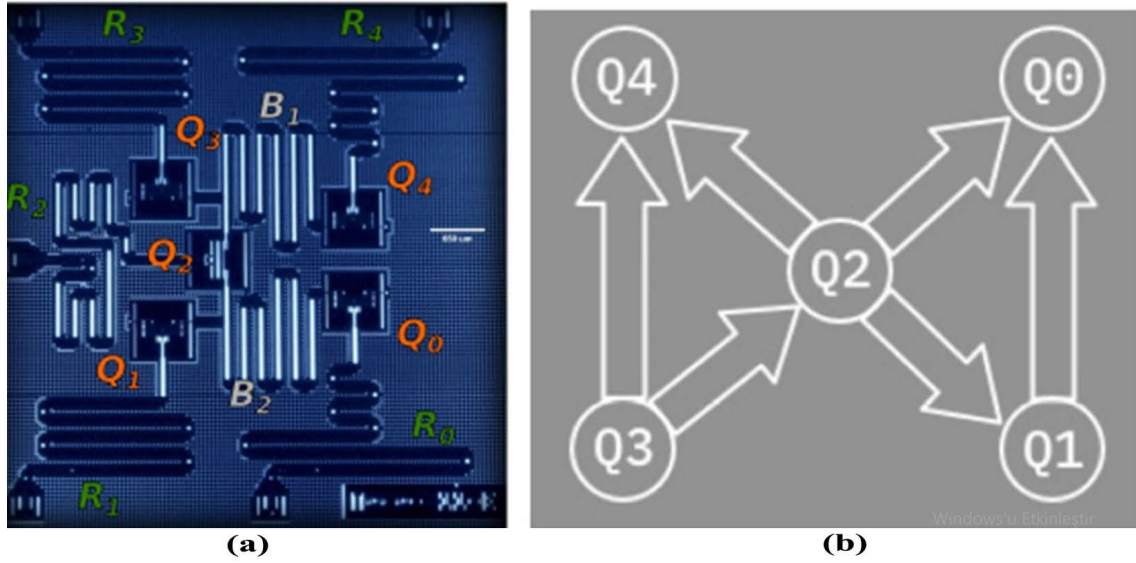
Bu tasarımda, süperiletkenlerin soğutulması için süper akışkanlar kullanılmaktadır. Süperiletkenler, mutlak sıfırın yaklaşık yüzde biri santigrat derecenin üzerinde, diğer bir ifadeyle fizik yasalarının izin verdiği teorik olarak en düşük sıcaklık seviyesinde soğutulur. Elektronlar soğutulmuş bu süper iletkenlerden geçirildiğinde, Cooper çiftleri adı verilen ve elektronların düşük sıcaklıkta birbirlerine bağlanmasıyla oluşan bir eşleşme gerçekleştirirler. Bu eşleşme Şekil 1.15'deki gibi bir çeşit süperiletken tünel bağlantısı olan Josephson bağlantısını oluşturur (<https://www.ibm.com/quantum-computing/what-is-quantum-computing/>).



Şekil 1.15 Süperiletken Josephson tünel bağlantısı

Çalışmada, IBM'in herkese açık kuantum bilgisayarları tercih edildi ve diğer kuantum işlemcilere ve arka uçlara (backends) "IBM Quantum Experience" platformu üzerinden ücretsiz olarak erişilebilmesine rağmen, Abhijith ve ark.nın da (2020) çalışmalarında kullandıkları 5 kubitlik bir bilgisayar olan ibmqx4 arka ucu kullanıldı. Bu işlemcilere, "IBM Quantum Experience" web sitesi (<https://quantum-computing.ibm.com/>) ziyaret edilerek erişilebilir.

Her kubit, müzik notasına benzer bir prensiple farklı bir frekanstadır ve çalışma frekansları sistemin tasarımcıları tarafından belirlenir. Şekil 1.16'da IBM'in 5-qubitlik Tenerife (ibmqx4 işlemcisi) makinesinin 5 süperiletken kubitinin bağlantı mimarisi ve dizilimi gösterilmiştir (<https://link.springer.com/article/10.1007/s11128-019-2229-2/figures/1>). Süperiletken kuantum bilişiminde, şarj gürültüsüne karşı azaltılmış hassasiyete sahip olacak şekilde tasarlanmış bu kubitler transmon kubitleri olarak adlandırılmaktadır.



Şekil 1.16 5-kubitlik ibmqx4 rezonatör mimarisi (a) ve kubit bağlantı yapısı (b)

Çip, 0.021 K sıcaklığında seyreltme buzdolabında tutulur ve her bir kubit arasındaki bağlantı düzlemsel dalga kılavuzu (CoPlanar Waveguide – CPW) rezonatörleri ile sağlanır (Behera ve ark., 2019). Rezonatör frekansları ve kubit çalışma frekansları Tablo 1.1'de gösterildiği gibidir (<https://github.com/Qiskit/ibmq-device-information/tree/master/backends/tenerife/V1>).

Tablo 1.1 5-kübitlik ibmqx4 işlemcisinin rezonatör ve kübit frekansları¹

Qubit	$\frac{\omega_i^R}{2\pi} (GHz)$	$\frac{\omega_i}{2\pi} (GHz)$
Q0	6.52396	5.2461
Q1	6.48078	5.3025
Q2	6.43875	5.3562
Q3	6.58036	5.4317
Q4	6.52698	5.1824

Her kübit belli bir CPW tarafından kontrol edilir ve okunur. CNOT (Controlled NOT) kapılarının eşleşme haritası, $Q1 \rightarrow [Q0]$, $Q2 \rightarrow [Q0, Q1, Q4]$, $Q3 \rightarrow [Q2, Q4]$ şeklindedir ve CNOT kapısının uygulanması için $a \rightarrow [b]$ formunda gösterilen eşleşmede, a kontrol kübiti ve b hedef kübitidir (Behera ve ark., 2019).

1.2.4.2. Kübit Kavramı ve Bloch Küresi ile Temsili

“Kuantum Biti”, Hilbert uzayı adı verilen ve sonlu veya sonsuz sayıda boyuta sahip karmaşık bir vektör uzayında temsil edilen iki boyutlu bir karmaşık vektördür. Vektörler genellikle $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ve $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ şeklinde gösterilir.

Kuantum bitlerinin gösteriminde Dirac ya da bir diğer adıyla “bra-ket” notasyonu çok sık kullanılır. Bu gösterimde $|0\rangle$ ve $|1\rangle$ kübitleri sırasıyla ket-0 ve ket-1 olarak telaffuz edilir. Ancak kuantum mekaniğinde bitlerin evresi belli değildir ve saf bir kübit süperpozisyon durumundadır. Süperpozisyon durumu ket-0 ve ket-1 kübitlerinin lineer bir kombinasyonu olarak ifade edilebilir. Bu durumun matematiksel gösterimi Denklem 1.1’deki gibidir.

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (1.1)$$

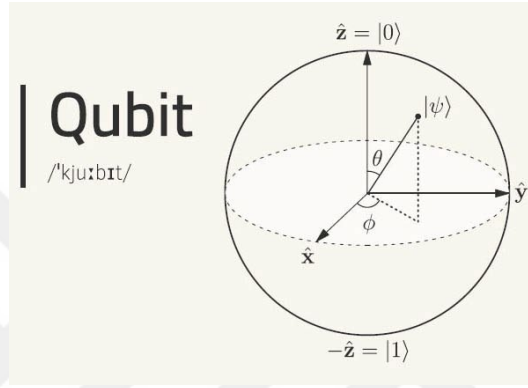
Burada, α ve β kompleks vektörlerdir ve sırasıyla $|0\rangle$ ve $|1\rangle$ kübitlerinin olasılık genliklerine karşılık gelir. Olasılık genliklerinin kompleks eşlenikleriyle çarpımlarının

¹ Çalışmanın yürütüldüğü Temmuz 2021 tarihi itibarıyla IBM firmasının kullanımda olan ve farklı kübitlerde 21 sistemi bulunmaktadır. Bunlardan yalnızca 5-kübitlik olan sistemler serbest erişime açıktır. Bunun yanında, yüksek kübitlik işlemler için serbest erişime açık simülörleri bulunmaktadır. Bilgisayarların işlemci mimarileri ve erişim durumları zamana göre değişiklik göstermektedir.

$(|c|^2 = c^*c)$ toplamı her zaman 1'e eşittir. Bu durum matematiksel olarak Denklem 1.2'de gösterildiği gibidir.

$$|\alpha|^2 + |\beta|^2 = 1 \quad (1.2)$$

Tüm kuantum durumlarını barındıran durum uzayı olan Hilbert uzayında kubitlerin daha iyi anlaşılması amacıyla Bloch küresi olarak adlandırılan bir küre notasyonu kullanılır. Şekil 1.17'de gösterilen 3 boyutlu küre notasyonunda θ açısı olasılık genliklerinin dağılımını, ϕ açısı ise bağıl fazı verir.



Şekil 1.17 Bloch küresi notasyonu

Bir kubitin Bloch küresiyle 3 boyutlu trigonometrik gösteriminde, süperpozisyon evresi için olasılık genlikleri olan α ve β Denklem 1.3'teki gibi ifade edilir.

$$\alpha = \cos\left(\frac{\theta}{2}\right), \beta = e^{i\phi} \sin\left(\frac{\theta}{2}\right) \quad (1.3)$$

Bir kuantum sistemini temsil eden Hilbert uzayı için en önemli koşullardan birisi bir vektörün, kendi eşleniğinin ters çaprazıyla (transpozitesiyle) çarpımının 1 olmasıdır ($\langle\psi|\psi\rangle = 1$). Burada $\langle\psi|$, $|\psi\rangle$ 'nin eşleniğinin ters çaprazıdır (transpozesidir) ve $|\psi\rangle^\dagger$ simgesiyle gösterilir. Bu, vektörün karesinin (iç çarpım tanımı gereği vektörün her bileşeninin karesi alınır ve toplanır) bire eşit olması gerektiğini belirten normalizasyon koşulu olarak adlandırılır.

Eğer $\theta = \pi/2$ ve $\phi = \pi/2$ olarak ayarlanırsa,

$$\cos\left(\frac{\pi}{4}\right) = \sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}}$$

Diğer bir ifadeyle $|\alpha|$ ve $|\beta|$ olasılıksal genlikler, $\frac{1}{\sqrt{2}}$ değerine sahip olur ve $(\frac{1}{\sqrt{2}})^2 +$

$(\frac{1}{\sqrt{2}})^2 = 1$ olarak elde edilir.

$$e^{i\frac{\pi}{2}} = i, (e^{i(-\frac{\pi}{2})} = -i)$$

Böylece, verilen açı değerlerinde bir kübit Y eksenini yönünde döndürülür ve $|\psi\rangle = \frac{|0\rangle + i|1\rangle}{\sqrt{2}}$ süperpozisyon değeri elde edilir.

Eğer, $\theta = \pi/2$ ve $\phi = 0$,

$$\cos\left(\frac{\pi}{4}\right) = \sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}}$$

$$e^{i0} = 1, (e^{i\pi} = -1)$$

Bir kübit X eksenini yönünde döndürülür ve $|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ süperpozisyon değeri elde edilir.

Kübitlerin bu şekilde kontrol edilmesi, “kuantum kapıları – quantum gates” denilen özel devrelerle gerçekleştirilir.

1.2.4.3. Kuantum Kapıları

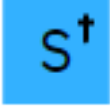
Klasik bir bilgisayara benzer şekilde, bir kuantum bilgisayarı, herhangi bir mantıksal kuantum hesaplamasını uygulayan evrensel kapılara sahiptir. İki kübitli kontrollü değil kapısı (CNOT) ve tek kübitli kapı kullanarak çok kübitli bir mantık kapısı oluşturmak mümkündür (Kanamori, 2006).

Kuantum bilgisayar kapıları çoğunlukla matematiksel olarak vektörler ve matrisler kullanılarak temsil edilir. Bir kübit hem bit hem de işaret verisine sahiptir. Bit çevirme, işaret kaydırma hem bit çevirme ve hem de işaret kaydırma işlemleri sırasıyla Pauli-X, Pauli-Z ve Pauli-Y kapıları ile yapılır.

IBM kuantum besteleycisi platformu tarafından yaygın olarak kullanılan kuantum kapıları, Tablo 1.2’de listelenmiştir.

Tablo 1.2 IBM kuantum besteleycisi üzerinde kullanılan kapılar

Kuantum kapıları	Besteleyici referans şekli	Operatör ismi ya da kısa açıklaması
H kapısı		Hadamard

CX kapısı		Kontrollü-X (CNOT kapısı)
Id kapısı		Birim matris
Rx kapısı		X eksenı boyunca dönüş
Ry kapısı		Y eksenı boyunca dönüş
Rz kapısı		Z eksenı boyunca dönüş
X kapısı		Pauli-X (değil kapısı)
Y kapısı		Pauli-Y
Z kapısı		Pauli-Z
S kapısı		Faz (Z eksenı boyunca $\pi/2$ 'lik açıyla dönüş)
Sdg kapısı		Ters faz (Z eksenı boyunca $-\pi/2$ 'lik açıyla dönüş)
T kapısı		Z eksenı boyunca $\pi/4$ 'lük açıyla dönüş
Tdg kapısı		Z eksenı boyunca $-\pi/4$ 'lük açıyla dönüş
CCX kapısı		İki kübitlik CNOT kapısının 3 kübitlik olanıdır. Toffoli kapısı olarak da bilinir.
SWAP kapısı		SWAP – Karşılıklı takas

Bu kapılardan bazılarının matris notasyonları ise Tablo 1.3’te gösterildiği gibidir.

Tablo 1.3 Bazı temel kuantum kapılarının matris gösterimleri

Operatör	Kapı	Matris
Pauli-X (X)		$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y (Y)		$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z (Z)		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Faz kapısı (S)		$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
T kapısı (T)		$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
Kontrollü Değil (Controlled Not – CNOT, CX)		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}_2$
SWAP		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Bu kapıların matris formasyonu ve IBM Kuantum Besteleyicisi gösterimleri aşağıdaki gibidir:

Pauli-X kapısı $= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \oplus$ ile temsil edilir.

² IBM kuantum besteleyicisinde kullanılan CNOT kapısı formu kullanılmıştır. Farklı kaynaklarda diğer formu tercih edilmektedir.

Pauli-Y kapısı $= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$  ile temsil edilir.

Pauli-Z kapısı $= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$  ile temsil edilir.

Bu kapılar aynı zamanda üniter (unitary) matris olarak da adlandırılır. Üniter matris bilindiği üzere kompleks eşleniğinin (conjugate) ters çaprazı ile çarpıldığında birim matris üreten matrislerdir. Diğer bir ifadeyle kompleks eşleniğinin ters çaprazı o matrisin tersine eşit olan matrislerdir. Bu durumu 2 x 2 kare matris üzerinden hatırlayacak olursak;

$$A = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \Rightarrow A^{-1} = \frac{1}{\det[A]} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$A^* = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$$

$$A^\dagger = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = A^{-1}$$

Bu sonucu veren matrisler, $AA^{-1} = I$ (birim matris) $= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ sonucunu sağlar. Bu kapılar Hermitian matris özelliğine de sahiptirler. Kompleks eşleniğinin ters çaprazı kendisine eşit olan matrislerdir. Ölçülebilir matrislerdir ve değerleri beklenen değer olarak adlandırılır. Verilen bir Hilbert uzayı evresi için, L'nin gözlemlenebilir bir değer olduğu varsayılırsa, L değeri şu şekilde bulunur:

$$\langle L \rangle = \langle \psi | L | \psi \rangle \quad (1.4)$$

Bir kübit, 2 x 2 üniter X kapısıyla gösterildiği gibi döndürülebilir:

$$\bigoplus |1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0x0 & 1x1 \\ 1x0 & 0x1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

Görüldüğü üzere, $|1\rangle$ kübit evresi $|0\rangle$ kübit evresine döndürülmüştür.

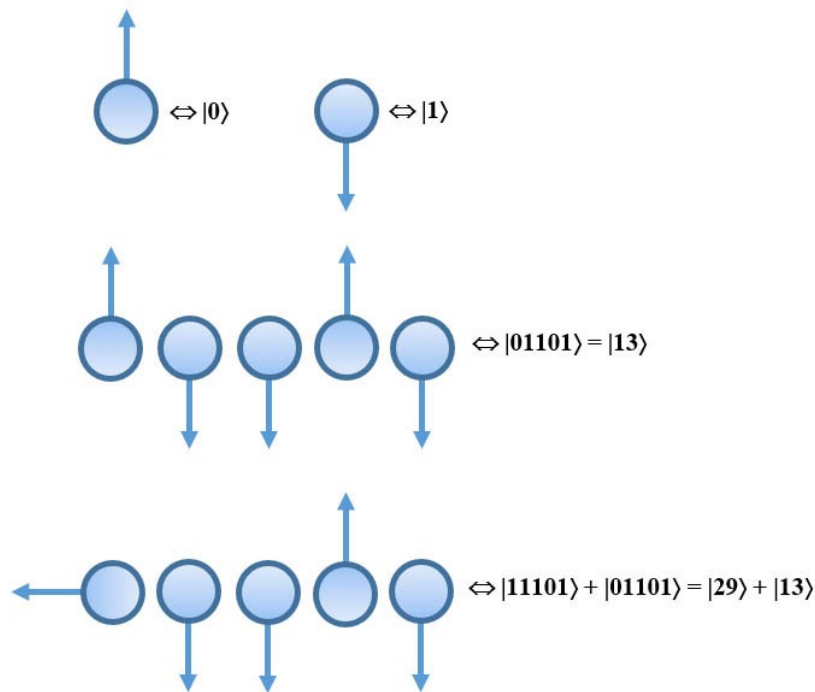
1.2.4.4. Kuantum Bilişimi ve Kuantum Kapılarında İşlemler

Kuantum bilişiminin en önemli iki bileşeni süperpozisyon ve dolanıklıktır (entanglement). Süperpozisyon, kübitlerin aynı anda $|0\rangle$ ya da $|1\rangle$ evresinde olması durumunu ifade eder. Kübitlerin durumu gözlemlendiğinde belli bir olasılıkla 0 ya da 1 durumundadırlar. Bu durum ölçme işlemi (measurement) gerçekleştirilene kadar devam eder. Ölçme işlemi gerçekleştirildiğinde artık süperpozisyon evresi daha fazla devam

etmez ve bu koherent süperpozisyon durumu süperpozisyon olmayan belli bir evreye düşer (Hughes ve ark., 2021). Diğer bir ifadeyle, ölçme işlemi sonucunda klasik dijital bilgisayarlarda olduğu gibi belli bir değer elde edilir.

Bu doğal fenomen kubitlerin klasik dijital bilgisayarlarda olduğundan farklı olarak daha fazla bilgi tutabilmesini sağlar. N kubitlik bir kuantum bilgisayarı 2^n kadar keyfi bir evrede olabilir. Klasik bir bilgisayarda ise herhangi bir zamanda bu 2^n evreden yalnızca bir tanesi içinde yer alınabilir. Kuantum mekaniğinin deterministik bilgisayarlar yoluyla simüle edilmesi zorluğu da buradan ileri gelmektedir.

Şekil 1.18’de temsili olarak 5-kubitlik bir kuantum bilgisayarında Q4 kubitinin süperpozisyon evresinde olması durumunda muhtemel ölçülebilecek değerler gösterilmiştir. Aynı anda 29 ve 13 bilgilerinin süperpozisyon halinde olduğu görülmektedir. Bu olasılıksal davranışın kontrol edilmesi ve ihtiyaç duyulan bilginin elde edilmesi çok yüksek bir işlem gücü gerektirir. Feynman (1982); “Bir kuantum davranışının klasik bilgisayarlarla simüle ettirilmesi üstel olarak genişleyen bir zaman gerektirecektir. Ancak evrensel bir kuantum bilgisayarının tasarımıyla gerçek anlamda kuantum davranışlarını modellemek ve çözümlemek mümkün olabilecektir” demiştir.



Şekil 1.18 Süperpozisyon evresinin temsili gösterimi

Hadamard kapısı ($\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$) $|0\rangle$ ve $|1\rangle$ evrelerini sırasıyla $|+\rangle$ ve $|-\rangle$ evrelerine döndürür. Kısacası kubitleri süperpozisyon evresine dönüştürmek için kullanılır.

$$\text{Hadamard kapısı} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Örneğin, eğer $|0\rangle$ kubit evresi üzerine Hadamard kapısı işlemini uygularsak, aşağıdaki gibi süperpozisyon evresi elde edilir:

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 - 1 \cdot 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} =$$

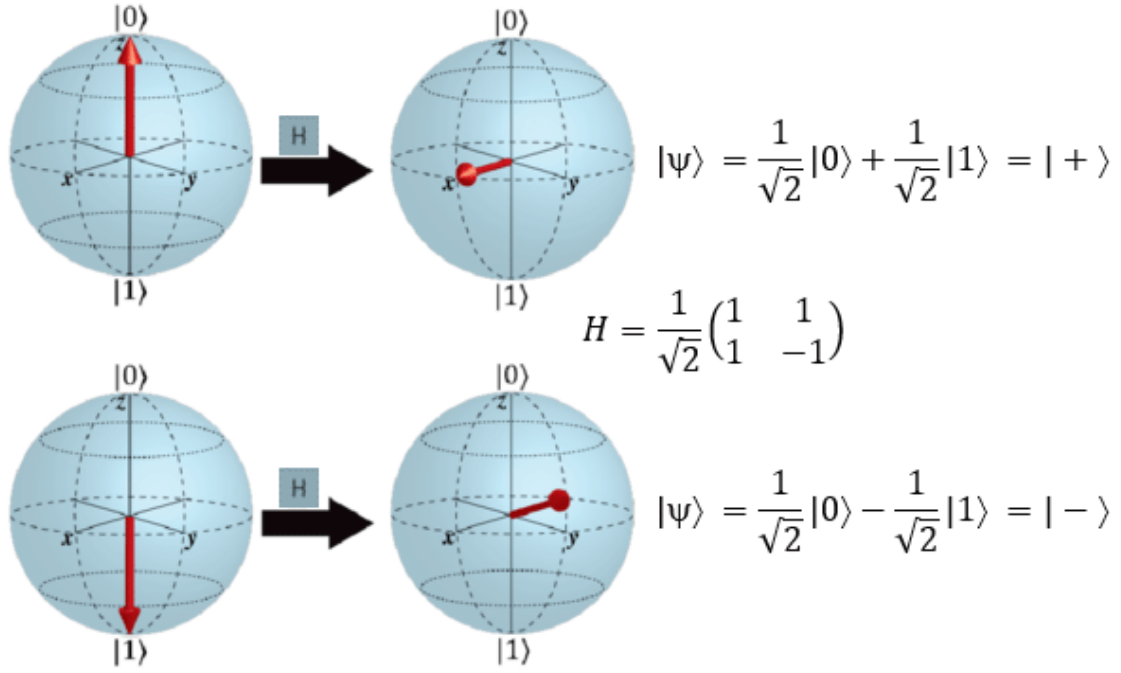
$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle = |+\rangle \text{ (Pozitif x eksenine doğrultusunda bir vektör)}$$

Hadamard kapısını $|1\rangle$ kubit evresi üzerine uygularsak, aşağıdaki gibi süperpozisyon evresi elde edilir:

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} |1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 0 - 1 \cdot 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} =$$

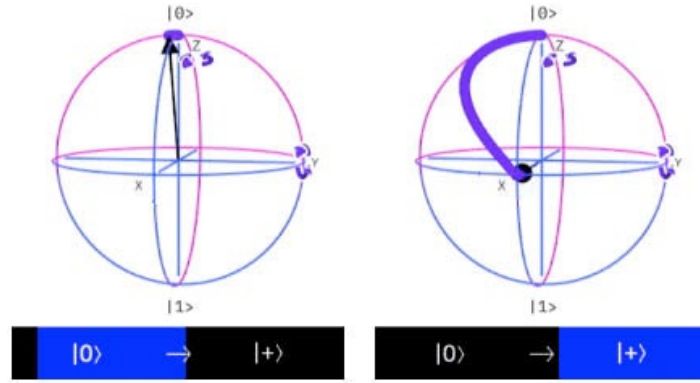
$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle = |-\rangle \text{ (Negatif x eksenine doğrultusunda bir vektör)}$$

Her iki işlemin Bloch küresi üzerindeki temsili Şekil 1.19'da gösterildiği gibidir (Moreno-Pineda ve ark., 2017).



Şekil 1.19 Tek kubitlik Hadamard kapısının etkisinin Bloch küresinde gösterimi

Hadamard kapısı eşdeğer olarak, Z eksenini etrafında $\phi = \pi$ kadarlık ve Y eksenini etrafında $\theta = \pi/2$ kadarlık iki rotasyonun birleşimidir. Bu durum Şekil 1.19’da görüldüğü gibidir.



Şekil 1.20 trans dönüşümü

Kuantum bilişiminde bir diğer önemli fenomen ise, birden fazla kubitin birbiriyle dolanıklık halinde olması durumudur. Dolanıklık, genellikle atom altı parçacıklar arasındaki doğrudan etkileşimle meydana gelen fiziksel bir olgudur (Ladd ve ark., 2010). Dolanıklık halindeki parçacıklar üzerinde yapılan konum, momentum, dönme, polarizasyon gibi fiziksel özelliklerin ölçümlerinin ilişkili olduğu gözlemlenmiştir (Hughes ve ark., 2021). Örneğin, toplam dönüşlerinin sıfır olduğu bilinecek şekilde

üretilen bir çift parçacığı düşünün. Parçacıklardan birinin belirli bir eksen boyunca saat yönünde döndüğü belirlenirse, aralarındaki dolanıklık nedeniyle diğer parçacığın beklendiği gibi saat yönünün tersine döndüğü tespit edilecektir (Preskill, 2012).

Kübitler ile klasik bitler arasındaki önemli bir ayırt edici özellik, birden fazla kübitin kuantum dolanıklığı sergileyebilmesidir. Kuantum dolanıklığını gösteren en basit sistem, iki kübitlik sistemdir. Dolanıklık durumunu elde etmek için CNOT kapısı kullanılır. CNOT kapısı kuantum bilişiminde kullanılan en önemli çok kübitli kapıdır (Hughes ve ark., 2021). Nasıl ki klasik dijital sistemlerdeki tümleşik devrelerde çoğu mantık işlemi NAND ve NOR kapılarıyla gerçekleştiriliyorsa, benzer durum CNOT kapılarıyla kuantum bilgisayarları için geçerlidir.

İki kübitli bir sistemde kübitlerin karmaşık genliklerini göstermek için 4 boyutlu bir vektör matrisi kullanabiliriz.

$$|a\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle = \begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}$$

Tek kübitlik işlemlerde olduğu gibi kompleks genliklerin eşlenikleriyle çarpımlarının karelerinin toplamı, diğer bir ifadeyle tüm evrelerin muhtemel olasılıklarının toplamı 1 olur.

$$|a_{00}|^2 + |a_{01}|^2 + |a_{10}|^2 + |a_{11}|^2 = 1 \quad (1.5)$$

Burada her bir evrenin 4 boyutlu vektör matrisi ile temsili şu şekilde gösterilir:

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

CNOT kapsının iki kübitin dolanıklık durumuna getirilmesinde kullanılmasına en iyi örnek pozitif Bell evresidir ($|\Phi^+\rangle$). Bell evresini elde etmeden önce kübitlerin pozisyona hazır hale getirilmesi için ilk kübitin Hadamard kapısıyla süperpozisyona getirilmesi gerekir.

$H|0\rangle = |+\rangle$, ardından ikinci kübitle bu evrenin tensör çarpımı alınır.

$$|0\rangle \otimes |+\rangle = |0+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ 0 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} (|00\rangle + |01\rangle)$$

IBM kuantum besteleycisi platformunda tercih edilen CNOT kapısı formuna göre, q0 kontrol kütübi, q1 ise hedef kütübidir. Bu durumda eğer kontrol kütübi 1 ise hedef kütübin evresi değiştirilir. Bir önceki işlemde elde edilen evreye CNOT kapısı uygulandığında yeni durum şu şekilde gerçekleşir:

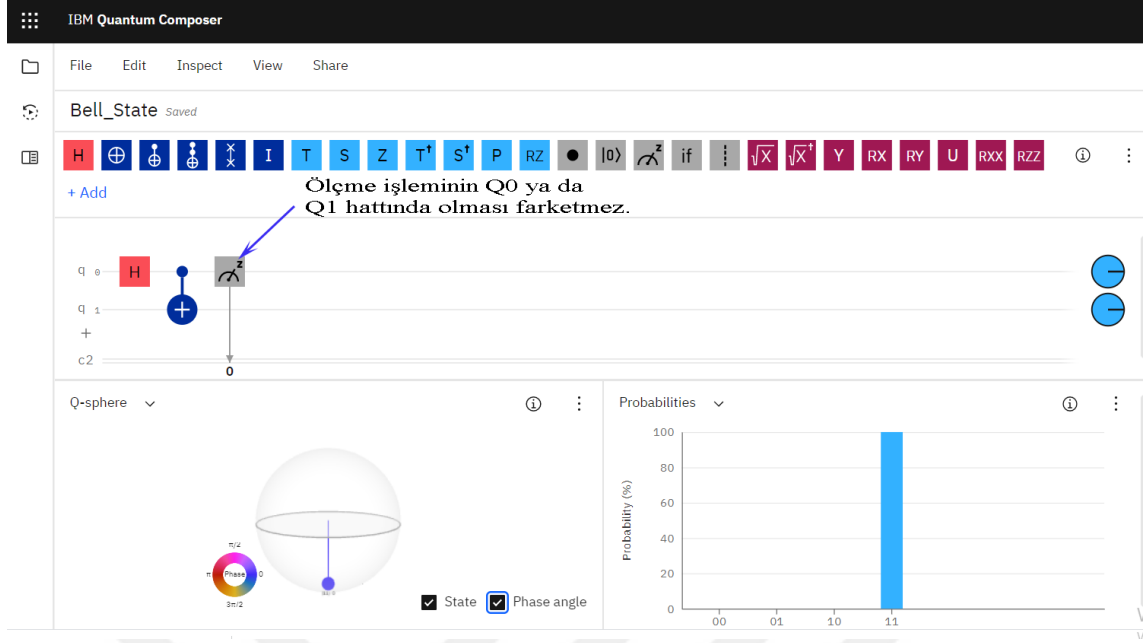
$$\text{CNOT}|+0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Bu bir Bell durumu olarak bilinir. Bu durumun $|00\rangle$ durumunda ölçülme olasılığının %50 ve $|11\rangle$ durumunda ölçülme olasılığının %50 olduğunu görebiliriz. En ilginç olanı, $|01\rangle$ veya $|10\rangle$ durumlarında ölçülme şansının %0 olmasıdır.

Bu süperpozisyon durum, iki ayrı kütibit durumu olarak yazılamaz. Kütibitlerimiz süperpozisyonda olsa da birini ölçmek bize diğierinin durumunu söyleyecek ve sistemin süperpozisyonunu çökertecektir. Örneğin, ilk kütibiti ölçecek ve $|1\rangle$ evresini elde edecek olursak, kütibitlerimizin kolektif evresi şu şekilde olacaktır:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \xrightarrow{\text{ölçme işlemi}} |11\rangle$$

Bu kütibitleri birbirlerinden herhangi bir mesafede ayırsak bile, bir kütibiti ölçtüğümüz anda süperpozisyon durumu çökecektir. Bu işlemin IBM kuantum besteleycisinde gerçekleştirildiği devre Şekil 1.21’de gösterildiği gibidir.



Şekil 1.21 Bell evresini gerçekleştiren devre

1.2.4.5. Kuantum Bulut Programlama Platformları

Bulut programlama, internet üzerinden birçok kullanıcının kullanabileceği veri merkezlerini tanımlayan bir bulut bilişim ürünüdür (Sharma ve ark., 2014). Bulut tabanlı kuantum hesaplama, şirketlerin ve araştırmacıların bulut ağları ve web tarayıcıları aracılığıyla kuantum algoritmalarını test etmelerini sağlayan yeni bir teknolojidir. Fiziksel kuantum bilgisayarların geliştirilmesi ve uygulanması, özellikle maliyetler açısından zorlu bir süreçtir. Kuantum bilgisayarların pratik kullanımındaki zorluklar, IBM, Microsoft, Google, Amazon, D-wave ve Rigetti gibi yüksek teknoloji şirketlerinin bu bilgisayarlara bulut mimarisi üzerinden erişim alternatifi sunmalarının yolunu açmıştır (Soeparno ve Perbangsa, 2021).

Yapılan literatür taraması bulgularına göre, IBM firmasının Quantum Experience olarak adlandırdığı kuantum bilişimi bulut ağını 2016 yılıyla birlikte kullanıma sunan ilk firma olduğu saptanmıştır.

IBM gibi bu şirketlerden bazıları, kullanıcıların fiziksel kuantum bilgisayarlarına serbestçe erişmesine izin verirken, diğerleri henüz vermemektedir. Bu yüksek teknoloji şirketlerinin süper bilgisayarlarıyla kuantum algoritmalarını bulut platformu üzerinden simüle etmek başka bir alternatif olarak karşımıza çıkmaktadır.

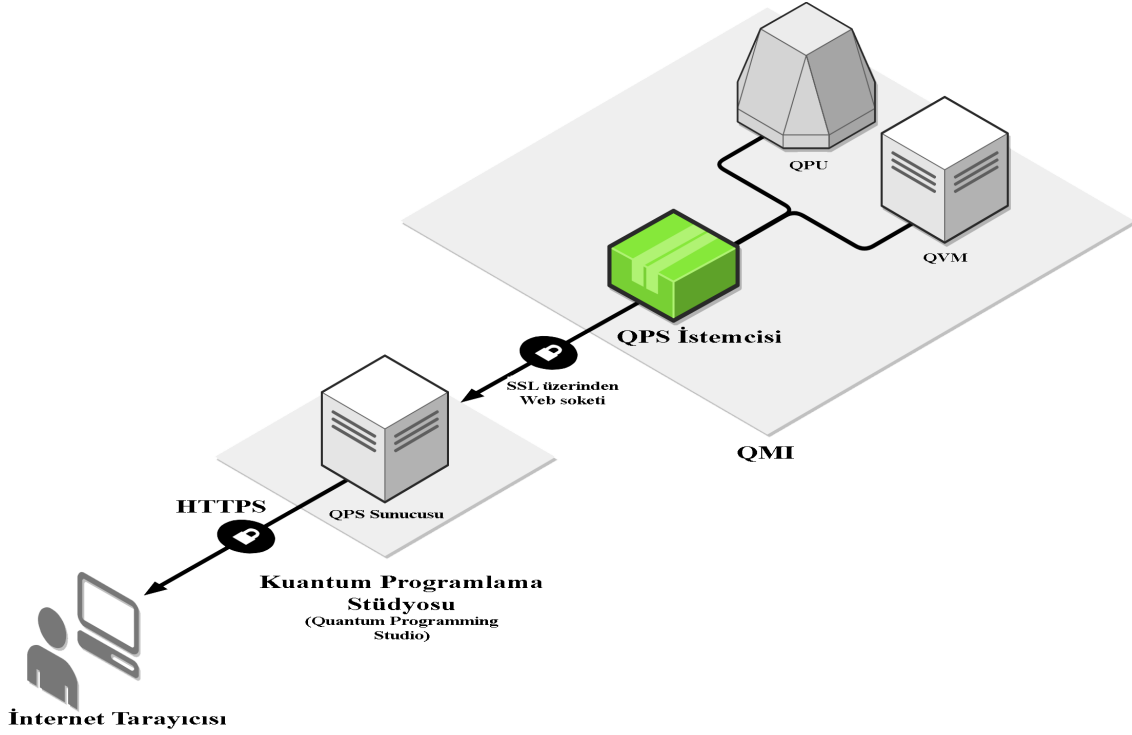
Son yıllarda çeşitli araştırmacılar, bulut bilişimi ağı üzerinden bu bilgisayarların

performanslarına dair gerçekleştirdikleri testlerin sonuçlarını yayınlamaktadırlar. Soeparno ve Perbangsa (2021), IBM bulut bilişimi (<https://www.ibm.com/quantum-computing/>) ve Quantum Inspire (<https://www.quantum-inspire.com/>) bulut bilişimi performanslarını karşılaştırmalı olarak test etmişlerdir. Başka bir araştırmada (Genç ve ark., 2020), klasik bir bilgisayarda pi sayısı algoritmasının ve IBM fiziksel kuantum arka ucu ve simülatör arka ucunda pi sayısı kuantum algoritmasının test edilmesi gerçekleştirilmiştir.

HQC (Hybrid Quantum-Classical) algoritmaları olarak bilinen bir dizi kuantum algoritması, kuantum makine öğrenimi ve kuantum kimyası gibi uygulamalar için araştırmacılar tarafından bulut bilişim ağı üzerinden test edilmektedir. “Uyarlanabilir hibrit kuantum-klasik” (AHQC) algoritmalarını tasarlamak ve test etmek için çok uygun olan algo2qpu adlı üst düzey bir sistem, bulut tabanlı “Rigetti Computing's Forest” üzerinde gerçekleştirilmiştir (Sim ve ark., 2018).

Son yıllarda kuantum bilgisayarlarının programlanmasıyla ilgili bir dizi çalışma yapılmakta olup, bilindik bir dizi kuantum algoritmasını, kuantum mekaniği ve davranışı üzerine bilgisi olmayan programcıların da kullanarak kuantum bilgisayarlarını programlayabilmeleri için daha anlaşılır notasyona sahip diller geliştirilmektedir. Bichsel ve ark.'nın (2020) tanıtımını yapmış oldukları “Silq” dili de bunlardan biri olup, Grover’ın meşhur kuantum arama algoritması üzerinden sunumunu gerçekleştirmişlerdir.

Farklı web tabanlı ve bulut tabanlı kuantum programlama mimarileri vardır. Bunların en bilinenleri Rigetti QCS, Rigetti Forrest SDK ve IBM Qiskit'tir. Web tabanlı kuantum programlamanın temel prensibi Şekil 1.22’de gösterilmektedir (https://quantum-circuit.com/docs/qps_client#how-it-works).



Şekil 1.22 Web tabanlı kuantum programlamanın ilkesel işleyişi

Sistemin çalışma ilkesi kısaca şu şekilde açıklanabilir (<https://research.aimultiple.com/quantum-computing-cloud/>):

- Kuantum Programlama Stüdyosunun (Quantum Programming Studio) istemci uygulaması, yerel makinede veya web tarayıcısı üzerinden bulutta çalışır.
- Kullanıcı, Kuantum Makinesi İmajına (Quantum Machine Image – QMI) güvenli soket katmanı üzerinden bağlanır.
- QMI, pyQuil ya da Qiskit gibi araçları kullanarak kuantum yazılım uygulamalarını çalıştırır.
- Gerçekleştirilen kod kuantum sanal makinesinde (Quantum Virtual Machine – QVM) çalıştırılır.
- QVM kuantum işlemcilerinde çalışacak bir dalga formu üretir.
- QMI bu dalga formlarını, kuantum işlemcisi ünitesine (Quantum Processing Unit – QPU) gönderir ve yine bu işlemciden alır.
- QPU çözüm setinden gerekli bilgiyi gönderir ve QMI bu bilgiyi işleyerek klasik bilgisayara geri gönderir.

Alandaki gelişmeler, bulut bilişim ağları üzerinden kullanılan kuantum teknolojilerinin

katlanarak artacağını ve çok daha fazla kullanılacağını göstermektedir.

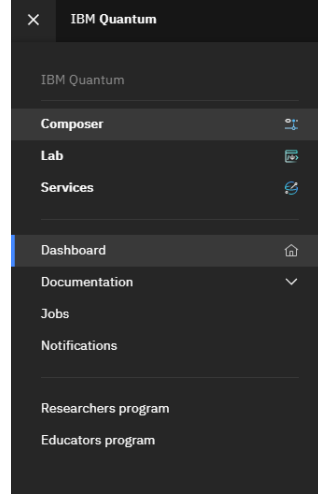
1.2.4.6. IBM Kuantum Besteleyicisi

Kuantum besteleyicisi (Quantum Composer), bir kuantum işlemcisinin programlanması için web üzerinden erişilebilen grafik kullanıcı arayüzüdür. Kübitlerin evrelerini kontrol eden iyi tanımlanmış kapı ve operatör kütüphanesiyle kuantum algoritmaları tasarlamayı sağlayan bir araçtır. Görüntüsü itibariyle müzikal beste yapmayı sağlayan müzik notası dizgi arayüzünü anımsatmaktadır. Bu araca alternatif olarak QASM satır kodlama editörüyle de aynı işlemler yapılabilir.

Bu çalışmada IBM'in kuantum besteleyicisi platformu örnek olarak alınmıştır. IBM'in kuantum besteleyicisi platformu, sürükle ve bırak temelli web tabanlı bir programlama ortamıdır.

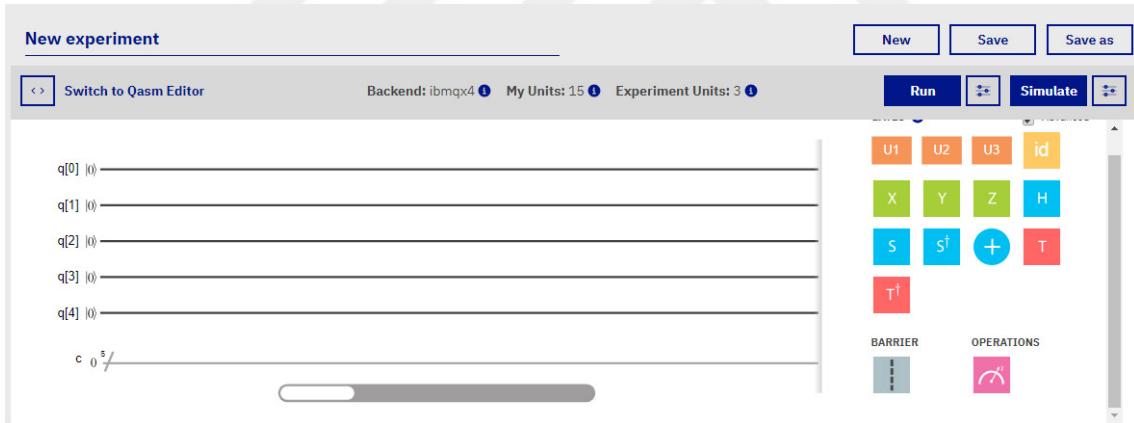
IBM ayrıca Qiskit adlı açık kaynaklı bir bulut programlama ortamı da sağlar. Kuantum bilgisayarlarını programlamak için açık kaynaklı bir sistemdir. Kurulum gerekmez ve yalnızca gerçek bir kuantum donanımı arka uçlarında veya kuantum simülatörlerinde kod çalıştırırsınız. Qiskit, kuantum devrelerini işletmek için jupyter notebook tarafından desteklenen bir Python kitaplığı olarak görülebilir (Genç ve ark., 2020).

Qiskit IBM kütüphaneleri yerel bilgisayara da kurulabilir ve Apache web sunucusu aracılığıyla yerel bilgisayardan IBM cihazlarına iş göndermek mümkündür. JSON formatında veri gönderip alarak Jupyter Notebook üzerinden işletilebilir. Bir kullanıcı hesabı oluşturulduktan sonra, Şekil 1.23'te gösterildiği gibi, biri "Composer" ve diğeri "Lab" olmak üzere iki tasarım alternatifi vardır.

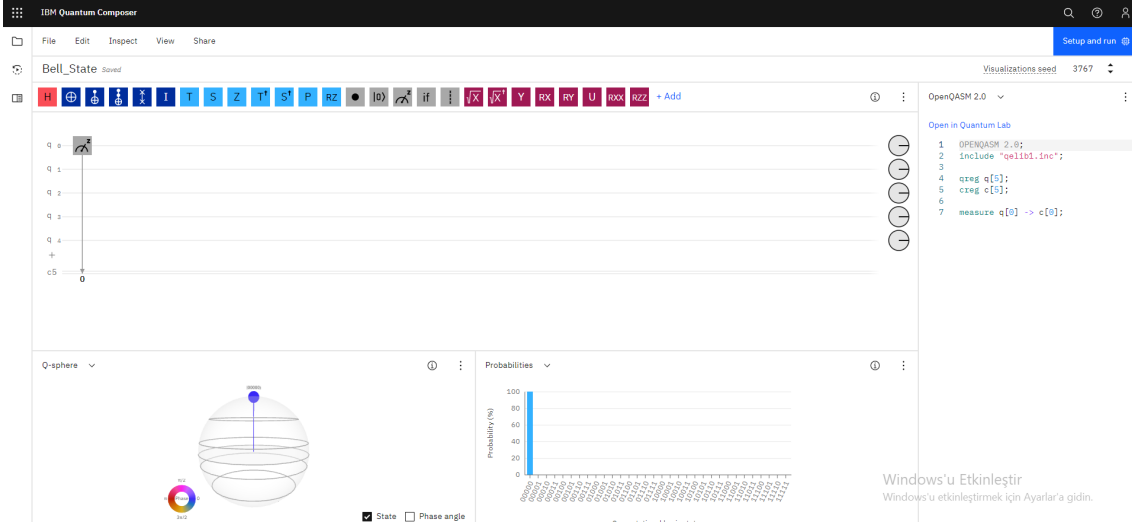


Şekil 1.23 Kuantum bilgisayar programlama tercih menüsü

“Composer” seçeneğini seçerseniz boş bir besteci editör ekranı gelir. Şekil 1.24’te ve Şekil 1.25’te sırasıyla farklı tarihlerde elde edilmiş besteleyici ekranı çıktıları görülmektedir. Ekran çıktılarından da görüleceği üzere kullanıcı arayüzü değişmekte ve yeni özellikler eklenmektedir.



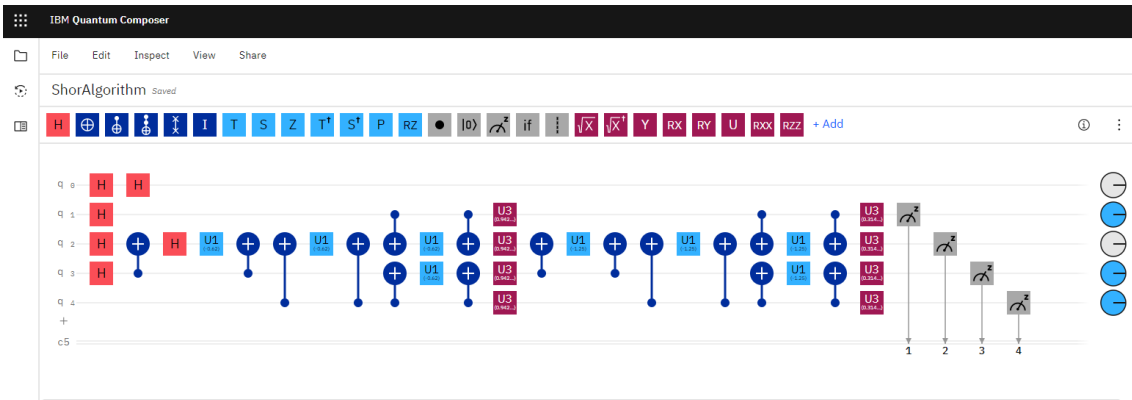
Şekil 1.24 İlk nesil IBM kuantum besteleyicisinin görüntüsü



Şekil 1.25 Temmuz 2021 tarihi itibarıyla güncel kuantum besteleyicisi arayüzü

IBM kuantum besteleyicisi platformu zaman içinde bir dizi değişikliğe uğramıştır. IBM kuantum besteleyicisinde her bir kübit satırı için kubitler iyi tanımlanmış bir aşamada hazır hale getirilerek bir kuantum devresi başlatılır. Bu noktada, tüm kübitler, IBM kuantum besteleyicisi tarafından kullanıcılar için $|0\rangle$ olarak yapılandırılır.

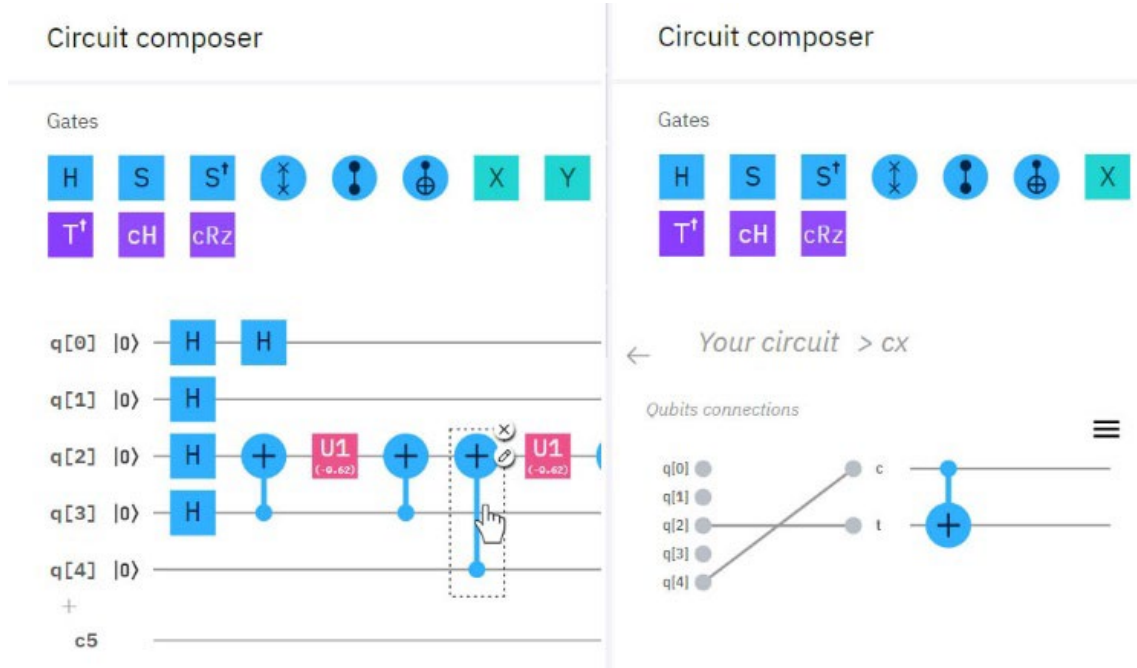
Şekil 1.26'da, “Kuantum Yaklaşık Optimizasyon Algoritması” ile ilgili örnek bir çalışma gösterilmiştir (Ab-hijith ve ark., 2020). Menüden kuantum kapılarını seçebilir ve seçilen kapıyı ilgili kübit satırına sürükleyebilirsiniz. Kuantum kapıları, farklı renk ve sembollerde kare kutular olarak temsil edilmektedir. Bunlar farklı zamanlar, genlikler ve fazlar için bir frekans çalıştırır.



Şekil 1.26 Örnek bir IBM Quantum Experience çalışması

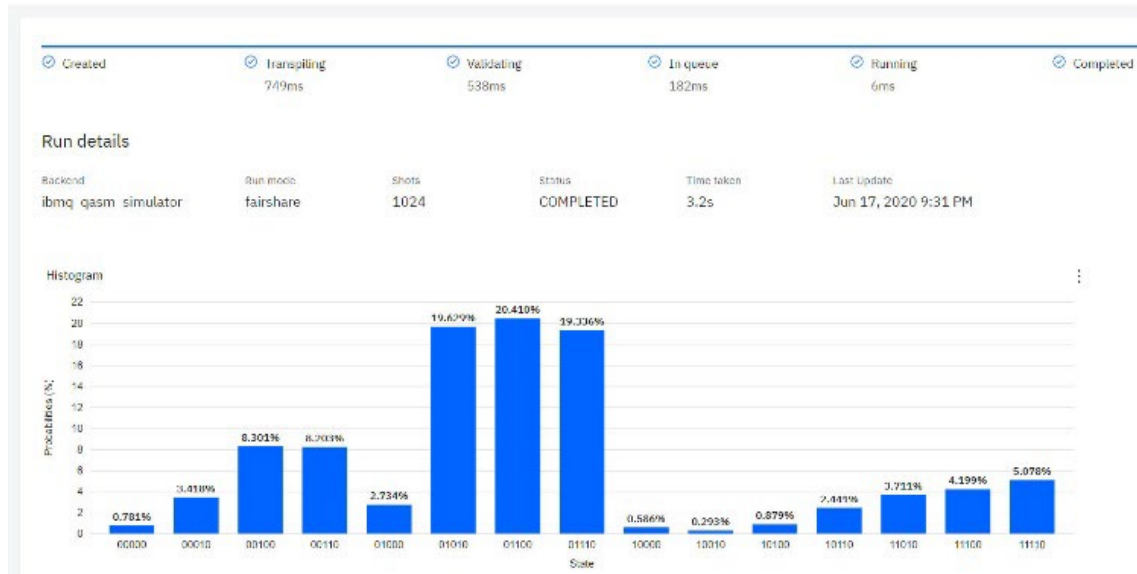
Devre besteleyicisinde ilgili kapı sembolü üzerine fare ile gelip tıkladığınızda, Şekil 1.27'deki gibi düzenleme seçeneği belirir. Kalem ikonuna tıkladıktan sonra karşımıza

kalem ikonu düzenleme arayüzü gelir ve kapı bağlantılarını düzenleyebilirsiniz.



Şekil 1.27 Devre besteleyicisi üzerinde düzenleme işlemi gerçekleştirme

Tasarımınızı bitirdikten sonra programı RUN komutu ile çalıştırırsınız. Bir süre sonra Şekil 1.28'de gösterildiği gibi her bir kübit evresinin olasılık dağılımını gösteren histogram tablosu şeklinde bir sonuç raporu alırsınız.



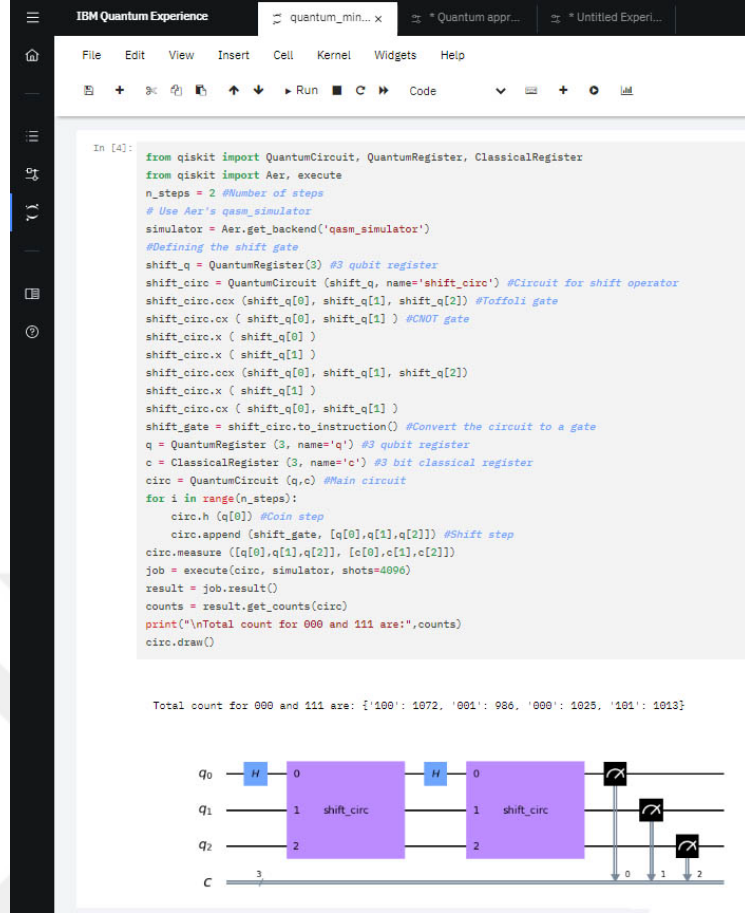
Şekil 1.28 Tasarım için histogram çıktısı

Şekilde görüldüğü gibi, her dağılımdaki tüm $q[0]$ kubitleri 0'dır. Çünkü yalnızca 1, 2, 3 ve 4 kubitlerinden ölçüm alınmıştır. Elde edilen ağırlıklı kubit durumları beklenen değerleri göstermektedir. Diğerleri ise çeşitli faktörlerin neden olduğu ölçüm hatalarıdır. "Time taken" işin toplam işlem süresini gösterir.

Kuantum besteleyicisi platformu ayrıca OpenQASM adında bir kod düzenleyiciye sahiptir. İsteğe bağlı olarak bu düzenleyiciyi kullanarak semboller yerine kubitlerin ve geçitlerin köşeli parantez gösterimlerini kullanarak programlama yapabilirsiniz.

Bir başka bulut tabanlı kuantum bilgisayar programlama platformu, Qiskit Notebook'tur. Dilerseniz IBM kuantum bilgisayarlarını jupyter notebook ile herhangi bir kurulum yapmadan, bulut tabanlı veya Anaconda paketini bilgisayarınıza kurarak programlayabilirsiniz. Qiskit platformu üzerinden sisteme iş gönderebilmeniz ve sonuçları almanız kullanıcı hesabınıza tanımlı olarak oluşturulmuş olan API Token anahtar değeriyle sağlanır.

Qiskit notebook, Jupyter notebook tarafından desteklenmektedir ve bu platform web tarayıcılarında çalışmaktadır. Bu web tabanlı platform, programlama dili olarak Python'ı kullanır. Örnek olarak, "Quantum Minimal Spanning Tree" algoritmasının uygulaması Şekil 1.29'da gösterilmiştir.



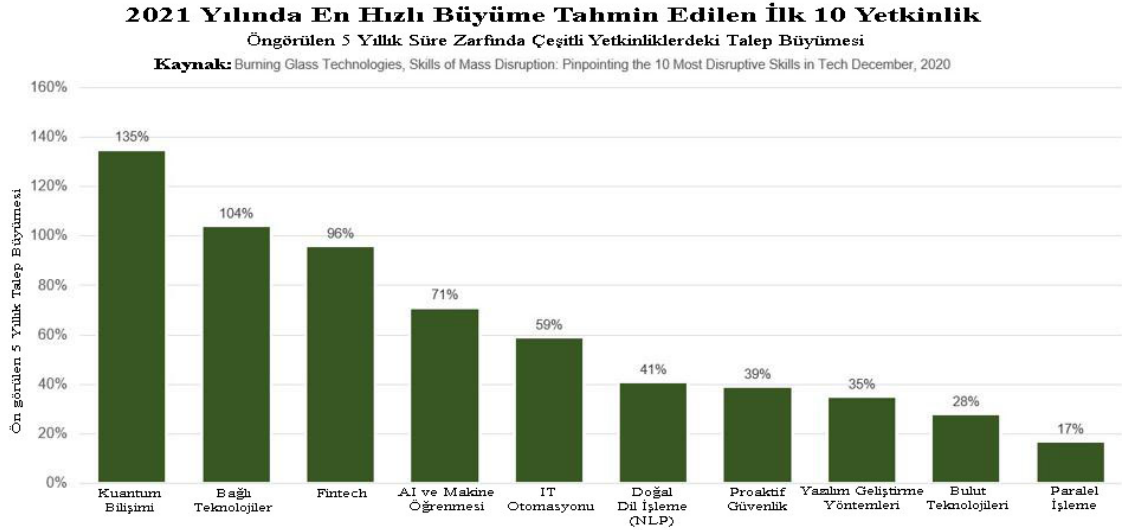
Şekil 1.29 Jupyter Notebook ile IBM kuantum bilgisayar programlaması

Bu arabirimin sağlayıcı, arka uç ve iş olmak üzere üç bölümü vardır. “Sağlayıcı” kısmı arka uçlara (backends) erişir ve arka uç nesnelerini sağlar. Kuantum devresini çalıştıran kısım “arka uç”tur. “İş” bölümü, gönderilen işin kaydını tutar. Günümüzün bulut tabanlı akıllı yapıları da bu ve benzeri altyapıları kullanmaktadır.

2. MATERYAL ve YÖNTEM

Bilgi ve iletişim teknolojilerinin önemli bir bileşeni olan bilgisayar teknolojilerinde 20. yy. süresince çok ciddi gelişmeler yaşanmıştır. İlk elektromekanik röleli bilgisayarlardan, transistörlü bilgisayarlara ve hemen ardından da entegre devre temelli bilgisayarlara giden süreçte bilgisayarların işlem gücü ve bellek kapasitesi katlanarak artmıştır. Çalışmanın başında da belirtilen birtakım sınırlılıkların ortaya çıkmasıyla birlikte klasik Newton fiziği ve yarıiletken fiziğinden farklı bir fizik kuramı yasasına dayanan kuantum mekaniğini kullanan kuantum bilgisayarları tasarlanmıştır. Uzun süredir üzerinde çalışmalar yapılan kuantum algoritmalarının gerçek kuantum bilgisayarları ve yüksek işlem gücüne sahip kuantum simülatörleriyle test edilmesi mümkün olmuş, bu durum beraberinde kuantum bilişimi dediğimiz farklı bir alanı ortaya çıkartmıştır.

Lider iş piyasası analitiği uzmanı “Burning Glass Technologies” isimli şirketin 2020 yılında yapmış olduğu bir pazar araştırmasının sonuçları, “Top 10 Tech Job Skills Predicted To Grow The Fastest In 2021” başlığıyla Forbes dergisinde yayınlanmıştır. Araştırma sonuçlarına göre 2021 yılında en hızlı gelişim göstereceği tahmin edilen yetkinlikler Şekil 2.1’de (<https://www.forbes.com/sites/louiscolumbus/2021/12/27/top-10-tech-job-skills-predicted-to-grow-the-fastest-in-2021/?sh=65b871576fde>) gösterilmiştir. Bu sonuçlara göre kuantum bilişiminin çok hızlı bir yükseliş göstereceği açıkça görülmektedir. Önümüzdeki birkaç yıl içinde bu konuda yetkin ve iyi donanımlı uzman ihtiyacında hızlı bir yükseliş yaşanacağı açıktır.



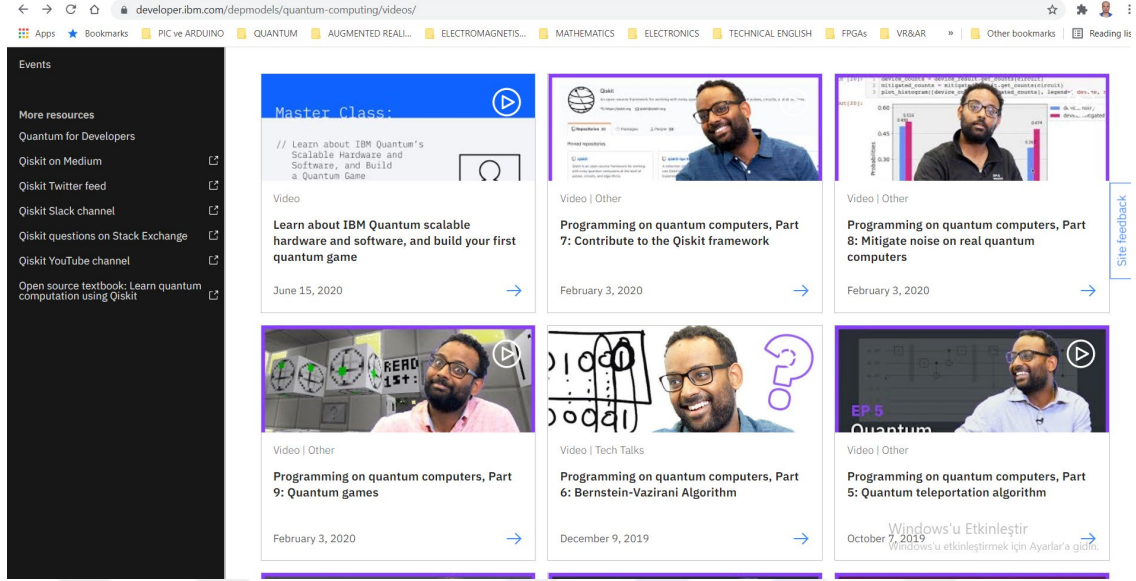
Şekil 2.1 “Burning Glass Technologies” firmasına göre 2021 yılında en hızlı gelişim göstereceği tahmin edilen yetkinlikler

Bu çalışma kapsamında da yeni eğilimlerin doğuracağı fırsatları görmek suretiyle, farklı mühendislik disiplinlerinde okuyan ve kuantum bilişiminin temel ilkeleri hakkında bilgi sahibi olmak isteyen kişilerin güncel eğitim teknolojileri eşliğinde eğitilmesini sağlayacak bir platform konsepti ortaya konmuştur.

2.1. Tasarım Konseptinin Planlanması

Tasarımın ana iskeletini Unity3D yazılım platformu kullanılarak tasarlanan 3 boyutlu ortam oluşturmaktadır. Bu çalışma, Unity'nin 3D tasarım stüdyosunun VR uygulamaları için kullanımına ve bu şekilde hedeflenen eğitim platformunun gerçekleştirilmesine odaklanmaktadır. Unity, 2 boyutlu tasarımı ve geliştirilmiş bir C# tabanlı komut dosyası dilini de destekleyen popüler bir 3 boyutlu oyun tasarım programıdır (Ferrone, 2019). Bu platform her ne kadar 2 veya 3 boyutlu oyun tasarımı platformu olarak lanse edilse de güçlü araçları ile sürükleyici bir eğitim platformu tasarımı sunan yapıya sahiptir (Linowes, 2020). Böylece günümüz modern eğitim araçlarının kullanımı ile kuantum programlamanın dikkat çekici ve eğlenceli hale getirilmesi istenmektedir.

IBM'in “Quantum computing Videos” başlığı altında sunmuş olduğu eğitim portalında, (bkz. Şekil 2.2) “Quantum games” başlığıyla, kuantum bilgisayarlarında programlamayı oyun temelli olarak öğreten bir sunuma da yer verilmiştir (<https://developer.ibm.com/videos/programming-on-quantum-computers-pt-9/>).



Şekil 2.2 IBM Kuantum bilişimi eğitim videoları

Çalışmanın ilk belirlenen konseptinde, kuantum kapılarının 3 boyutlu modelleri kullanılarak sürükleyip bırak yöntemiyle VR ortamda kurgulanan tasarımın, IBM Quantum Experience sunucularına API Token anahtarıyla bağlantı kurup, farklı bir platform üzerinden kuantum simülatörlerine ya da bilgisayarlarına veri gönderimi gerçekleştirmek vardı. Ancak IBM firması bulut sunucularına dış kaynaklı olarak yalnızca Qiskit platformu üzerinden bağlantıya izin verdiği için, “Composer” ve “Lab” uygulamalarına VR ortamından erişime imkân verecek 3 boyutlu bir web tarayıcısının sisteme entegre edilmesi arayışına girilmiştir. Bu süreçte amacı gerçekleştirmeyi sağlayacak açık kaynaklı Vuplex 3D WebView tarayıcısı çalışmaya entegre edilmiştir.

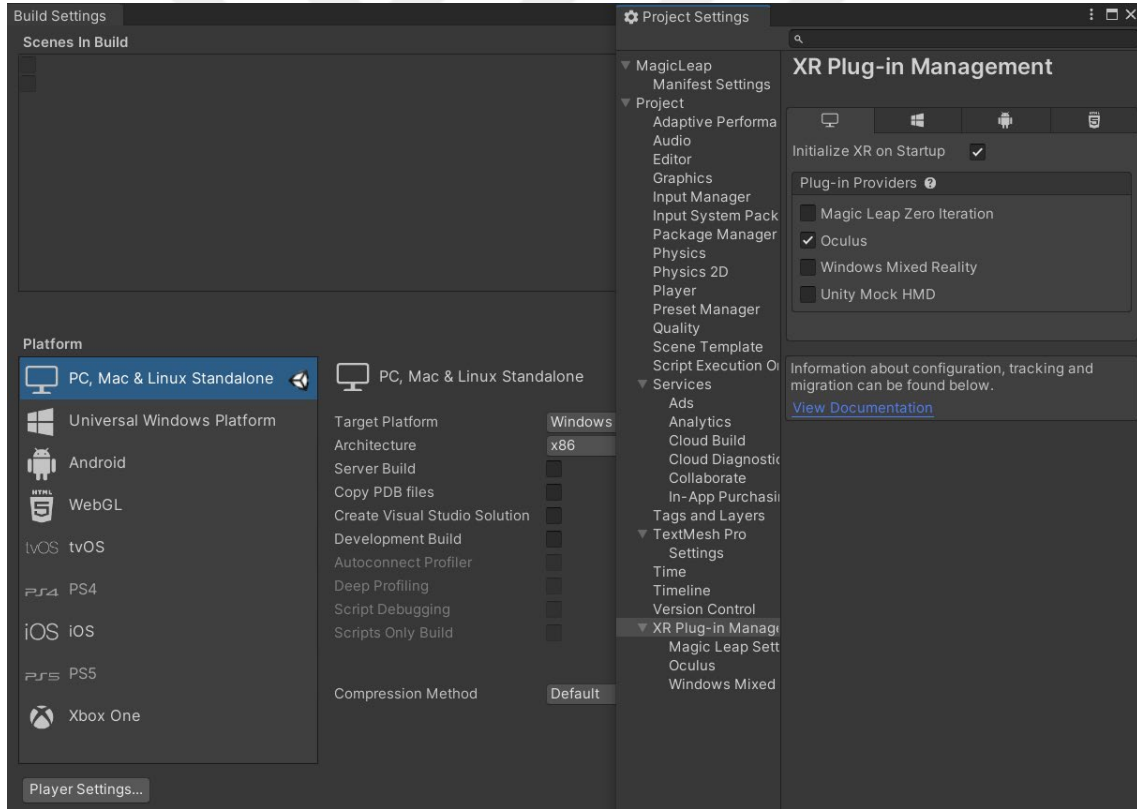
Kuantum bilişimi, lineer cebir ve vektör matematiğinin temellerini ve önemli miktarda fizik bilgisini gerektirir. Günümüzde kuantum bilgisayarlarının programlanmasında yaygın olarak kullanılan platformlar kuantum kapılarını kullanmaktadır. Kuantum kapıları kullanarak program yazmak, von Neumann mimarisi bilgisayarlarını makine dilinde programlamaya benzer. Bu yüzden yeni başlayanlar için bu biraz karmaşık görünebilir.

Jupyter Notebook platformunda python kodu ile üst düzey programlama bu sorunu bir nebze ortadan kaldırırsa da sanal gerçeklik platformunda hazırlanacak bir uygulama kapıların davranışlarını öğretmek için daha eğlenceli bir yol sağlayabilir.

Öncelikle uygulamada kullanılacak malzemelerin hangi platformda olacağına karar

verilmiştir. Daha sonra IBM kuantum besteleycisine benzeyen bir arayüz kullanılması ve bu arayüzün 3D VR ortamında nasıl görüntüleneceği kararlaştırılmıştır. Ayrıca bu VR ortamına gerçek zamanlı bir web tarayıcısı eklenmiştir. Böylece, VR ortamında gerçekleştirilen uygulamaların gerçek derleme sonuçlarının platforma entegre edilmiş tarayıcı üzerinden alınması amaçlanmıştır.

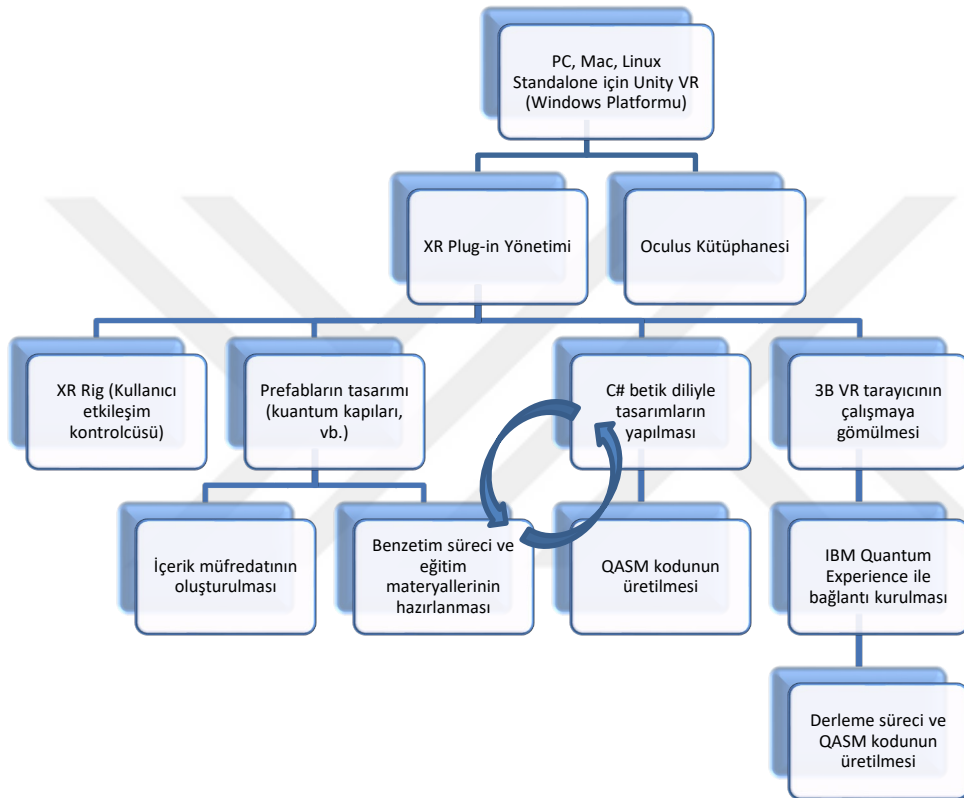
Öncelikle tasarımın yapılacağı şablon olarak URP belirlenmiş ve ardından hangi VR platformunun kullanılacağına karar verilmiştir. Tasarım “Oculus Rift” donanımı üzerinde gerçekleştirileceği için PC, Mac ve Linux Bağımsız platformu seçildikten sonra gelişmiş VR, AR ve MR uygulamaları tasarlamak için kullanılan XR Plug-in Management ve Oculus kitaplıkları etkinleştirilmiştir. Şekil 2.2’de uygun ortamın oluşturulması için, Unity’nin “Build Settings” ve “Project Settings” pencerelerinde yapılan tercihler görülmektedir.



Şekil 2.3 Tasarım için uygun platformun seçimi ve gerekli ayarların yapılması

Vuplex VR/AR, kullanıcıya bazı gelişmiş araçlar ve 3D kullanıcı kontrollü web tarayıcı API'si sağlar. Kullanıcının C#tan JavaScript'e veya tam tersi şekilde mesaj göndermesine izin verir. Bu, web sayfalarının etkileşimli kontrolünü ve web sayfalarının herhangi bir

VR uygulamasında kullanılmasını sağlar. Bu tarayıcı bir ekran klavyesi ile birlikte gelir. Uygulamanın temel prensibi Şekil 2.4'te gösterilmektedir. Tasarlanan araç, hem kuantum besteleyicisinin bulut tabanlı uygulaması için bir web tarayıcısı hem de VR platformunda bir simülasyon süreci içermektedir. Bu, kullanıcının daha sürükleyici bir etki ile kuantum bilgisayarlarda kapıların kullanımını öğrenmesini sağlar. Bu araç, kullanıcıların hem VR ortamında hem de gerçek zamanlı bir web tarayıcısında çalışmasına olanak tanır.



Şekil 2.4 QU-VR eğitim aracının tasarım süreci

2.2. Eğitim İçeriğinin Belirlenmesi

Eğitimin hiyerarşik bir süreç şeklinde işletilmesine karar verilmiş ve böyle bir modele kaynaklık etmesi açısından Microsoft firmasının kuantum eğitimleri için tasarlamış olduğu kuantum kataları (<https://docs.microsoft.com/tr-tr/azure/quantum/tutorial-qdk-intro-to-katas>) ve Hughes ve ark.nın (2021) kitap içeriği referans alınmıştır. Çalışmanın 2 kübitlik bir model üzerinden yürütülmesi ile temel konseptin oturtulması hedeflenmiştir. Bunun için;

- Temel kapılar,

- Ölçme işlemleri,
- Çok kübitli kapılar,
- Süperpozisyon,
- Grover algoritması ve
- Işınlanma (Teleportation) uygulamalarının ele alınmasının yeterli olacağı değerlendirilmiştir.

Microsoft kuantum kataları, IBM Qiskit platformuna benzeyen ve Q# olarak adlandırılan bir kuantum programlama diliyle çalıştırılan bir dizi eğitim setini ifade etmektedir. Q# yazılımı Qiskit platformuna benzer şekilde Jupyter Notebook kernelini kullanan web tabanlı bir geliştirme platformudur. Uygulama kapsamında ele alınacak konuları maddeler halinde kısaca açıklayabiliriz.

2.2.1. Temel Kapılar

Temel kapılar uygulamasında başta Pauli-X (Not), Pauli-Y ve Pauli-Z kapıları gibi tek kübitli kapıların davranışının öğretilmesi hedeflenmektedir. Bu uygulamada, ölçme işlemi gerektirmeden kapıların mevcut kübitlerin üzerindeki etkisinin ne olduğu araştırılır.

Uygulama hedefleri “Challenge” başlıklarıyla kullanıcıya sunulmaktadır ve örnek bir uygulama hiyerarşisi şu şekilde gösterilmiştir:

Challenge-1: $|0\rangle$ evresini $|1\rangle$ evresine dönüştüren ya da tersini gerçekleştiren kapıyı seçin.

Girdi: Bir kübit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ evresindedir.

Çıktı: $|\psi\rangle = \beta|0\rangle + \alpha|1\rangle$ evresine dönecektir.

Gerek IBM kuantum besteleyicisi gerek Microsoft kuantum simülatöründe olsun, her kübit öncelikle belli bir başlangıç değerine optimize edilmektedir. Örneğin IBM kuantum besteleyicisinde tüm kübitler başlangıç değeri olarak $|0\rangle$ evresine ayarlanır. Microsoft simülatöründe ise temel kapı uygulamalarında başlangıç durumu $|\psi\rangle = 0.6|0\rangle + 0.8|1\rangle$ şeklindedir.

Tasarladığımız uygulamada bunlardan farklı olarak, bir kübit evresi ($|\psi\rangle$) rastgele olarak her simülasyonda farklı tamsayı genliklerinde olacak şekilde konumlandırılır. Bu durum Bloch küresiyle de kullanıcıya gösterilir. Ardından VR ortamda kullanıcı kumandayla

doğru kapıyı seçip yerine yerleştirdiğinde doğru dönüşüm işlemi yapılır ve yeni sonuç Bloch küresinde gösterilir. Böyle bir uygulamanın, Microsoft katalarının uygulaması sonucu elde edilen çıktısı Şekil 2.5’te gösterilmiştir.

In [1]: %kata T101_StateFlip

```
operation StateFlip (q : Qubit) : Unit is Adj+Ctl {
  // The Pauli X gate will change the |0> state to the |1> state and vice versa.
  X(q);
}
```

The starting state:

Qubit IDs	0		
Basis state (bitstring)	Amplitude	Meas. Pr.	
0>	0.6000 + 0.0000i		36.0000%
1>	0.8000 + 0.0000i		64.0000%

The desired state:

Qubit IDs	0		
Basis state (bitstring)	Amplitude	Meas. Pr.	
0>	0.8000 + 0.0000i		64.0000%
1>	0.6000 + 0.0000i		36.0000%

The actual state:

Qubit IDs	0		
Basis state (bitstring)	Amplitude	Meas. Pr.	
0>	0.8000 + 0.0000i		64.0000%
1>	0.6000 + 0.0000i		36.0000%

Şekil 2.5 Örnek bir temel kapı uygulaması

Şekilde görüldüğü üzere kübit evresi işlem sonucu $|\psi\rangle=0.8|0\rangle+0.6|1\rangle$ durumuna dönmüştür. VR ortamında yapılan tasarımda ise girdinin farklı evrelerde olması durumu da canlandırılmıştır.

2.2.2. Ölçme İşlemleri

Kuantum bilişiminin doğası açıklanırken, daha önce de söylendiği gibi, ölçme işlemi gerçekleştirilmediği sürece kübitin hangi evrede olacağı bilinmemektedir. Ölçme işlemi gerçekleştiğinde süperpozisyon evresi çöker ve eğer sistem tek kübitlikse ilgili kübit $|0\rangle$ ya da $|1\rangle$ evresine geçer. İki kübitlik bir sistemse, kübitlerin evresi $|00\rangle$, $|10\rangle$, $|01\rangle$ ya da $|11\rangle$ evresinden birine geçer.

VR ortamında planlanan tasarımda, ölçme işlemi öncesi tek kübit ya da iki kübitlik bir sistem, olasılık genliklerinin kompleks eşlenikleriyle çarpımlarının toplamı 1 olacak

şekilde rastgele olarak bir evreye alınır. Örneğin;

Tek kubitlik bir sistem için, $|\psi\rangle=0.8|0\rangle+0.6|1\rangle$

Çift kubitlik bir sistem için, $|\psi\rangle=0.5|00\rangle+0.5|10\rangle+0.5|01\rangle+0.5|11\rangle$ süperpozisyon evreleri oluşturulur. Ölçme işlemi gerçekleştirildiğinde süperpozisyon evresinin çökmesi canlandırılır. Deterministik bilgisayarlar ile kuantum bilişimi işlemi gerçekleştirmenin zorluğu burada ortaya çıkmaktadır. Verilen örneklerde kompleks genlik değerleri yerine tamsayı değerleri kullanılmıştır.

Kuantum bilişiminde kompleks genliklerin eşlenikleriyle çarpımlarının toplamalarının 1 olması doğal olarak gerçekleşir. Bunun iki kubitlik bir sistem için bile simülasyonunu gerçekleştirmek oldukça yüksek bir kaynak tüketimine neden olacaktır. Bu durumu şu şekilde açıklayabiliriz:

Kompleks genliklerimizi $(a+ib)$ formunda yazacak olursak, iki kubitlik bir sistemin genlik dağılımı,

$|\psi\rangle=(a+ib)|00\rangle+(c+id)|01\rangle+(e+if)|10\rangle+(g+ih)|11\rangle$ şeklinde olacaktır.

$(a+ib)(a-ib) + (c+id)(c-id) + (e+if)(e-if) + (g+ih)(g-ih) = 1$ durumunu sağlamalıdır.

2.2.3. Çok Kubitli Kapılar

Çok kubitli kapılarla ilgili uygulamada, iki kubitli kapıların en önemlilerinden olan CNOT ve SWAP kapıları kullanılmıştır. CNOT kapısı kuantum bilişiminde çok önemlidir ve VR tasarımı bu kapının kullanımı gerçekleştirilmiştir. Uygulama, 1.4.4. nolu başlıkta açıklandığı üzere IBM kuantum bilişimi platformunda tercih edilen CNOT kapısına göre gerçekleştirilmiştir. Kontrol kubitinin (q_0) 1 olması durumunda hedef kubitin (q_1) evresi değişmektedir. IBM'in kullandığı ket notasyonunda iki kubitlik bir sistemde kubitlerin dizilimi $|q_1q_0\rangle$ şeklindedir. Benzer şekilde üç kubitlik bir sistemde ise kubitlerin dizilimi $|q_2q_1q_0\rangle$ şeklinde gerçekleşir.

$CNOT|00\rangle=|00\rangle$

$CNOT|01\rangle=|11\rangle$

$CNOT|10\rangle=|10\rangle$

$CNOT|11\rangle=|01\rangle$

VR uygulamada kullanıcıların, Hadamard kapısıyla birlikte CNOT'ın kullanımı sonucu, kuantum bilişiminin en önemli fenomenlerinden olan Bell evresini anlamaları hedeflenmiştir.

İki kubitli kapılardan bir diğer önemli kapı ise SWAP kapısıdır.

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

İki kubitlik sistemde kubitler farklıysa, SWAP kapısıyla bu kubitlerin evreleri yer değiştirilir.

$$\text{SWAP}|00\rangle = |00\rangle$$

$$\text{SWAP}|01\rangle = |10\rangle$$

$$\text{SWAP}|10\rangle = |01\rangle$$

$$\text{SWAP}|11\rangle = |11\rangle$$

2.2.4. Süperpozisyon

Kuantum bilişiminde kubitlerin süperpozisyona alınması için kullanılan Hadamard kapısının VR ortamda ayrıntılı uygulamaları tasarlanmıştır. Örnek uygulamalardan bazıları şu sunulan işlemleri gerçekleştirmektedir.

Challenge-2: $|0\rangle$ evresinde olan bir kubit $|+\rangle$ evresine dönüştüren kapıyı seçin.

Girdi: Bir kubit $|\psi\rangle = |0\rangle$ evresindedir.

Çıktı: $|\psi\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ evresine dönecektir.

In [2]: %kata T101_PlusState

```
operation PlusState (q : Qubit) : Unit {  
  H(q);  
}
```

The desired state:

Qubit IDs	0		
Basis state (bitstring)	Amplitude	Meas. Pr.	
0>	0.7071 + 0.0000i		50.0000%
1>	0.7071 + 0.0000i		50.0000%

The actual state:

Qubit IDs	0		
Basis state (bitstring)	Amplitude	Meas. Pr.	
0>	0.7071 + 0.0000i		50.0000%
1>	0.7071 + 0.0000i		50.0000%

Şekil 2.6 Tek kubitlik bir süperpozisyon uygulaması

Challenge-3: İki kubitlik bir sistemin tüm temel vektörlerini süperpozisyona sokacak kapı düzenini gerçekleştirin.

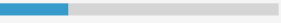
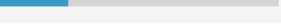
Girdi: İki kubitlik bir sistem $|\psi\rangle=|00\rangle$ evresindedir.

Çıktı: $|\psi\rangle=|+\rangle \otimes |+\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$ evresine dönecektir.

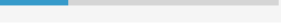
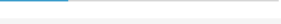
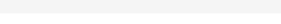
```
In [15]: %kata T103_AllBasisVectors_TwoQubits

operation AllBasisVectors_TwoQubits (qs : Qubit[]) : Unit {
  H(qs[0]);
  H(qs[1]);
}
```

The desired state:

Qubit IDs	0, 1		
Basis state (bitstring)	Amplitude	Meas. Pr.	
00>	0.5000 + 0.0000i		25.0000%
01>	0.5000 + 0.0000i		25.0000%
10>	0.5000 + 0.0000i		25.0000%
11>	0.5000 + 0.0000i		25.0000%

The actual state:

Qubit IDs	0, 1		
Basis state (bitstring)	Amplitude	Meas. Pr.	
00>	0.5000 + 0.0000i		25.0000%
01>	0.5000 + 0.0000i		25.0000%
10>	0.5000 + 0.0000i		25.0000%
11>	0.5000 + 0.0000i		25.0000%

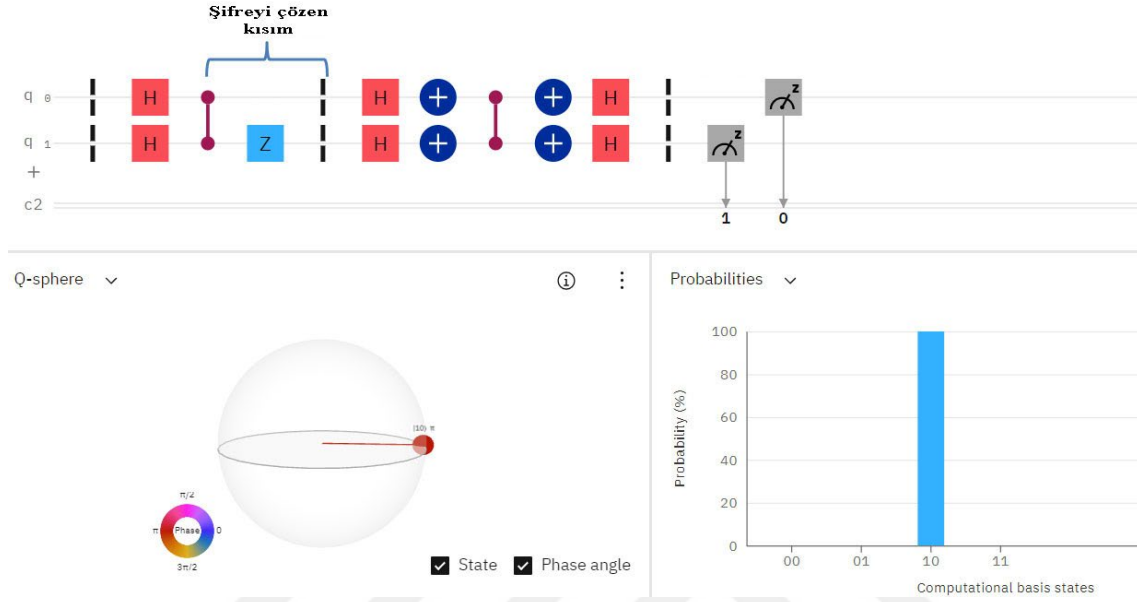
Şekil 2.7 Çift kübitlik bir süperpozisyon uygulaması

2.2.5. Grover Algoritması

Grover algoritması bizzat algoritmaya adını veren Lov Kumar Grover tarafından (Grover, 1996) açıklanmış bir hızlı tarama algoritmasıdır. Algoritmayı önemli kılan, bir kuantum bilgisayarının klasik bir bilgisayardan daha yüksek işlem hızına sahip olacağını ifade eden ilk algoritma olmasıdır (Dwyer, 2020). Bu algoritma, rastgele dizilmiş N öğeli bir veri tabanında, belirli bir öğeyi büyük O 'su $\sqrt{N}$ olacak bir hızda bulabilmeyi taahhüt eder (Abhijith, 2020). Algoritmayı önemli kılan bir diğer özellik ise diğer bazı problemlere de uygulanabilen ve “kuvvetlendirme” olarak isimlendirilen bir ilkeye sahip olmasıdır (Dwyer, 2020).

Grover algoritmasının temel mantığı, aranacak bir bilginin olasılık genliğinin diğerlerine göre kuvvetlendirilerek ayırt edilebilir hale getirilmesine dayanmaktadır. Örneğin 2 bitlik bir şifrenin tahmin edilmek istendiğini varsayalım. Klasik bir bilgisayarda her bir olası durum denenecektir. Böyle bir durumda 2^N eşitliğinden $2^2 = 4$ kadar deneme durumu ortaya çıkar. Grover algoritması, bu işlemin kuantum bilişimiyle yapılması durumunda kare kökü ($\sqrt{2^N} = \sqrt{2^2} = \sqrt{4} = 2$) kadar bir süreçte halledilebileceğini ortaya koymaktadır.

Şekil 2.8’de 2 kubitlik bir sistem için Grover algoritmasının uygulama örneği gösterilmiştir. Bu uygulama örneğinde 2 bitlik bir şifrenin tahmin edilmesi istenmektedir. Şekilde, “Şifreyi çözen kısım” olarak gösterilen yere tahmini gerçekleştirecek ve “kâhin (oracle)” olarak adlandırılan kapı ya da kapılar gelir.



Şekil 2.8 Grover algoritmasının uygulama örneği

Algoritmanın anahtar ögesi çift düğümlü olarak görülen CZ (Controlled-Z) kapısıdır. CNOT kapısına benzer olarak, kontrol kubitinin $|1\rangle$ evresinde olması durumunda kontrol edilen kübityi Z ekseninde π radyan kadar döndürür. Bu durumda işleme tabi tutulan evrenin yalnızca işareti değişmiş olur. İki kubitlik bir sistem için,

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \text{ evresi}$$

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle - d|11\rangle \text{ evresine döner.}$$

Görüldüğü üzere iki kubitlik bir sistemde yalnızca $|11\rangle$ evresinde faz değişimi olmaktadır ($e^{i\pi} = -1$).

İki kubitlik bir sistem için, Grover algoritması şekildeki bariyerlerle gösterildiği gibi üç bölümden oluşmaktadır.

- İlk kısımda tüm olası şifreleri süperpozisyona almak için Hadamard kapıları kullanılmaktadır.

- İkinci kısım şifreyi bulacak olan kâhinin yerleştirildiği kısımdır. Bu örnekte şifre ‘10’ dır ve q1 hattına Pauli-Z kapısı yerleştirilmiştir. Bu bölgede gerçekleştirilecek farklı kapı dizilimleriyle diğer muhtemel şifre değerleri çözümlenebilir. VR ortamında kullanıcılardan bu bölgeye müdahale etmeleri beklenmektedir.
- Üçüncü kısım Grover kuvvetlendirmesi işleminin gerçekleştirildiği yerdir ve iki kübitlik bir sistem için sabittir. Daha fazla kübitlik bir sistem için sistem daha karmaşık olacaktır.

Grover kuvvetlendirmesinin temel ilkesini kısaca şu şekilde açıklayabiliriz:

Kâhinin temel görevi doğru şifreyi etiketlemek ve yalnızca olasılık genliğinin fazını değiştirmektir. Böylece olası tüm şifrelerin süperpozisyon evresi korunmuş olur. Örneğin şifre ‘10’ ise kâhin $1/2(|00\rangle+|01\rangle-|10\rangle+|11\rangle)$ süperpozisyon evresini üretir. Eğer bu süperpozisyon evresi ölçülecek olursa tüm mümkün şifrelerin olasılıkları $1/4$ olacaktır. Negatif olasılık genliklerinin de kare işleminden dolayı olasılıkları pozitif olduğundan, burada farklı bir işlem uygulanarak negatif genlikli evrenin diğerlerinden ayırt edilmesi sağlanır.

Her bir olasılık genliğinin karesi yerine ortak genliğin karesi alınır ve ardından süperpozisyon evresinin genlik ortalaması çıkartılır.

$$0.5^2(+1+1-1+1) = 0.5$$

Ardından ortalama değerin her bir katsayıya göre farkı alınır.

$$1-0.5 = 0.5, 1-0.5 = 0.5, -1-0.5 = -1.5, 1-0.5 = 0.5$$

Ardından bu farkların tersleri alınır ve ortalama ile toplanır.

$$-0.5+0.5 = 0, -0.5+0.5 = 0, 1.5+0.5 = 2, -0.5+0.5 = 0$$

Son işlemde, görüldüğü üzere ‘10’ evresinin genlik katsayısı 2 olurken, diğerleri 0’a çekilmiştir. Süperpozisyon evresinin yeni durumu şu şekilde gerçekleşir:

$$|\psi\rangle=0.5(0|00\rangle+0|01\rangle+2|10\rangle+0|11\rangle) = |10\rangle$$

Son işlemle birlikte tespit edilmek istenen evrenin genlik olasılığı %100’e çıkartılmıştır. Daha yüksek kübitlik işlemlerde tespit edilmek istenen evrenin %100 doğrulukta belirlenebilmesi için daha fazla sayıda tahmin işlemi gerçekleştirilmesi gerekecektir.

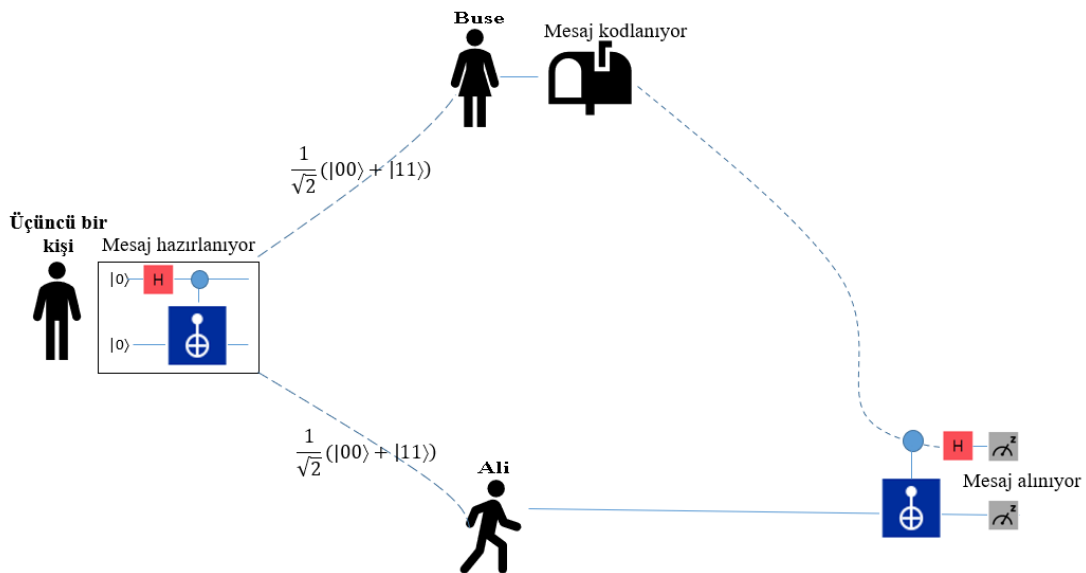
Örneğin $N=100$ olan bir dizide $\sqrt{100} = 10$ adımlık bir proses sonucunda %100 doğruluk değerine ulaşılabacaktır.

2.2.6. Işınlanma (Teleportation)

Kuantum ışınlama, bir konumdaki göndericiden bir miktar uzaktaki alıcıya kuantum bilgilerini aktarmak için kullanılan bir tekniktir (Hughes, 2021). Çağrışım yaptığı anlamdan farklı olarak nesnelerin değil kuantum bilgisinin transferini ifade eder. Bu sürecin işleyişini Buse ve Ali olarak isimlendirilen iki hayali karakter üzerinden açıklayabiliriz. Buse'nin Ali'ye 2-bitlik bir mesaj göndermek istediğini varsayalım. Mesaj, ışınlanma olarak adlandırılan bir süreçle iletilecektir. Kuantum bilişiminde ışınlanma protokolü aşağıda takip eden adımlardan oluşur:

- **Hazırlık:** Buse ve Ali isimleriyle ifade edilen kişilere gönderilmek üzere dolanıklık durumunda olan bir çift kübitin oluşturulmasıdır.
- **Mesajın iletimi:** Mesaj kübiti Buse'nin kübitiyle dolanıklık haline getirilir ve Ali'ye gönderilmek üzere iki klasik bit çıkartılır.
- **Orijinal mesajın oluşturulması:** Ali'nin kübitini mesaj kübitinin orijinalde olduğu duruma getirmek için, Ali'nin Buse'den aldığı iki klasik bit kullanılır.

Bu adımların temsili sunumu Şekil 2.9'da (Dwyer, 2020) gösterilmiştir.



Şekil 2.9 Teleportasyon sürecinin temsili gösterimi

Hazırlık aşamasında kullanıcıdan dolanıklık durumunda iki kübit hazırlamaları istenir.

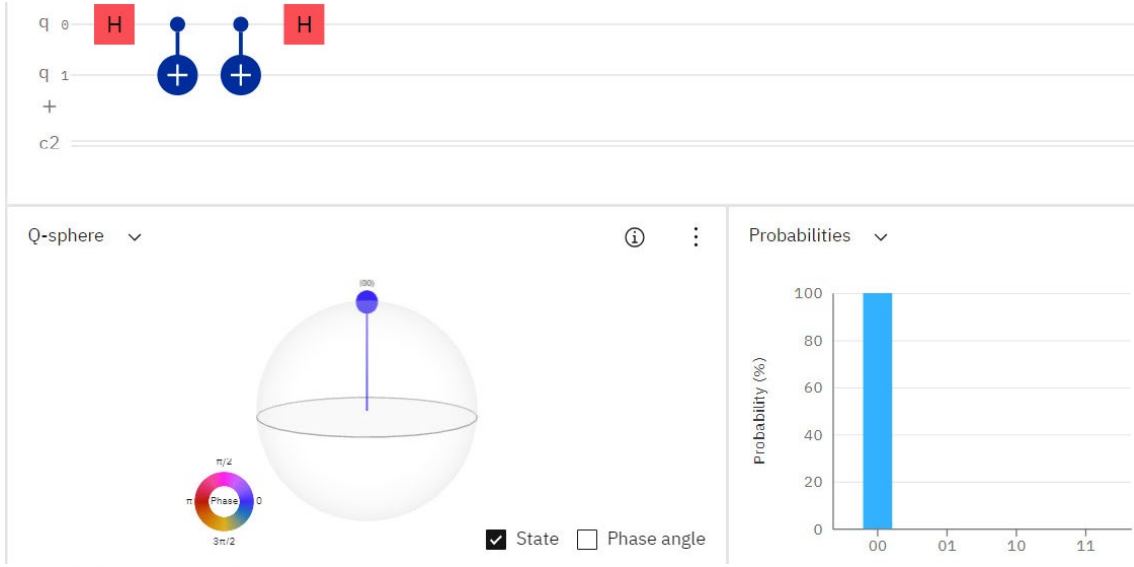
Bu durumda kullanıcı sırasıyla $H(qBuse)$ ve $CNOT(qBuse, qAli)$ kapılarını işletir. Burada $qAli$, Ali'ye ait kubit evresini, $qBuse$ ise, Buse'ye ait kubit evresini ifade eder. $qAli$ ve $qBuse$ kubitleri başlangıçta $|0\rangle$ evresindedirler. Bu aşamada pozitif Bell evresi ($|\Phi^+\rangle$) elde edilmiş olur. Bu işlem sonucunda, Buse Ali'nin kubitini üzerine yalnızca kendisinin nasıl okunacağını bildiği bir bilgi, Ali ise Buse'nin kubitini üzerine yalnızca kendisinin nasıl okunacağını bildiği bir bilgi yazar.

Hadamard kapısıyla dolanıklık haline getirilmiş bir sistemde kubitlerin evresi çökmeden orijinal bilginin okunabilmesi tarama tekniği denilen bir yöntemle yapılır. Tarama işlemiyle ölçme işleminin yapılmasından önce sisteme tekrar Hadamard kapısı uygulanır. Hadamard kapısının kendisiyle çarpımı birim matris elde edilmesini sağladığından orijinal bilginin sistemin evresine zarar vermeden okunması mümkündür.

Bir sonraki adımda, Buse 2-bitlik bir mesaj kodlamak için kendi kubitine bir işlem yapar. Bu işlemde evresi bilinmeyen 3'ncü bir kubit ($qTeleport$) gerekmektedir. Bu süreçte Buse yalnızca klasik mesajlaşma yöntemleriyle iletim sağlayabilmektedir. Ali'nin kendi kubit setindeki bilgiyi okuyabilmesi için Buse'nin ona bu bilgiyi nasıl okuyabileceğinin açıklamasını yaptığı klasik bir mesaj iletimi gerçekleştirmelidir. Kısacası her durumda kodlanmış bilginin Ali'ye iletiminde, kablosuz haberleşme, fiber optik haberleşme ya da klasik bakır hat haberleşmesi gibi sinyallerin ışık hızı sınırı içinde iletilebildiği bir altyapı kullanılması gerekmektedir.

Bu işlemi yaptıktan sonra mesajı Ali'ye gönderir. Ardından Ali'nin tarafında ters Bell evresi işlemi gerçekleştirilir ve Buse'nin tarafında yapılmış kodlama işlemi çözüme hazır hale getirilir. Son aşamada klasik bitler üzerinde ölçme işlemleri yapılarak orijinal mesaj türetilir.

Şekil 2.10'da bu sistemin kuantum besteleycisindeki yerleşimi gösterilmiştir. Şekilde görüldüğü üzere başlangıçtaki duruma geri dönmek için sistemin çıkışında Bell evresinin tersi işlem gerçekleştirilmiştir. Böylece arada Buse'nin yapacağı her bir değişiklik onun göndereceği mesaj bilgisi olarak ortaya çıkacaktır.



Şekil 2.10 Bell evresinin başlangıç durumuna döndürülmesi

Buse'nin kendisine gelen kübit evresine yapacağı işlem sonucu ortaya çıkacak mesaj için arada yapılacak işlem örneği Şekil 2.11'de gösterilmiştir.



Şekil 2.11 Örnek kodlama

Şekilde görüldüğü üzere araya Pauli-X (değil) kapısı uygulanmıştır. Böylece $|00\rangle$ evresinin q0 kübiti $|0\rangle$ evresinden $|1\rangle$ evresine, $|11\rangle$ evresinin q0 kübiti $|1\rangle$ evresinden $|0\rangle$ evresine döndürülmüştür. Ali'nin tarafında yer alan CNOT kapısıyla $|01\rangle + |10\rangle$ evresi $|10\rangle + |11\rangle$ evresine döndürülür. Ardından yine Ali'nin tarafındaki Hadamard kapısı uygulandığında, Ali'nin tarafında $|10\rangle$ evresi elde edilir. Bu işlemin en başından adım adım uygulaması şu şekildedir:

$$\begin{aligned}
1- & \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1x1 & + & 1x0 \\ 1x1 & - & 1x0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \\
2- & |0\rangle \otimes |+\rangle = |+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ 0 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|00\rangle + |01\rangle) \\
3- & \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \\
4- & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle) \\
5- & \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|10\rangle + |11\rangle) \\
6- & \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right) = \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle
\end{aligned}$$

Benzer şekilde Pauli-X kapısı yerine Pauli-Z kapısı kullanıldığında Ali'nin tarafında $|01\rangle$ evresi elde edilir. Pauli-X ve Pauli-Z kapıları birlikte kullanıldığında ise $|11\rangle$ evresi elde edilecektir. VR ortamındaki uygulama örneğinde ise iki ardışık kapının kullanılması yerine, Pauli-X ve Pauli-Z kapılarının birlikte kullanımına karşılık gelen Pauli-Y kapısı kullanılmıştır. Kısacası tablodaki X ve Z kapılarının birlikte kullanımı yerine yalnızca Y kapısıyla da benzer çıktı elde edilir. Tüm bu işlemleri ve çıktıları Tablo 2.1'de gösterildiği gibi özetleyebiliriz.

Tablo 2.1 İki bitlik mesaj iletimi

Buse'nin mesajı	Buse'nin kapıları	Ali'ye gönderilen kübit	Ali'nin CNOT kapısı sonrası	Ali'nin H kapısı sonrası
00	Yok	$ 00\rangle + 11\rangle$	$ 00\rangle + 11\rangle$	$ 00\rangle$
01	Pauli-Z	$ 00\rangle - 11\rangle$	$ 00\rangle - 01\rangle$	$ 01\rangle$
10	Pauli-X	$ 01\rangle + 10\rangle$	$ 10\rangle + 11\rangle$	$ 10\rangle$

11 X ve Z birlikte $|10\rangle - |01\rangle$ $|10\rangle - |11\rangle$ $|11\rangle$

Tüm olası evrelerin olasılık genliklerinin dağılımlarına da bakılacak olursa, Tablo 2.2’de gösterildiği gibi özetlenebilir.

Tablo 2.2 İki kübitlik bir sistemde olasılık genliklerinin dağılımı

Mesaj	Evre	H	CX	X	Z	CX	H
00	00	$1/\sqrt{2}$	$1/\sqrt{2}$			$1/\sqrt{2}$	1
	01	$1/\sqrt{2}$	0			$1/\sqrt{2}$	0
	10	0	0				0
	11	0	$1/\sqrt{2}$				0
01	00	$1/\sqrt{2}$	$1/\sqrt{2}$		$1/\sqrt{2}$	$1/\sqrt{2}$	0
	01	$1/\sqrt{2}$	0		0	$-1/\sqrt{2}$	1
	10	0	0		0		0
	11	0	$1/\sqrt{2}$		$-1/\sqrt{2}$		0
10	00	$1/\sqrt{2}$	$1/\sqrt{2}$	0			0
	01	$1/\sqrt{2}$	0	$1/\sqrt{2}$			0
	10	0	0	$1/\sqrt{2}$		$1/\sqrt{2}$	1
	11	0	$1/\sqrt{2}$	0		$1/\sqrt{2}$	0
11	00	$1/\sqrt{2}$	$1/\sqrt{2}$	0	0		0
	01	$1/\sqrt{2}$	0	$1/\sqrt{2}$	$-1/\sqrt{2}$		0
	10	0	0	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	0
	11	0	$1/\sqrt{2}$	0	0	$-1/\sqrt{2}$	1

Tablonun ilk sütununda Buse’nin Ali’ye ileteceği mesaj bilgisi yer alır. İkinci sütun iki kübitlik bir devrenin alacağı evreleri gösterir. Diğer sütunlar, gösterilen her bir kapının uygulandıktan sonra evrelerin aldıkları genlik dağılımlarını göstermektedir.

Yukarıda Teleportasyon işleminde Bell evresinin temel dönüşümü anlatılmıştır. Gerçek bir teleportasyon sürecinde 3’ncü bir kübit (qTeleport) gerektiği belirtilmişti. Buse Ali’ye iletme istediği ve evresini bilmediği 3’ncü bir kübit alır ve sonradan aldığı bu kübiti de

kendi kütitiyle CNOT kapısıyla etkileşime sokar. Bu işlemden sonra kendi kütiti (qBuse) ölçer. Ardından qTeleport Hadamard kapısına gönderilir ve çıkışı ölçülür. qBuse qAli ile dolanıklık halinde olduğundan, qTeleport ile CNOT etkileşimi qAli'nin de evresini değiştirir.

Ket notasyonunda bu üç kütiti $|qTeleport\ qBuse\ qAli\rangle$ şeklinde gösterilebiliriz. 3'ncü kütit qTeleport bilinmeyen bir evrede olduğundan, $\alpha|0\rangle + \beta|1\rangle$ şeklinde temsil edilir. 3 kütitlik teleportasyon süreci adım adım şu şekilde gösterilebilir:

- 1- Sisteme iletilecek qTeleport kütiti eklenmesiyle, Bell evresine sokulmuş qBuse ve qAli ile bu 3'ncü kütit çarpılır.

$$\frac{1}{\sqrt{2}}[\alpha|0\rangle + \beta|1\rangle](|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$$

- 2- Ardından CNOT kapısının uygulanmasıyla aşağıdaki durum değişimi gerçekleşir.

$$\frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle)$$

- 3- Hadamard kapısı uygulandıktan sonra 3 kütitin evresi aşağıdaki gibi değişecektir.

$$\frac{1}{2}(\alpha(|0\rangle + |1\rangle)|00\rangle + \alpha(|0\rangle + |1\rangle)|11\rangle + \beta(|0\rangle - |1\rangle)|10\rangle + \beta(|0\rangle - |1\rangle)|01\rangle)$$

- 4- Çarpımlar dağıtıldığında aşağıdaki sonuç elde edilecektir.

$$\frac{1}{2}(\alpha|000\rangle + \alpha|100\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|110\rangle + \beta|001\rangle - \beta|101\rangle)$$

- 5- Benzer terimler birleştirildiğinde aşağıdaki sonuç elde edilir.

$$\frac{1}{2}(|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle))$$

Son durumda parantez içindeki kütit evreleri Ali'nin kütiti 4 olası süperpozisyon evresini göstermektedir. Bunlar şu şekilde listelenebilir.

- $\alpha|0\rangle + \beta|1\rangle$
- $\alpha|0\rangle - \beta|1\rangle$
- $\alpha|1\rangle + \beta|0\rangle$
- $\alpha|1\rangle - \beta|0\rangle$

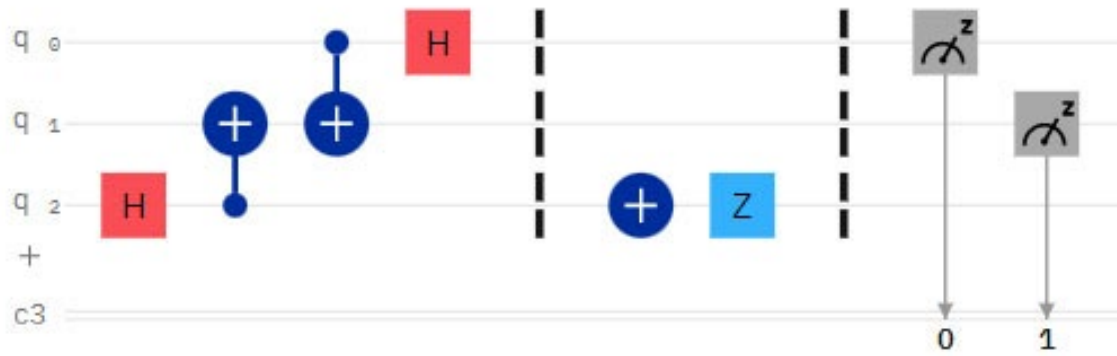
Görüldüğü üzere Ali'ye ait kütitin evresi, teleportasyon kütiti Buse'nin üzerinden CNOT ve Hadamard kapılarının uygulanmasıyla değişmiştir. Buse'nin ve teleportasyon

kübitlerinin ölçülmesinin nedeni, Ali'nin kübitinin bu ikisine bağlı olduğunu göstermektir. Artık bundan sonra Buse bilinen iletişim kanallarından biri üzerinden iki klasik biti Ali'ye gönderir. 5'nci adımdaki duruma göre ölçüm çıktısı 00, 10, 01, 11 durumlarından her birinin %25'lik dağılımı şeklinde gerçekleşecektir.

Teleportasyon süreci boyunca, iletilen qBuse'nin evresi $\alpha|0\rangle + \beta|1\rangle$ orijinal evresinde kalmaz. Böylece iletim sürecinin herhangi bir evresinde iletilen kübitin evresini kümse tam olarak bilemez. Buse'nin ölçümüne bağlı olarak Ali teleport edilen kübitin evresini X ve/veya Z kapılarını kullanarak orijinal durumuna getirebilir.

Tam teleportasyon işlemi kuantum kapılarıyla gerçekleştirildiğinde 3 kübitlik sistemin modeli Şekil 2.12'de gösterildiği gibi olacaktır. Burada;

q0 – qTeleport q1 – qBuse q2 – qAli kübitlerine karşılık gelmektedir.



Şekil 2.12 Buse ve Ali arasında teleportasyon

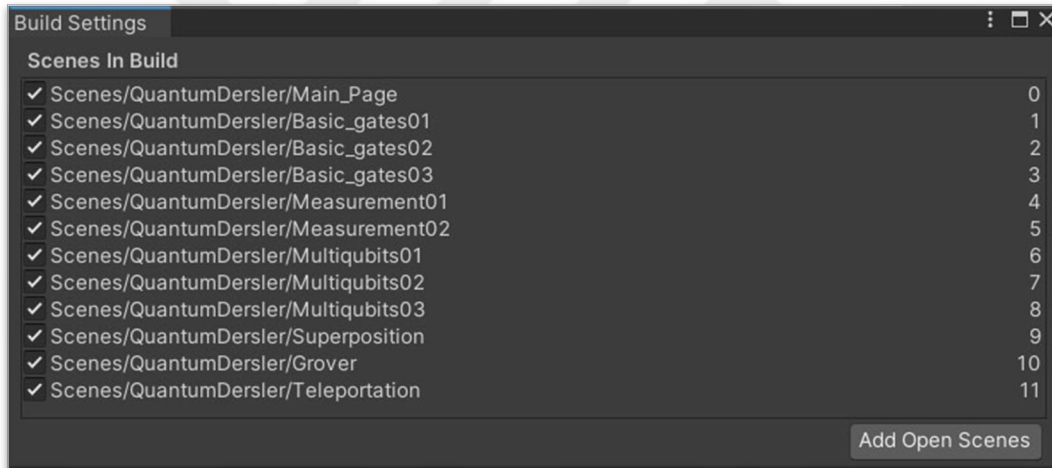
Bu protokol kuantum verisinin fiziksel bir karşılaşmaya gerek kalmadan yalnızca klasik iletişim kanalı yoluyla iletilmesini sağlar.

3. VR KUANTUM PROGRAMLAMA PLATFORMU

Unity 3D kullanılarak VR ortamında bir materyal tasarımının yapılması için örnek bir tasarım şablonunun ortaya konması gerekir. Unity bağımsız sahnelerden oluşan bir tasarım arayüzü ile gelir. Bu kapsamda öncelikle tasarımı gerçekleştirilecek sahnelere karar verilmiştir. Sahnelerde kullanılacak nesneler ve donanım unsurları için gerekli kütüphaneler belirlendikten sonra tasarım mimarisi oluşturulmuştur.

3.1. Sahne Yönetimi Tasarımı

Sahnelerin hiyerarşik bir düzende listelenmesi ve her bir sahne içinde birbirlerinin arasında geçiş yapılması için bir menü hiyerarşisi tasarlanmıştır. Şekil 3.1’de çalışma kapsamında konu içerikleri hazırlanmış sahnelerin “Build Settings” ekranındaki çıktısı gösterilmiştir.

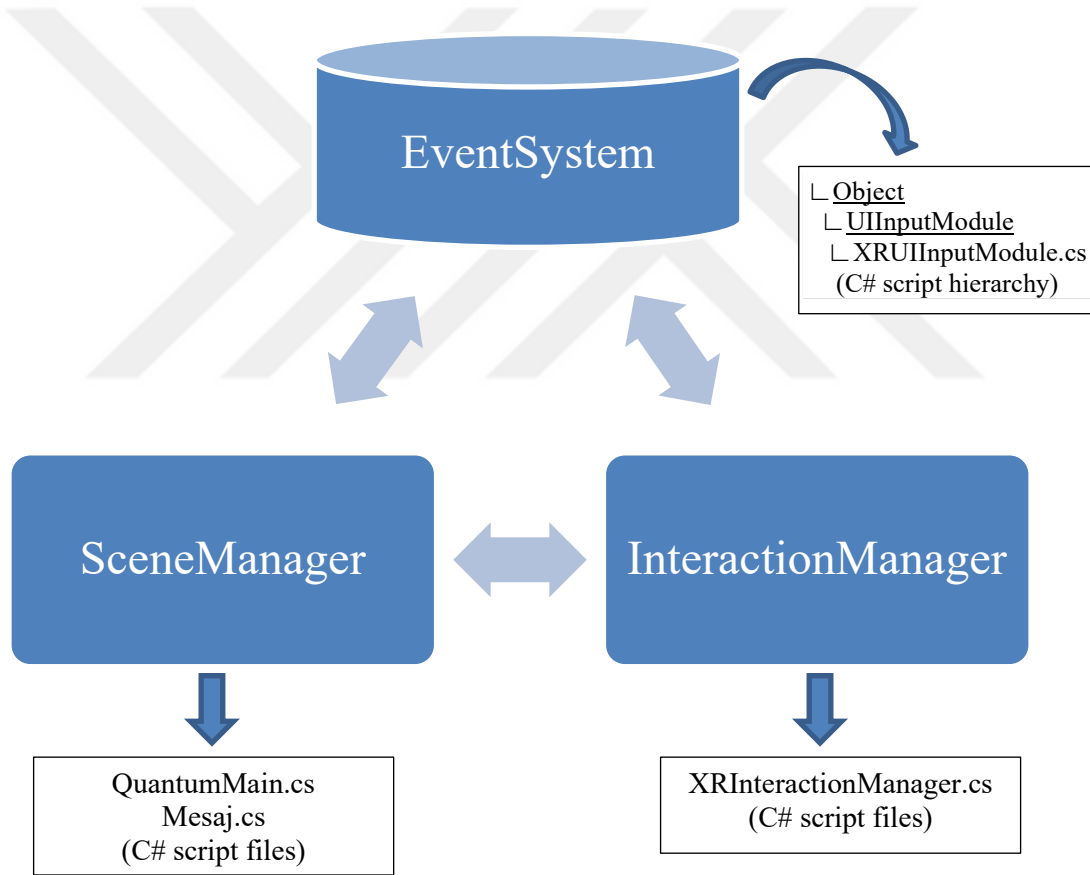


Şekil 3.1 Unity 3D ortamı için yapılan sahne tasarımları

Şekilde görüldüğü üzere toplamda 11 sahne vardır. İlk sahne tasarımın ana sayfasıdır ve bu sayfada temel kuantum kapıları ve kuantum bilgisayarının temelleri anlatılmaktadır. Uygulamanın belkemiğini üç ana blok yapı oluşturmaktadır. Bu yapının başında, Unity sahnelerindeki olayların ve eylemlerin işletilmesini sağlayan “EventSystem” olarak adlandırılan bir olay izleyicisi yer alır. Bu olaylar ve eylemler, fare ve el denetleyicisi gibi giriş elemanlarının ve “raycast” olarak adlandırılan ışının nesnelerle iletişimini kapsamaktadır. Her Unity sahnesinde yalnızca bir adet olay izleyicisi bulunur.

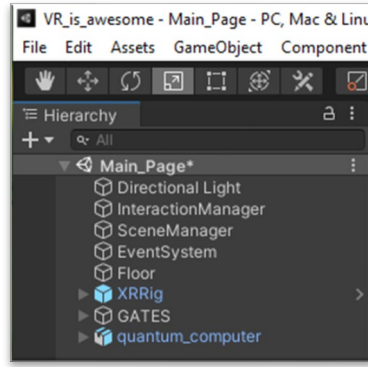
Bir diğer ana blok, VR ortamında 3 boyutlu ve UI etkileşimlerini kullanılabilir hale

getiren “InteractionManager” yapısıdır. Unity platformunda VR/AR uygulamaları için “XR Interaction Toolkit” aracının yüklenmesi gerekir ve “InteractionManager” bloğu bu aracın kütüphanesine erişmek için sahneye otomatik olarak eklenir. Bu iki yerleşik bloğun haricinde, kullanıcının sahnedeki tüm nesnelerle ve diğer bileşenlerle merkezi olarak ilgilenmesini sağlamak için “SceneManager” adıyla bir sahne yöneticisi eklenir. Sahne yöneticisi tasarımcı tarafından tasarlanan ve işlemlerin daha bütünsel şekilde gerçekleştirilmesini sağlamak amacıyla kullanılan bir yapıdır. Unity ya da harici kütüphaneler tarafından yerleştirilen “script” dosyaları haricinde, tasarımcı tarafından tasarlanmış tüm “script” dosyalarının bu bloğun içine yerleştirilmesine özen gösterilmiştir. Genel tasarım mimarisi Şekil 3.2’de gösterilmiştir.



Şekil 3.2 Unity’de genel tasarım mimarisi

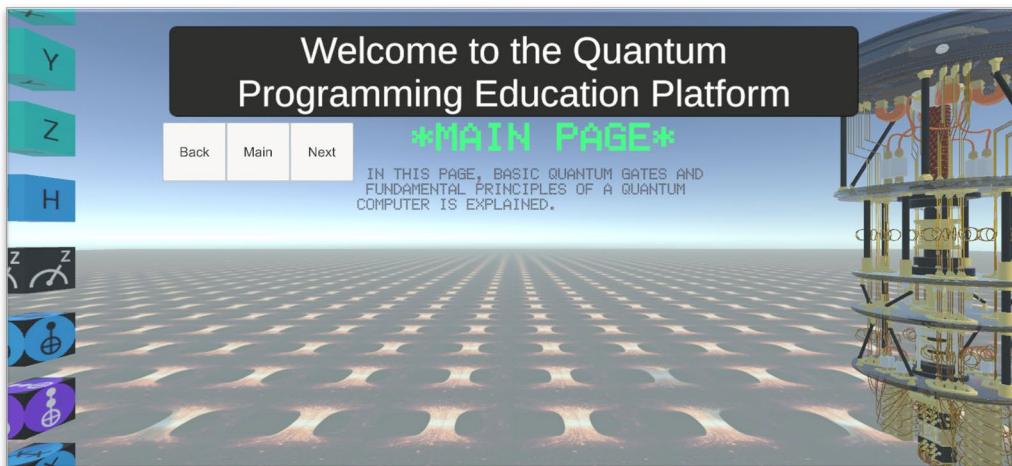
Bu üç ana bloğun haricinde sahnede kullanılan nesneler ve yardımcı dekorlar tasarım hiyerarşisinin düzenlendiği “Hierarchy” alanına eklenir. Şekil 3.3’te örnek olarak ana sayfa uygulamasının olduğu “Main_Page” sahnesinin nesne hiyerarşisi gösterilmiştir.



Şekil 3.3 Örnek bir tasarım hiyerarşisi

Şekilde görüldüğü üzere, “InteractionManager”, “SceneManager” ve “EventSystem” bileşenlerinin yanında, Oculus Rift donanım unsurlarını sahneye eklememizi sağlayan XRRig bileşeni, sahnede yer alan kapıların gruplandığı “GATES” nesnesi ve 3 boyutlu IBM Q kuantum bilgisayar modelinin parçalarını içeren “quantum_computer” nesnesi görülmektedir.

Şekil 3.4’te ana sayfadan bir kesit gösterilmiştir. Tasarım alanında, IBM Q kuantum bilgisayar referans alınarak tasarlanmış bir kuantum bilgisayar modeli görülmektedir. Sahnede kullanılan model ana sayfa için tasarlanmış olup, soğutma kanalları, altın kaplamalı pirinç çubukları, mikrodalga giriş hatları, amplifikatör katları arasında sinyal iletimi sağlayan fiber optik hatlar ve içinde işlemcinin yer aldığı elektromanyetik radyasyon koruyucusu gösterilmiştir.



Şekil 3.4 VR Eğitim platformu ana sayfası

Ana sayfada yer alan temel kapıların üzerine el denetleyicisi ile gelindiğinde, ilgili

kapının temel işlevleri kullanıcının karşısına çıkartılır. Kullanıcı bu sayfada kuantum bilgisayarını da el denetleyicisi ile kavrayabilmekte ve daha yakından bakabilmektedir. “Back”, “Main” ve “Next” butonları ile sahneler arasında gezinmek mümkündür.

Tablo 3.1’de sahneler arasında geçişin Unity platformunda gerçekleştirilmesini sağlayan kod bloğunun bir sonraki sahneye geçişini sağlayan kesiti gösterilmiştir. Bu kesit, “SceneManager” bileşeninin “QuantumMain.cs” dosyasının bir parçasıdır.

Tablo 3.1 Sahneler arasında gezinmeyi sağlayan kod bloğu

```
public void NextScene()
{
    switch (SceneManager.GetActiveScene().buildIndex)
    {
        case 0:
            SceneManager.LoadScene("Basic_gates01");
            break;
        case 1:
            SceneManager.LoadScene("Basic_gates02");
            break;
        case 2:
            SceneManager.LoadScene("Basic_gates03");
            break;
        case 3:
            SceneManager.LoadScene("Measurement01");
            break;
        case 4:
            SceneManager.LoadScene("Measurement02");
            break;
        case 5:
            SceneManager.LoadScene("Multiqubits01");
            break;
        case 6:
            SceneManager.LoadScene("Multiqubits02");
            break;
        case 7:
            SceneManager.LoadScene("Multiqubits03");
            break;
        case 8:
            SceneManager.LoadScene("Superposition");
            break;
        case 9:
            SceneManager.LoadScene("Grover");
            break;
        case 10:
            SceneManager.LoadScene("Teleportation");
            break;
        case 11:
            SceneManager.LoadScene("Main_Page");
            break;
    }
}
```


3.1.1. Sahnedeki Nesnelerin Kontrolü

Nesneleri kontrol etmek ve çeşitli giriş donanımlarıyla etkileşimlerini ayarlamak için “XR Interaction Toolkit” aracı bir dizi yardımcı araç sunar. Sahnedeki bir nesne seçiliyken, menü çubuğunda Component → XR seçeneği üzerine gelindiğinde seçili nesneye eklenebilecek çeşitli etkileşim araçlarının bir listesi gelir. Bunlardan uygulama kapsamında en sık yararlanılmış olanlar şunlardır:

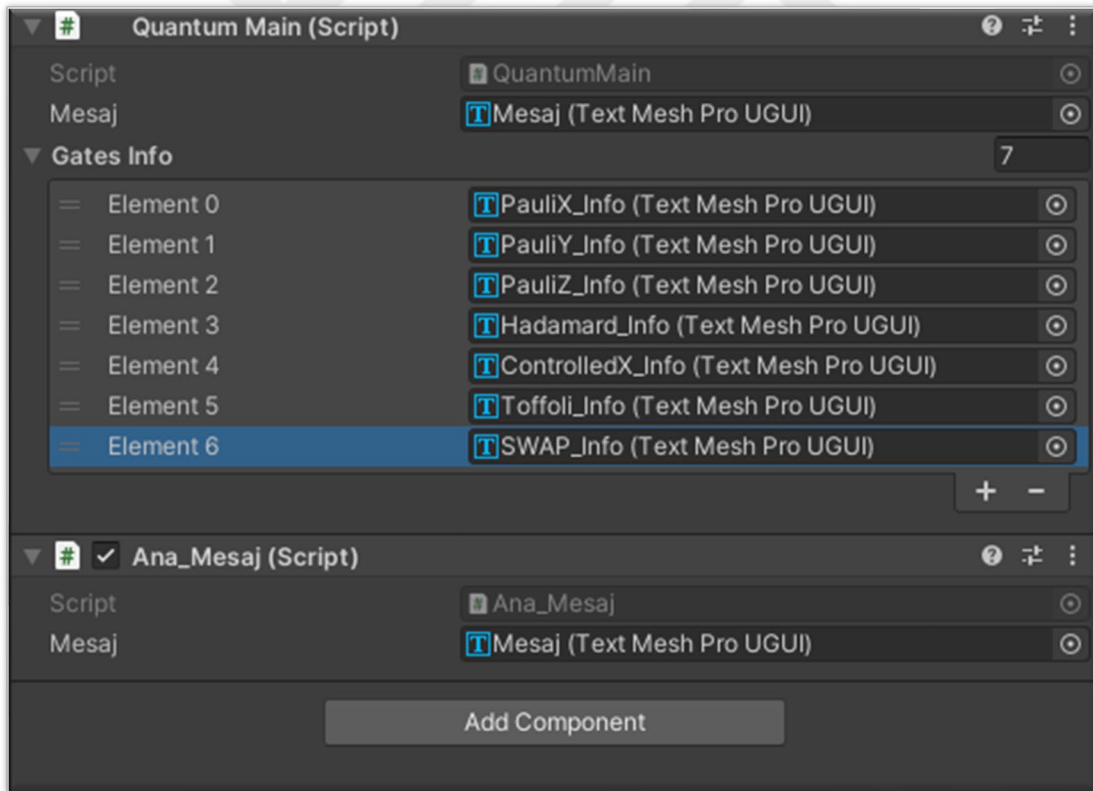
- **XR Controller:** Oculus Rift ve Quest gibi VR donanımlarının el denetleyicilerini kontrol etmek için kullanılan bileşendir. El denetleyicisinin üzerinde bulunan “Trigger (tetik anahtarı)”, “Grip (kavrama anahtarı)”, “Joystick (3 boyutlu yön kolu)” ve diğer butonların kontrollerinin ayarlanması sağlanır. Dolayısıyla bu nesne yaygın olarak XRRig aracının sağ ve sol el denetleyicilerine eklenir.
- **XR Ray Interactor:** El denetleyicileriyle kullanıldığında “XR Controller” aracıyla birlikte kullanılır. El denetleyicilerinin, ışın yardımıyla nesnelerle etkileşime girmesini sağlar. Özellikle dar ve sıkışık kapalı alanlarda adım atmak zorunda kalmadan, uzaktaki nesnelerin kontrolü için bu araç tercih edilir.
- **XR Direct Interactor:** Bu araç ta “XR Ray Interactor” aracı gibi, el denetleyicilerinin nesnelerle etkileşiminde kullanılır. Ancak bu araç adından da anlaşılacağı üzere doğrudan nesneleri tutma ve kavrama hissiyatı verecek şekilde etkileşime girmenizi sağlar. Nesnelerin kavranması ve onlarla etkileşime girilebilmesi için, kullanıcının el hareketlerini kontrol eden el denetleyicileriyle ekrandaki nesneye yaklaşması gerekir. Dolayısıyla bu araçla etkileşim kontrolü daha zordur ve daha fazla dikkat gerektirir. Uzaktaki bir nesneyi kavrayabilmek için ona yaklaşmanız gerekir. Bunun için ekran başlığıyla birlikte tanımlanmış sınır alanı içinde yürümeniz gerekir.

Not: “XR Ray Interactor” ve “XR Direct Interactor” araçları aynı nesnede birlikte kullanılamaz. Bir diğer ifadeyle, aynı sağ ve sol el denetleyicisiyle aynı anda iki araçla birlikte kontrol işlemi gerçekleştirilemez. Eğer bu iki aracın da kullanılması isteniyorsa, sağ el denetleyicisine biri, sol el denetleyicisine diğeri eklenebilir.

- **XR Grab Interactable:** El denetleyicileriyle etkileşime girmesi istenen nesnelere eklenen bir araçtır. Bu aracın eklendiği nesneler, sağ ve sol el denetleyicileriyle kavranabilir hale gelir.

- **XR Rig:** VR Başlığı ve sağ-sol el denetleyicilerinin sahneye eklenmesi istendiğinde kullanılan en önemli XR aracıdır. Sahneye ilgili VR donanımlarının eklenmesini ve kontrolünü sağlar.
- **XR Interaction Manager:** Sahnedeki tüm etkileşime girilebilir nesnelerin ve etkileşim araçlarının birbirleriyle etkileşime girmesini sağlayan ve bir bakıma aracılık işlevi gören önemli bir XR aracıdır.

Ana sayfa sahnesindeki (Bkz. Şekil 3.4) temel kapılara el denetleyicisinin ışını ile gelindiğinde ona ait bilgi ekranının görüntülenmesi için, “QuantumMain.cs” ana “script” dosyasında “GatesInfo” adıyla “TextMeshProUGUI” dizisi oluşturulmuştur. “TextMeshProUGUI” html etiketleriyle de şekillendirilebilen çok ayrıntılı bir metin düzenleme yüzeyidir. “Canvas” formatına benzer bir ara yüzle 3 boyutlu VR ortamlarında etkileşimli bilgilendirme yüzeyleri oluşturmak için kullanılır. Şekil 3.5’te “SceneManager” nesnesine eklenmiş “script” dosyalarının güncel durumu gösterilmiştir.



Şekil 3.5 "SceneManager" nesnesindeki "script" dosyaları

Şekilde, haklarında kullanıcıya bilgi verilmesi istenen bazı temel kapılar için hazırlanmış bilgi alanlarının dizi değişkenine yüklenmiş oldukları görülmektedir. Bu adımdan sonra

Tablo 3.2’de gösterildiği üzere, bu bilgi alanlarından seçilen kapıya göre aktif kılınacak olanın kontrolünü gerçekleştiren kod bloğu yazılır.

Tablo 3.2 İlgili kapıya göre bilgi ekranını gösteren kod bloğu

```
class QuantumMain : MonoBehaviour
{
    public TextMeshProUGUI mesaj;
    public TextMeshProUGUI[] GatesInfo;

    .
    .
    .

    public void OnTriggerEnter(Collider other)
    {
        foreach (TextMeshProUGUI e in GatesInfo)
        {
            e.gameObject.SetActive(false);
        }
        if (other.CompareTag("PauliX"))
        {
            GatesInfo[0].gameObject.SetActive(true);
        }
        else if (other.CompareTag("PauliY"))
        {
            GatesInfo[1].gameObject.SetActive(true);
        }
        else if (other.CompareTag("PauliZ"))
        {
            GatesInfo[2].gameObject.SetActive(true);
        }
        else if (other.CompareTag("Hadamard"))
        {
            GatesInfo[3].gameObject.SetActive(true);
        }
        else if (other.CompareTag("ControlledX"))
        {
            GatesInfo[4].gameObject.SetActive(true);
        }
        else if (other.CompareTag("Toffoli"))
        {
            GatesInfo[5].gameObject.SetActive(true);
        }
        else if (other.CompareTag("SWAP"))
        {
            GatesInfo[6].gameObject.SetActive(true);
        }
        else
        {
            mesaj.text = "NOT SELECTED...";
        }
    }
}
```

Dikkat edileceği üzere karşılaştırmalar ilgili kapı nesnesinin “Tag” olarak adlandırılan etiket bilgisi üzerinden yapılmaktadır. Unity’de her nesneye bir etiket adı verilebilir ve nesneler varsayılan olarak “Untagged”, bir diğer ifadeyle etiketsizdir. Etiketleme ilgili

nesneyi diğer nesnelerden ayırmayı sağlar. Böylece sahne içinde belli etiketlere sahip nesne ya da nesne gruplarını kontrol etmek daha kolay olur.

3.1.2. Sahnedeki Nesnelerde Fizik Yasaları

Nesneleri VR donanımlarıyla seçmek ya da onların üzerinde diğer fizik eylemlerini gerçekleştirmek için “XR Grab Interactable” aracı kullanılır. Kullanıcıların sahnede yer alan kuantum bilgisayarını yakından görebilmeleri için “XRGrabInteractable.cs” sınıf dosyası hiyerarşi menüsü içindeki “quantum_computer” oyun nesnesinin üzerine sürüklenerek bırakılır ya da “Add Component” butonuyla eklenir.

Nesnenin seçilebilmesi ya da diğer fizik işlemlerinin yapılabilmesi için “Rigidbody” ve “Collider” bileşenlerinin de eklenmesi gerekir. “XR Grab Interactable” aracıyla birlikte “Rigidbody” bileşeni de otomatik olarak gelir. VR platformu çalıştırıldığında, Newton fizik yasalarının işletilmesini engellemek için ya da düzenlemek için “Collider” ve “Rigidbody” bileşenlerinin parametrelerinde bir dizi değişiklik yapmak gerekir.

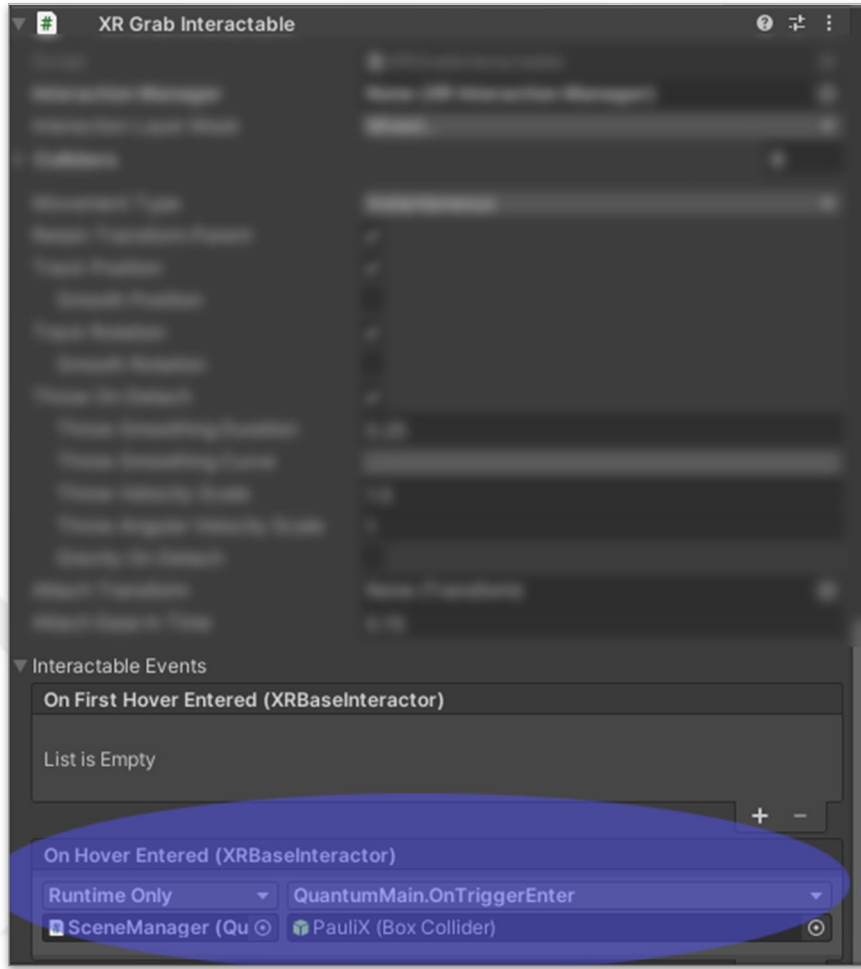
“Rigidbody” bileşeni, her nesne için yer çekim ivmesi ve kütle ağırlığı tanımlamayı sağlar. Ayrıca eklendiği nesnenin X-Y-Z düzlemlerinde dönme ve konum hareketlerini de sınırlamada kullanılır. “Is Kinematic” seçeneği yardımıyla, kuvvet etkisi ya da çarpışma gibi fiziksel eylemlerden hariç tutulup tutulmayacağı belirlenir. “Use Gravity” seçeneği, yer çekim ivmesinin kullanılıp kullanılmayacağını belirler. Yer çekim ivmesi etkinleştirildiğinde ve konumu serbest bırakıldığında ilgili nesne havadaysa ve destek aldığı bir dayanak noktası yoksa yere düşme eylemi gerçekleştirir.

“Collider” bileşeni eklendiği nesnenin özelliğine göre farklı şekil ve isimler alır. Örneğin sahneye eklediğiniz nesne 3 boyutlu bir küp ise bu bileşenin adı “Box Collider”, eğer küreyse “Sphere Collider” adını alır. Eklendiği nesnenin etrafında, adının çağrıştırdığı şekle uygun görünmez bir etkileşim alanı oluşturur. Eğer “Is Trigger” seçeneği işaretlenirse, diğer nesnelerle fiziksel bir etkileşime girmez. Bu durumda diğer nesnelerin içinden geçebilir. Gerçek bir nesne etkisinin istendiği durumlarda “Is Trigger” seçeneği işaretsiz bırakılır.

“Collider” bileşenlerinin eklendiği nesneye göre konumunu ve boyutunu ayarlamak mümkündür. Böylece nesnenin tam olarak kapladığı hacimden farklı yüzeylerde etkileşim alanı oluşturulabilir.

Unity platformunda tüm etkileşebilir (interactable) sınıf dosyalarının (nesnelerin) bir dizi etkileşebilir eylemleri vardır. Açıklandığı üzere, girdi aygıtlarıyla etkileşime girmesini istediğimiz sahne nesnelere eklediğimiz “XR Grab Interactable” bileşenin de bir dizi eylemi vardır. Örneğin, sağ ya da sol el denetleyicisinin sahne üzerindeki bir kuantum kapısının üzerine geldiğinde ilgili interaktif bilgilendirme mesajının etkinleşmesini istiyorsak, “On Hover Entered” eylemine, bu işlemi gerçekleştirecek yöntemi bildirmemiz gerekir.

Bir önceki başlıkta gösterilen “OnTriggerEnter” yönteminin kullanımı, örnek uygulamada “On Hover Entered” eylemine bu yöntemin yüklenmesiyle gerçekleştirilmiştir. Bunun için Şekil 3.6’da gösterildiği üzere, örnek bir kuantum kapısı nesnesi seçilir ve “Inspector” penceresinde “XR Grab Interactable” bileşenine gidilir. Burada “Interactable Events” başlığı altındaki “On Hover Entered (XRBasedInteractor)” eyleminde “List is Empty” uyarısının yazılı olduğu kısımda (+) işaretine tıklanarak yeni bir “Runtime Only” alanı açılır.



Şekil 3.6 "XR Grab Interactable" bileşenin "On Hover Entered" eylemine işletilecek bir yöntem eklenmesi

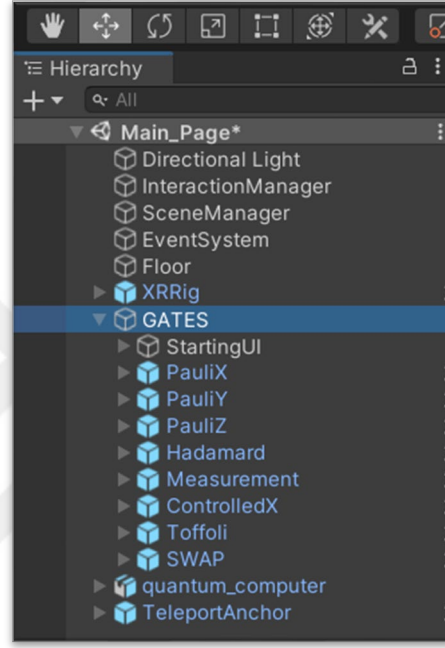
Bu alana sahne hiyerarşisindeki “SceneManager” nesnesi sürüklenerek bırakılır. Ardından “QuantumMain” isimli sınıf dosyası ve onun içindeki “OnTriggerEnter” yöntemi seçilir. Bu yöntemin “Collider” isimli bir girdisi vardır. Girdi alanına seçimi yapılmış olan ilgili kuantum kapısı seçilir ve sürüklenerek bırakılır. Bu işlem, hakkında farklı bilgi mesajı verilmesi istenen her bir nesne için tekrarlanır. Böylece nesneleri fiziksel olarak kavranmasının yanında duruma göre farklı işlemler de yapılabilir.

3.1.3. Dinamik Değişken Bilgi Mesajlarının Tasarımı

Sanal gerçeklik setinin el denetleyicileri ile sahnedeki nesnelerin “Collider” alanlarıyla sınırlanmış bölgelerine uzaktan ya da yakından temas edilerek görünür kılınan dinamik değişkenli mesajların nasıl etkinleştirildiği bir önceki başlık altında anlatılmıştı. “Main Page” isimli ana sayfada yer alan kapılardan örnek olarak, “Pauli-X”, “Pauli-Y”, “Pauli-

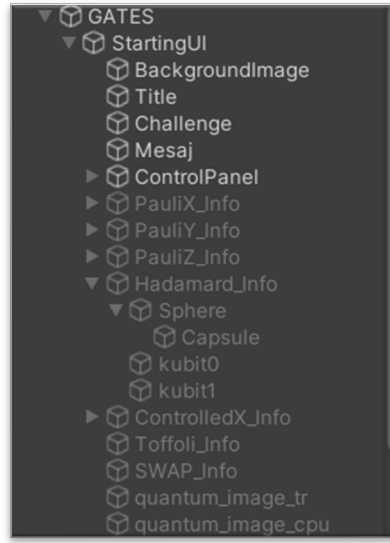
Z”, “Hadamard” ve “Controlled-X (CNOT)” kapıları için örnek dinamik deęiřken mesajlar hazırlanmıřtır.

Öncelikle bu iřlem için, řekil 3.7’de gösterildięi gibi “Hierarchy” alanında “GATES” isimli boş bir “GameObject” nesnesi oluřturulmuř ve o nesnenin altında “StartingUI” isimli büyük bir “Canvas” oluřturulmuřtur. Bu alıřma tuvali tüm bilgilendirme mesajlarını, kontrol nesnelerini ve resimleri barındırır.



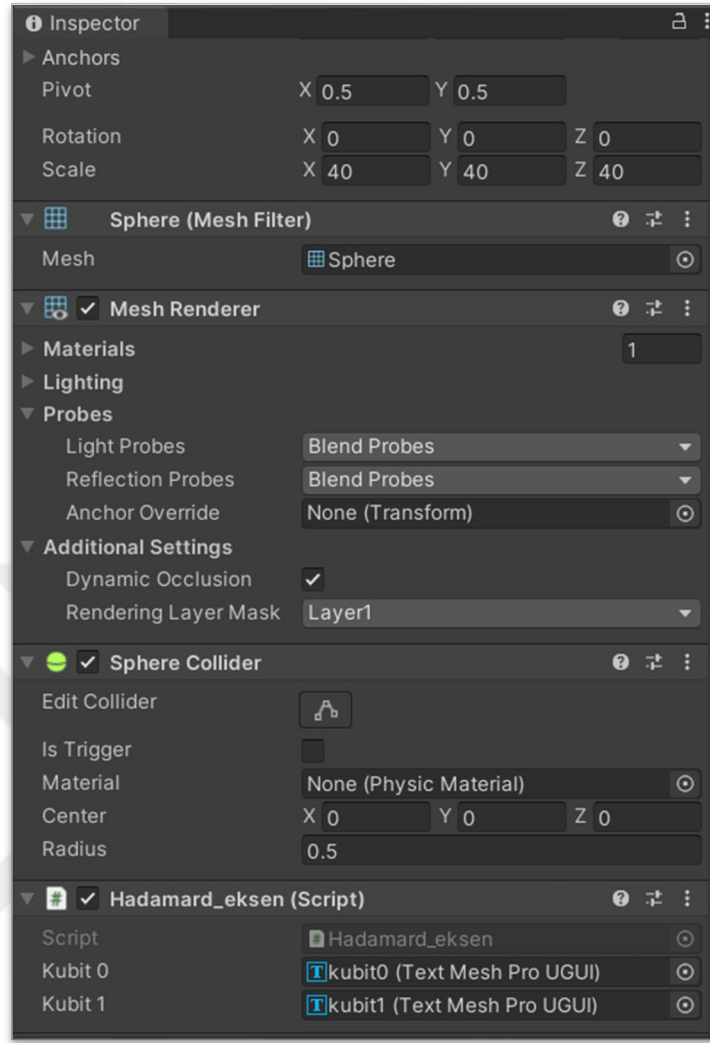
řekil 3.7 Hakkında dinamik deęiřken bilgi mesajı verilen kuantum kapılarının tasarım hiyerarřisindeki konumları

Sol ya da saę el denetleyicisinin, “Raycast” ışını ile üzerine gelinen kapı hakkında verilen dinamik deęiřkenli bilgilendirme mesajları “StartingUI” isimli tuvalin içinde yer alır. Bir önceki başlıkta anlatılan metin temelli mesajlara ek olarak, bu kısımda kübitlerin üzerinde ilgili kapının nasıl etki gösterdiğini hareketli ve animasyonlu olarak gösteren dinamik Bloch küresi ile bilgilendirme yapılmaktadır. “StartingUI” isimli tuvalin içinde hangi nesnelerin olduęunu gösteren daha ayrıntılı bir hiyerarři resmi řekil 3.8’de gösterilmiřtir. řekilde örnek olarak, hakkında dinamik řekilde bilgilendirme yapılan “Hadamard” kapısına ait bilgi nesneleri de gösterilmiřtir.



Şekil 3.8 "StartingUI" tuvalinin ayrıntılı hiyerarşisi

Her bir kuantum kapısının kubitlere olan etkisinin Bloch küresi üzerinde dinamik olarak gösterilmesi için, hiyerarşi içinde “Sphere” olarak isimlendirilmiş küre oyun nesnesine dinamik bilgilendirme içeriğini üreten bir C# kodu yazılmıştır. Küre nesnesinin altında, “Capsule” adıyla eksen oluşturacak oyun nesnesi ve “kubit0”, “kubit1” isimli ilgili kubit bilgisini ortamda görmemizi sağlayacak “TextMeshProUI” nesneleri görülmektedir. Şekil 3.9’da “Hadamard_Info” isimli bilgilendirme nesnesinin alt nesnesi olan “Sphere” nesnesi seçildikten sonra “Inspector” penceresindeki öğeleri görülmektedir.



Şekil 3.9 “Sphere” isimli alt nesnenin öğeleri

Tablo 3.3’te, Şekil 3.9’da görülen “Hadamard_eksen” isimli C# kodu gösterilmiştir. Bu kod yardımıyla Hadamard kapısının $|0\rangle$ ya da $|1\rangle$ kubitleri üzerinde nasıl bir etkisi olduğu canlandırılmaktadır. Benzer işlem haklarında bilgi verilecek diğer tüm kuantum kapıları için de yapılmıştır.

Tablo 3.3 Hadamard kapısının canlandırılmasını sağlayan C# kodu

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

public class Hadamard_eksen : MonoBehaviour
{
    public TextMeshProUGUI kubit0;
    public TextMeshProUGUI kubit1;
}
```

```

bool ters = false;
float timer = 0f;
int sayac = 0;
void Update()
{
    timer += Time.deltaTime;
    if (timer < 1 && !ters)
    {
        transform.Rotate(Time.deltaTime*120, -Time.deltaTime * 120, 0f);
    }
    if (timer < 1 && ters)
    {
        transform.Rotate(Time.deltaTime * 120, Time.deltaTime * 120, 0f);
    }
    if (timer >= 1 && sayac==0)
    {
        kubit0.text = "0.707|0> + 0.707|1>";
        kubit1.text = "|1>";
    }
    if (timer >= 1 && sayac == 1)
    {
        kubit0.text = "|0>";
        kubit1.text = "0.707|0> - 0.707|1>";
    }
    if (timer > 2)
    {
        kubit0.text = "|0>";
        kubit1.text = "|1>";
        timer = 0;
        ters = !ters;
        sayac++;
        if (sayac > 1) sayac = 0;
        if (!ters) transform.rotation = Quaternion.Euler(0, 0, 0);
        if (ters) transform.rotation = Quaternion.Euler(-180, 0, 0);
    }
}
}

```

Ayrıca yine ana sayfada yer alan 3 boyutlu kuantum bilgisayar modeli için de resimli bilgilendirme mesajları hazırlanmıştır. Kullanıcı 3 boyutlu kuantum bilgisayarı modelinde daha ayrıntılı görmek istediği ilgili alanlara, “Raycast” ışınını yaklaştırdığında “Image” alanında ilgili bilgilendirme resmi gösterilmektedir.

Unity’de içinde resim türü dosyaları göstermek için kullanılan “Image” UI nesnesi bulunmaktadır. Örnek uygulamada kuantum bilgisayarının, “Transistör”, “CPU” ve “Bobin” gibi unsurları hakkında resimli bilgi veren alanlar oluşturulmuştur. Kapılar için gerçekleştirilen tasarım uygulamasının benzeri uygulanmış olup, farklı olarak “TextMeshPro” UI nesnesi yerine “Image” UI nesnesi kullanılmıştır.

3.2. Temel Kuantum Bilişimi Uygulamalarının Tasarımı

VR platformunda kullanıcılar, temel ve basit kuantum kapısı uygulamalarını tasarlayabilir ve simüle edebilir. VR temelli eğitim platformunun tasarımında, temel kapılarda işlemler, çok kübitli kapılar, ölçme işlemleri, Süperpozisyon işlemi, Grover algoritması ve Teleportasyon işlemi olarak kuantum bilişimi hakkında temel bilgileri verecek şekilde bir yol belirlenmiştir. Tasarım iki kübit üzerinden olacak şekilde hazırlanmıştır.

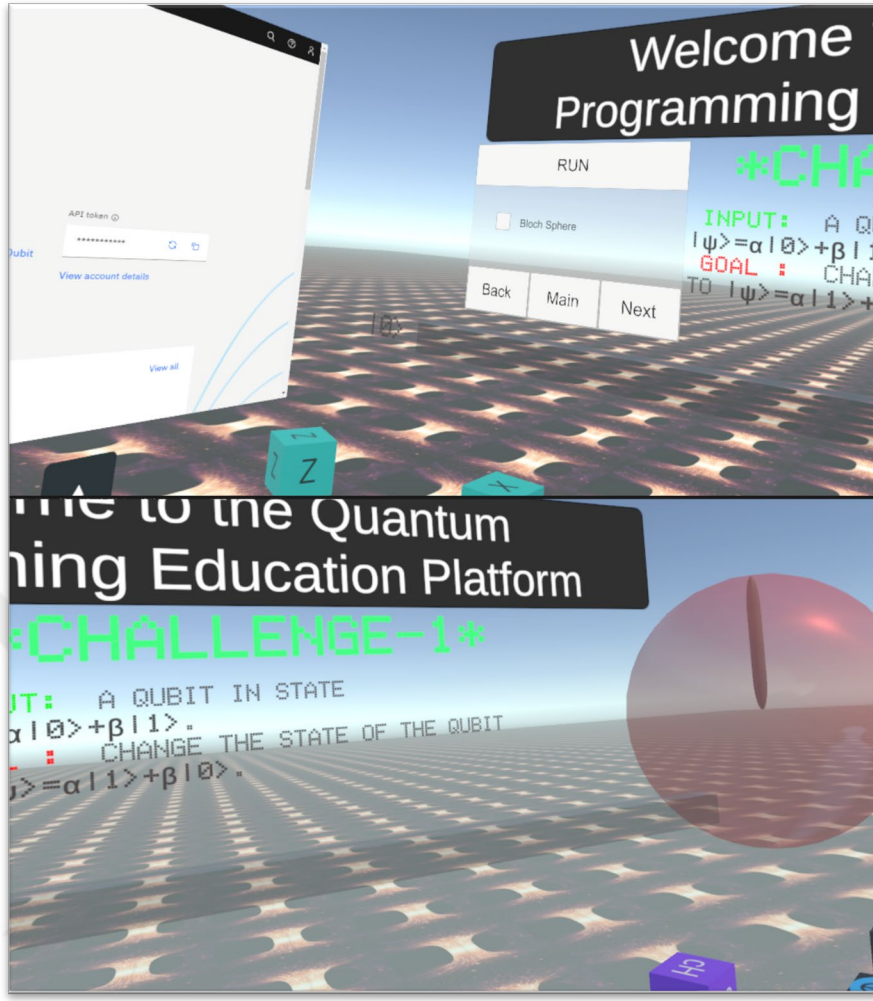
3.2.1. Temel Kuantum Bilişimi Uygulamalarında Görsel Tasarım

Bu örnekte kullanıcı tek kübitlik bir hat üzerinde işlem yapmakta ve kullanıcıdan $|0\rangle$ kübitini $|1\rangle$, $|1\rangle$ kübitini $|0\rangle$ yapacak kapıyı hattın uygun alanına yerleştirilmesi istenmektedir. Eğer kullanıcı Pauli-X kapısını seçer ve sürükleyerek kenetlenme alanına bırakırsa “Challenge-1” mesajının yerini “Success” mesajı alır. Böylece kullanıcıya doğru kapıyı seçtiği belirtilir.



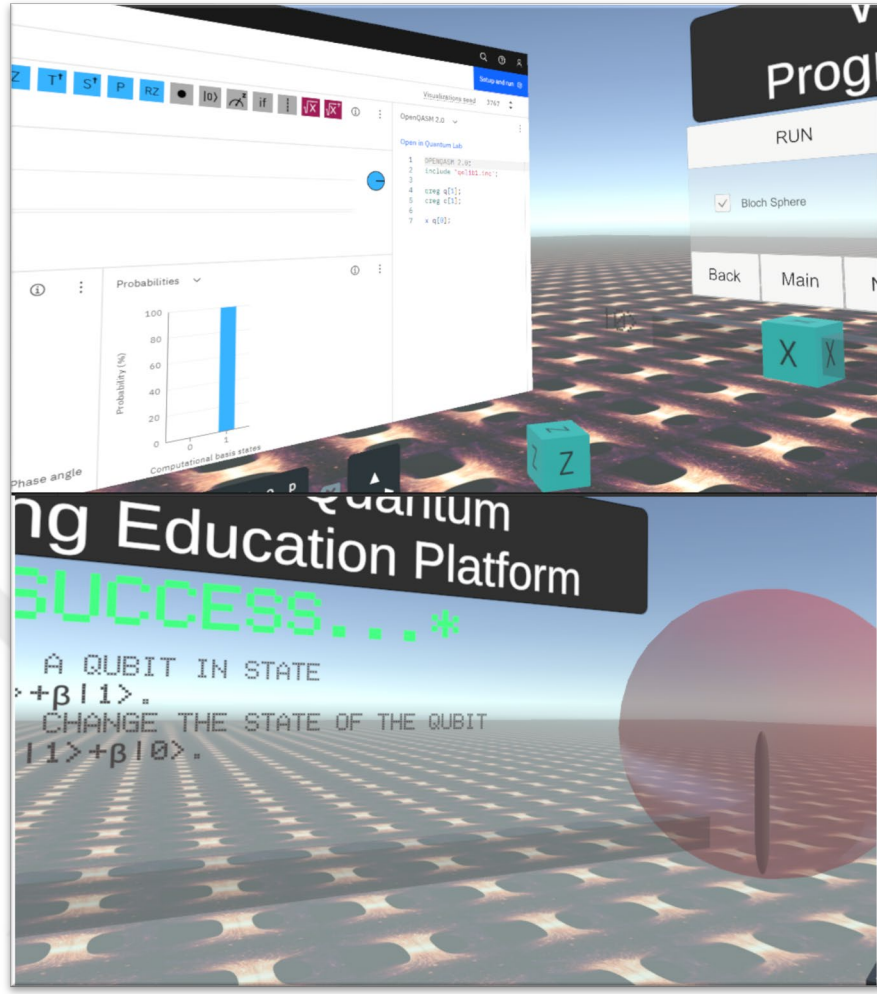
Şekil 3.10 VR platformunun örnek görünümü

Şekil 3.11’de sırasıyla boş kübit hattı ve Bloch küresinde kübitin evresi gösterilmiştir. Daha önceden söylendiği üzere “IBM Quantum Composer” platformunda kübitler $|0\rangle$ evresine alınarak işlem yapılmaktadır. Ancak bu VR temelli eğitim platformunda her bir kübit hattının başında işleme alınan kübit evresi, $|0\rangle$ ya da $|1\rangle$ olacak şekilde rastgele olarak yüklenmektedir.



Şekil 3.11 Temel kapı uygulamalarında örnek bir başlangıç ekranı

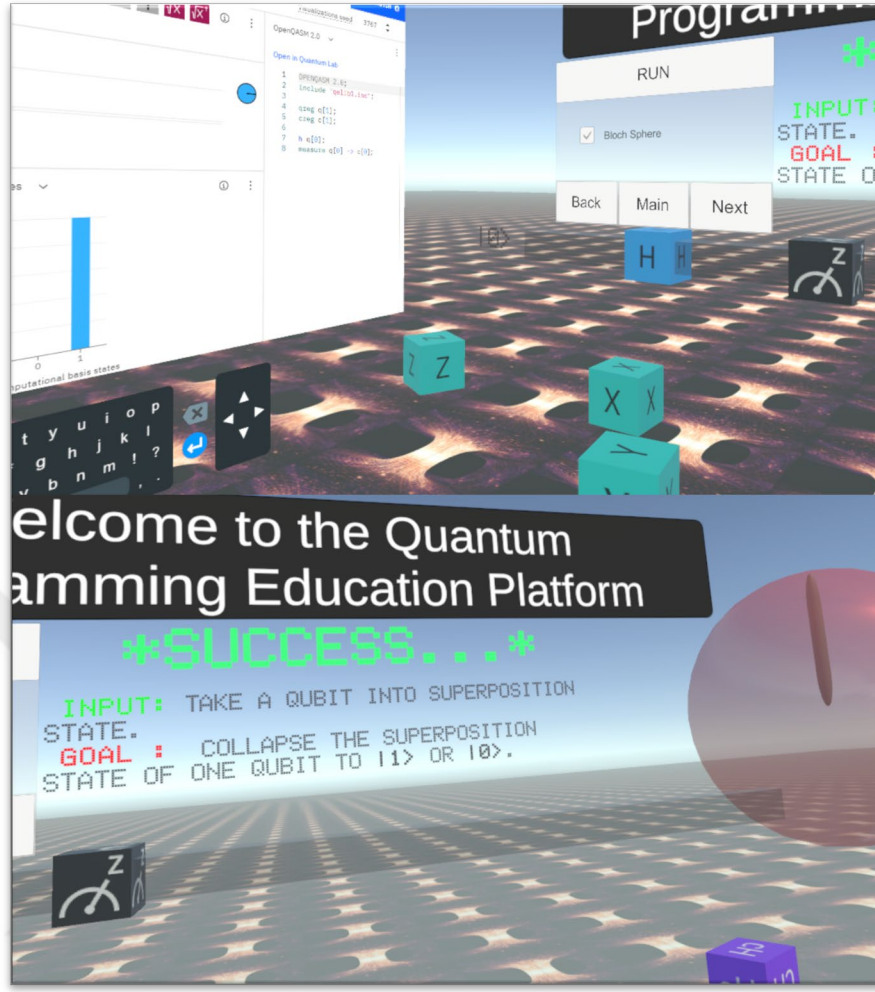
Verilen talimata uygun şekilde ilgili kapı kenetlenme alanına sürüklenip bırakıldığında ekranda “Success...” ifadesi belirir. Bu durumdayken “RUN” butonuna basılırsa VR ortamda bulunan 3 boyutlu VR tarayıcısında ilgili kuantum besteleycisi sayfasına gidilir. İşlemin gerçek çıktısı web sayfası üzerinden görüntülenir. Bu esnada, uygulanan kapının etkisine göre kübit evresinin değişimi de VR ortamdaki Bloch küresinde gösterilmektedir.



Şekil 3.12 Temel kapı uygulamalarında doğru işlemin gerçekleştirilmesi örneği

Örnek bir ölçme uygulamasında kullanıcıdan bir kübiti süperpozisyona getirmesi ve ardından ölçme işlemini gerçekleştirmesi istenmektedir. “IBM Quantum Composer” platformunda kuantum simülatörü ile çalışılırken, süperpozisyona alınan kübit ya da kübitlerin ilgili kübitine denk gelen ölçme operatörü sonucu her zaman “1” evresi elde edilmektedir. Gerçek kuantum bilgisayarlarında hata düzeltmesi ve kuantum kuvvetlendirmesi olarak adlandırılan bir teknik sonucu, çıktı olasılığı çok yüksek oranda “1” evresini sağlayacak şekilde işlem gerçekleşmektedir.

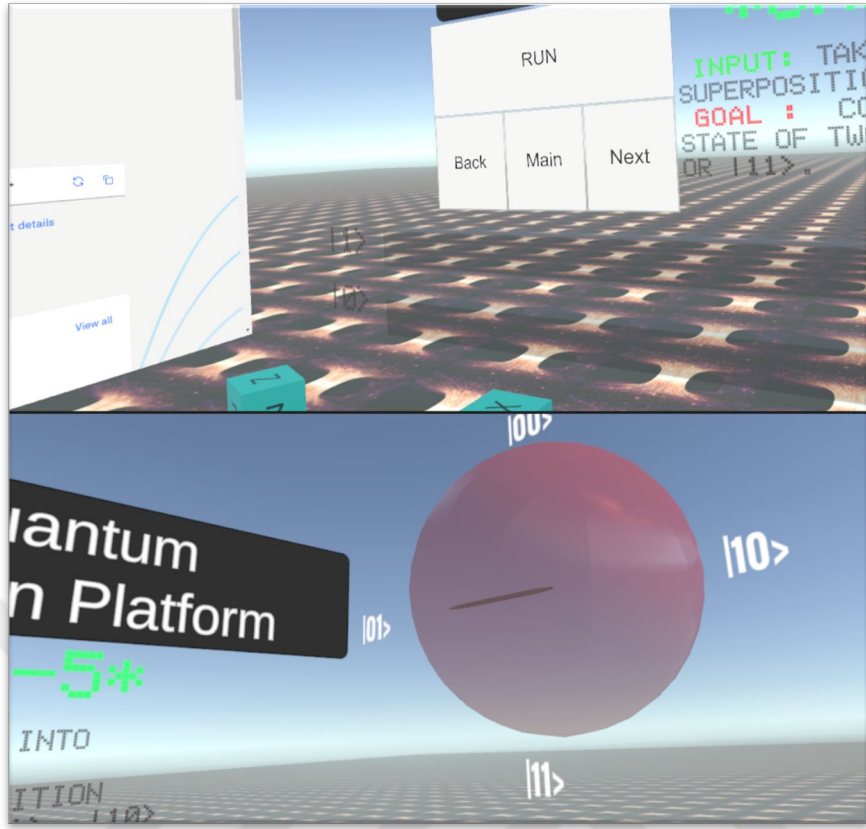
Bu VR temelli simülatör uygulamasında ise ölçme işleminin sonucu rastgele olarak “0” ya da “1” evresini eşit olasılıkla verecek şekilde işlem yapılmaktadır. Şekil 3.13’te temel bir ölçme işleminin sanal gerçeklik ortamındaki ekran çıktıları gösterilmiştir.



Şekil 3.13 Temel kapı uygulamalarında ölçme işlemi örneği

Şekilde görüldüğü üzere, VR ortamında da başlangıçta $|0\rangle$ kübiti ile açılış yapılmış ve ölçme işlemi sonucu süperpozisyon durumu çökerek yine $|0\rangle$ evresi ölçülmüştür. Ancak IBM kuantum besteleycisinin simülatöründe, öncesinde söylendiği üzere $|1\rangle$ ölçümü gerçekleştirilmiştir.

Çok kübitli uygulamada ise iki kübitin süperpozisyona getirilmesi ve süperpozisyon durumlarının ölçümü işleminin yapılması istenmektedir. Bu uygulamada diğerlerinden farklı olarak tüm kapılar matematiksel olarak modellenmiştir. Kapıların matris formları üzerinden gerçekleştirilen matematiksel işlemlere göre sonuç elde edilmektedir. Kapıların matematiksel olarak işleyişi bir sonraki başlık altında açıklanmıştır. Şekil 3.14'te iki kübitlik sistemin işleme hazır hale gelmesi ve kuantum evresinin Bloch küresi üzerindeki gösterimi görülmektedir. Örnek uygulama sırasında görüldüğü üzere, girişte $|01\rangle$ evresi görülmektedir.

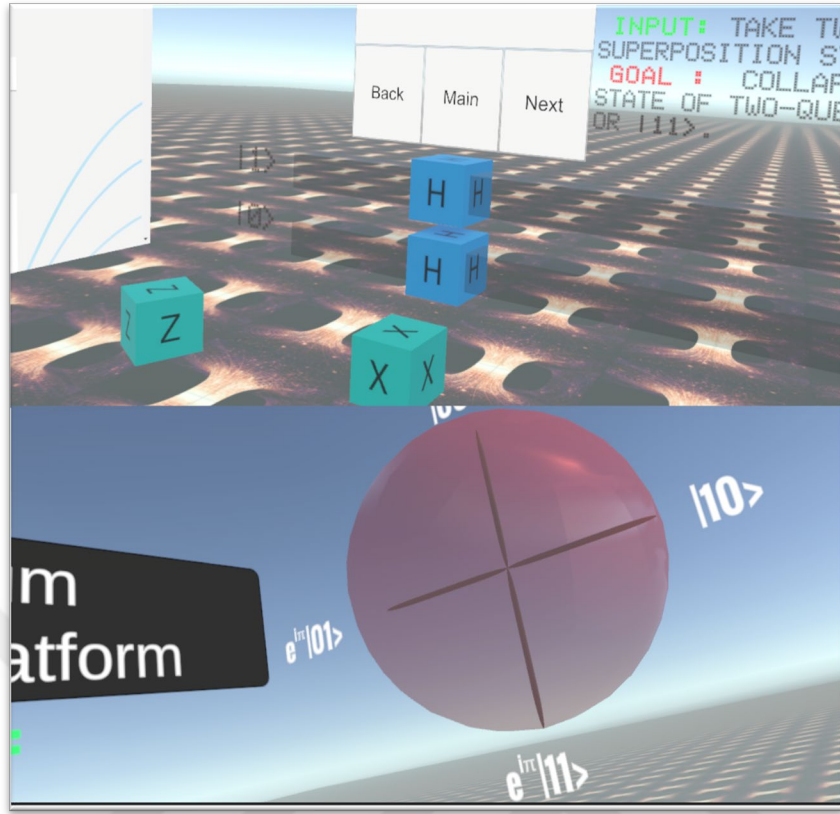


Şekil 3.14 İki kubitlik sistemde örnek bir başlangıç durumu

Kübit0 ve kübit1 hatlarına, “Challenge-5” başlığı altında istenen işleme uygun olarak Hadamard kapıları sürüklenip bırakıldığında aşağıda gösterildiği şekilde 4 evreli bir süperpozisyon durumu ortaya çıkar.

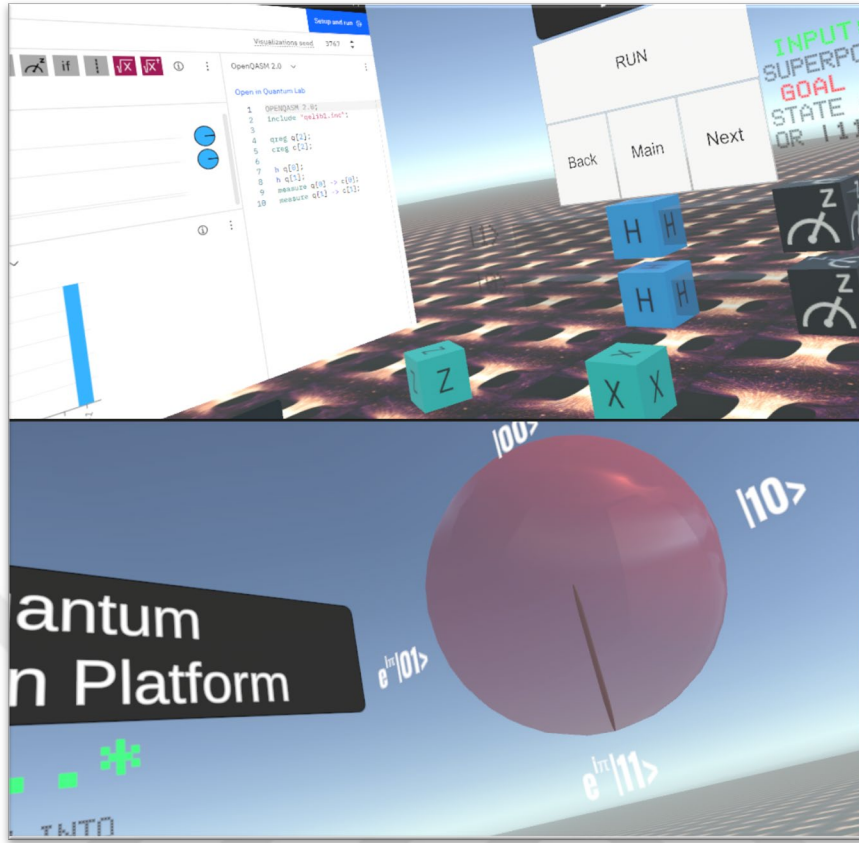
$$|\psi\rangle = |00\rangle + e^{i\pi}|01\rangle + |10\rangle + e^{i\pi}|11\rangle$$

Burada, $e^{i\pi}$ 180°'lik faz sapmasını ifade etmektedir. Kübitlerin evresinde bir değişim olmamıştır ancak faz kayması gerçekleşmiştir.



Şekil 3.15 Hadamard kapıları yerleştirildikten sonra elde edilen süperpoze durumu

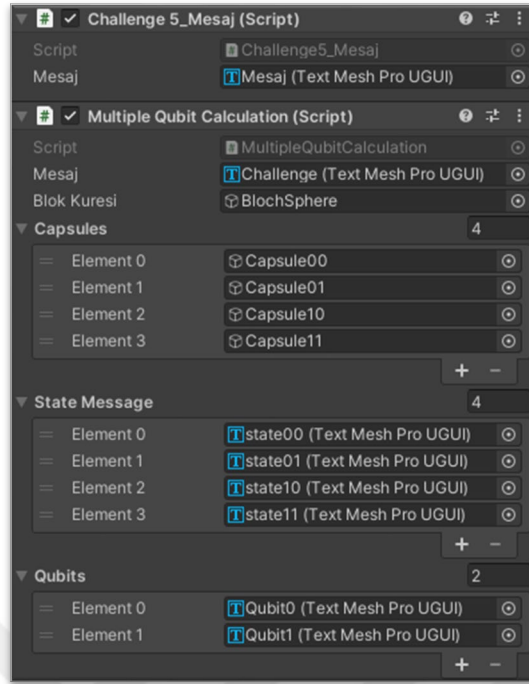
Ölçme işlemini gerçekleştirecek operatörler de ilgili alanlara yerleştirildiğinde, IBM kuantum besteleyicisinin simülátörünün çalışma ilkesine benzer şekilde bir ölçme işlemi gerçekleştirilmekte ve işlemin sonucunda $|11\rangle$ evresi ölçülmektedir. Ölçme işleminin çıktısı Şekil 3.16’da görülmektedir.



Şekil 3.16 İki kubitlik bir sistemde ölçme işlemi ve süperpoze durumunun çökmesi

Doğru kapı kombinasyonu gerçekleştirildiğinde “RUN” butonuna basılmasıyla birlikte IBM kuantum besteleycisindeki devre de çalıştırılır ve iki kubitlik süperpozisyon evresi için $|11\rangle$ ölçme değeri elde edilir. VR platformda gerçekleştirilen canlandırmada kubitlerin başlangıç değerleri rastgele olarak gelmektedir.

Kubitlerin rastgele olarak durumlarının belirlenmesi ve evrelerin Bloch küresinde gösterimi bir dizi değişkenin kontrolüyle gerçekleştirilmektedir. Çok kubitli ölçme işlemi için bu işlemin nasıl yapıldığına bakacak olursak, Şekil 3.17’de “SceneManager” nesnesi üzerinde yer alan “script” dosyaları ve değişkenler görülmektedir. “Capsules” isimli “GameObject” oyun nesnesi dizi formatındadır ve iki kubit için var olan 4 evrenin eksenini göstermek için kullanılmaktadır. “State Message” ismiyle gözüken nesne ise yine 4 elemanlık bir dizi olarak tanımlanmış “TextMeshProUI” nesnesidir. Bu değişken ile Bloch küresi üzerinde görülen evreler yazdırılır. “Qubits” ismiyle gözüken değişken ise kubit hatlarının başında oluşturulan kubit evrelerini tutan değişkendir.



Şekil 3.17 Çok kübitli ölçme işleminde değişkenlerin ayarlanması

Çoklu kübit hesaplaması işlemi için kullanılan kübitlerin rastgele olarak oluşturulması, bunların gösterimi ve Bloch küresinin başlangıç koşuluna ayarlanması işlemlerini gerçekleştiren kod örneği Tablo 3.4'te gösterilmiştir.

Tablo 3.4 Çok kübitli işlemde kübitlerin rastgele olarak ayarlanması

```

...
public TextMeshProUGUI mesaj;
public GameObject blokKuresi;
public GameObject[] Capsules;
public TextMeshProUGUI[] stateMessage;

public TextMeshProUGUI[] qubits;

int rand = 0;
int rand2 = 0;
int toplam = 0;
string[] states = { "|00>", "|01>", "|10>", "|11>" };

async void Start()
{
    SocketMultiFirstTag.firstTag = false;
    SocketMultiSecondTag.secondTag = false;
    SocketMultiThirdTag.thirdTag = false;
    SocketMultiFourthTag.fourthTag = false;
    foreach (GameObject game in Capsules)
    {
        game.gameObject.SetActive(false);
    }
    foreach (TextMeshProUGUI states in stateMessage)
    {

```

```

        states.gameObject.SetActive(true);
    }

    rand = Random.Range(0, 2);
    rand2 = Random.Range(0, 2);
    if (rand == 0 && rand2 == 0)
    {
        qubits[0].text = "|0>";
        qubits[1].text = "|0>";
        QubitLine0 = Qubit0;
        QubitLine1 = Qubit0;
    }
    else if (rand == 1 && rand2 == 0)
    {
        qubits[0].text = "|1>";
        qubits[1].text = "|0>";
        QubitLine0 = Qubit1;
        QubitLine1 = Qubit0;
    }
    else if (rand == 0 && rand2 == 1)
    {
        qubits[0].text = "|0>";
        qubits[1].text = "|1>";
        QubitLine0 = Qubit0;
        QubitLine1 = Qubit1;
    }
    else
    {
        qubits[0].text = "|1>";
        qubits[1].text = "|1>";
        QubitLine0 = Qubit1;
        QubitLine1 = Qubit1;
    }
}

```

Kapıların sürüklenip bırakıldığı hatlarda doğru kapının kenetlenmesini takip eden ve her bir kenetlenme soketi için ayrı ayrı yazılmış kodlar vardır. Örnek olarak açıklaması yapılan uygulamada, 2’si “Hadamard”, 2’si “Measurement” olmak üzere toplamda 4 adet kapı kullanılmaktadır. Dolayısıyla 4 ayrı C# kodu hazırlanmış ve her birinin doğru değer döndürmesi sonucu “Success...” mesajı elde edilmektedir.

Tablo 3.4’deki koda tekrar dikkat edilecek olunursa, “SocketMultiFirstTag”, “SocketMultiSecondTag”, “SocketMultiThirdTag” ve “SocketMultiFourthTag” sınıf dosyalarından sırasıyla “firstTag”, “secondTag”, “thirdTag” ve “fourthTag” isimleriyle döndürülen “bool” tipinde değişkenler görülmektedir. Bu değişkenlerin 4’ünün de “true” döndürmesiyle “RUN” butonu aktif hale gelir.

3 Boyutlu “Vuplex WebView” aracı, kullanıcıların dosyayı önceden tanımlanmış dosya yoluna indirmesini sağlar. Böylece VR başlıklar aracılığıyla dosya indirmek mümkündür. Kullanıcı, QASM formatındaki besteleyici dosyasını yerel bilgisayara indirebilir ve bu dosya içeriğini VR platformuna yükleyebilir.

Vuplex 3 boyutlu web aracında dosya indirme işleminin yapılabilmesi için, “MonoBehaviour” davranışına sahip sınıf dosyalarının temel fonksiyonlarından olan ve genelde “public” ya da “private” olarak ayarlanan “Start()” fonksiyonu “async” olarak ayarlanmalıdır. Orijinal web aracında dosya indirme ve açılan sayfada geriye gitme özellikleri aktif değilken, yapılan düzenlemeler ile dosya indirme özelliği kazandırılmış ve geçici bellekte depolanan eski sayfalara erişebilme özelliği etkinleştirilmiştir. Tablo 3.5’te dosya indirme özelliği kazandırılmış web aracının sahnenin ana dosyasında etkin kılınmasını sağlayan uygulama kodu gösterilmiştir. Koda dikkat edilecek olursa “Google” standart klavyesi de etkinleştirilmiştir. Klavyenin üzerindeki yön tuşlarından “Sol Ok” tuşuna el denetleyicisi ile tıklandığında bir önceki sayfaya gitmek de mümkündür.

Tablo 3.5 3 Boyutlu web aracının sahneye eklenmesi ve ön ayarlarının yapılması

```
class MultipleQubitCalculation : MonoBehaviour
{
    CanvasWebViewPrefab _canvasWebViewPrefab;
    CanvasKeyboard _keyboard;
    ...
    async void Start()
    {
        var canvas = GameObject.Find("Canvas");
        _canvasWebViewPrefab = CanvasWebViewPrefab.Instantiate();
        _canvasWebViewPrefab.transform.SetParent(canvas.transform, false);
        await _canvasWebViewPrefab.WaitForInitialized();
        _canvasWebViewPrefab.WebView.LoadUrl("https://quantum-
computing.ibm.com/");
        var webViewWithDownloads = _canvasWebViewPrefab.WebView as
IWithDownloads;

        string path = @"C:\Users\HAKANGENC\Downloads\circuit.qasm";
        if (webViewWithDownloads != null)
        {
            webViewWithDownloads.SetDownloadsEnabled(true);
            webViewWithDownloads.DownloadProgressChanged += (sender, eventArgs)
=> {
                Debug.Log(
                    $"DownloadProgressChanged:
                    Type: {eventArgs.Type},
                    Url: {eventArgs.Url},
                    Progress: {eventArgs.Progress},
                    FilePath: {eventArgs.FilePath},
                    ContentType: {eventArgs.ContentType}"
                );
                if (eventArgs.Type == ProgressChangeType.Finished)
                {
                    Debug.Log("Download finished");
                    File.Move(eventArgs.FilePath, path);
                }
            };
        }
    }
}
```

```

_keyboard = CanvasKeyboard.Instantiate();
_keyboard.transform.SetParent(canvas.transform, false);
_keyboard.InputReceived += (sender, eventArgs) => {
    if(eventArgs.Value == "ArrowLeft")
    {
        _canvasWebViewPrefab.WebView.GoBack();
    }
    else
    {
        _canvasWebViewPrefab.WebView.HandleKeyboardInput(eventArgs.Value);
    }
};
_positionPrefabs();
}
...
}

```

3.2.2. Temel Kuantum Bilişimi Uygulamalarında Matematik

Kuantum bilişiminde birtakım işlemleri daha iyi anlamak için kompleks sayı aritmetiğine, vektör ve matris aritmetiğine hâkim olmak gerekmektedir. Uygulamanın tasarımı sırasında C# kodlama platformu yardımıyla gerçekleştirilen bir dizi matematiksel işlem bu başlık altında sunulmuştur.

2.2.2. Ölçme işlemleri başlığında anlatılan ve kübit evrelerinin muhtemel olasılık dağılımını veren genliklerin, tamsayı olarak kabul edilmesi durumunda, $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ evresinin genliklerinin rassal olarak dağılımını simüle edecek algoritma Tablo 3.6'daki gibi uygulanabilir.

Tablo 3.6 İki kübitlik sistem için rassal genlik değeri üretimi

```

// İki kübitlik sistem için 4 rassal genlik değeri üretme algoritması
int tamam = 0;
double genlik = 0.0f;
double rast = 0.0f;
List<double> rastDouble = new List<double>();
rastDouble.Add(Random.Range(-1.0f, 1.0f));
genlik = genlik + rastDouble[tamam] * rastDouble[tamam];
while (tamam<2)
{
    rast = Random.Range(-1.0f, 1.0f);
    genlik = genlik + rast*rast;
    if (genlik < 1.0f)
    {
        rastDouble.Add(rast);
        tamam++;
    }
    else
    {
        genlik = genlik - rast * rast;
    }
}

```

```

    }
    rast = Math.Sqrt(1 - genlik);
    rastDouble.Add(rast);

```

Burada genlik değerlerini “Double” değişken türünde tutacak bir “List” veri tipi tanımlanır. İlk rassal değer (-1.0f – 1.0f) aralığında alınır. İlk rassal değer |1.0| değerinden küçük olacağından karesi de |1.0| değerinden küçük olacaktır. Üretilen bu ilk olasılık genliği “a” değeridir. Bundan sonraki adımda sırasıyla “b” ve “c” olasılık genlikleri üretilcektir. Algoritmanın temel mantığı, her bir adımda genlik değerlerinin karelerinin toplamalarının alınması ve 1.0 değerinden küçük olup olmadığının kontrolüne dayanmaktadır. Eğer son rassal değer elde edilmesi sonrası yeniden hesaplanan genlik değeri 1.0’den büyükse son üretilen rassal değer yerine yeni bir değer türetilir. Bu işlem ilk 3 genlik değerlerinin karelerinin toplamaları 1.0’den küçük olacak şekilde devam ettirilir. Gerekli koşul sağlandığında, son rassal genlik değeri “d”,

$$|d|^2 = 1 - |a|^2 + |b|^2 + |c|^2$$

$$d = \sqrt{1 - |a|^2 + |b|^2 + |c|^2}$$

İşleminden yararlanarak hesaplanır. Böylece tüm rassal değişkenlerin karelerinin toplamalarının 1.0’a eşit olması garanti edilmiş olur. Eğer rassal genlik değerlerinin karmaşık sayı formunda olması istenirse yapılacak değişiklik Tablo 3.2’de gösterildiği gibi olacaktır. Bu durumda, C# dilinin “Complex” sayı sınıfına başvurulur. “Complex” sayı sınıfıyla reel ve imajiner sayılar üzerinde ve onların eşlenikleriyle çok pratik şekilde işlem yapılabilir. Tablo 3.7’de kompleks rassal genlik değeri üretimi algoritmasının C# kodu sunulmuştur.

Tablo 3.7 İki kübitlik sistem için kompleks rassal genlik değeri üretimi

```

/* İki kübitlik sistem için 4 kompleks genlik değeri üretme
algoritması */
int tamam = 0;
double genlik = 0.0f;
double rast = 0.0f;
double rastImaj = 0.0f;
List<Complex> rastDouble = new List<Complex>();
rast = Random.Range(-1.0f, 1.0f);
rastImaj = Random.Range(-1.0f, 1.0f);
rastDouble.Add(new Complex(rast, rastImaj));
genlik = genlik + Complex.Multiply(rastDouble[tamam],
Complex.Conjugate(rastDouble[tamam])).Magnitude;
while (tamam<2)
{
    rast = Random.Range(-1.0f, 1.0f);

```

```

        rastImaj = Random.Range(-1.0f, 1.0f);
        Complex kompleks = new Complex(rast, rastImaj);
        genlik = genlik + Complex.Multiply(kompleks,
        Complex.Conjugate(kompleks)).Magnitude;
        if (genlik < 1.0f)
        {
            rastDouble.Add(kompleks);
            tamam++;
        }
        else
        {
            genlik = genlik - Complex.Multiply(kompleks,
        Complex.Conjugate(kompleks)).Magnitude;
        }
    }
    rast = Math.Sqrt(1 - genlik);
    rastDouble.Add(new Complex(rast, 0.0f));

```

Bu uygulamada, tam sayı genlik değerlerini tutan “double” türündeki “List” veri tipi “Complex” türünde tanımlanmıştır. C# dilinde “Complex” sınıfında oluşturulan verilerin reel ve imajiner olmak üzere iki “double” değeri vardır. Kod bloğunda görüldüğü üzere hem reel hem de imajiner kısımlar (-1.0f – 1.0f) aralığında rassal olarak üretilmektedir.

“Complex” sınıfının çok sayıda yöntemi ve “public” özelliği (property) vardır. Bu uygulamada, iki kompleks sayının çarpımını gerçekleştiren “Multiply()” ve bir kompleks sayının eşleniğini almada kullanılan “Conjugate()” yöntemleri kullanılmıştır. “Magnitude” özelliğiyle karmaşık sayının büyüklüğü döndürülür. Algoritmanın tek kusuru son olasılık genliğinin reel olarak tanımlanmak zorunda olmasıdır. Bu durum, toplam genlik değerini 1 yapacak sonsuz sayıda kompleks sayı olmasından ileri gelmektedir.

Kuantum bilişiminde kullanılan kapılar çoğunlukla “unitary” formda olup, kübitler üzerindeki etkilerin hesaplanmasında matris formunda işlem yapmak gerekir. Dolayısıyla vektörler konusuna ve matris formunda işlem yapmaya hâkim olmakta fayda vardır. Kısacası kuantum bilişiminin temelinde lineer cebir vardır. Kullanılan kapılar ve kübitler dizi değişken şeklinde oluşturulmuştur. Tablo 3.8’de tamsayı dizi değişken tipinde oluşturulan örnek kapıların nasıl hazırlandığı gösterilmiştir.

Tablo 3.8 Kullanılan kapılar ve kübitlerin matris formunda hazırlanması

```

int[,] Hadamard = new int[2, 2]{
    {1,1},
    {1,-1}
};
int[,] Identity = new int[2, 2]{

```

```

        {1,0},
        {0,1}
    };
    int[,] Tag1 = new int[2, 2]{
        {1,0},
        {0,1}
    };
    int[,] Tag2 = new int[2, 2]{
        {1,0},
        {0,1}
    };
    int[,] Tag3 = new int[2, 2]{
        {1,0},
        {0,1}
    };
    int[,] Tag4 = new int[2, 2]{
        {1,0},
        {0,1}
    };
    int[,] CNOT = new int[4, 4]{
        {1,0,0,0},
        {0,0,0,1},
        {0,0,1,0},
        {0,1,0,0}
    };
    int[,] Measure0 = new int[4, 4]{
        {0,0,0,0},
        {0,1,0,0},
        {0,0,0,0},
        {0,0,0,1}
    };
    int[,] Measure1 = new int[4, 4]{
        {0,0,0,0},
        {0,0,0,0},
        {0,0,1,0},
        {0,0,0,1}
    };
    int[,] hat0 = new int[2, 1]{
        {0},
        {0}
    };
    int[,] hat1 = new int[2, 1]{
        {0},
        {0}
    };
    int[,] Qubit0 = new int[2, 1]{
        {1},
        {0}
    };
    int[,] Qubit1 = new int[2, 1]{
        {0},
        {1}
    };
    int[,] QubitLine0 = new int[2, 1]{
        {1},
        {0}
    };
    int[,] QubitLine1 = new int[2, 1]{
        {1},
        {0}
    };

```



```

};

int[,] sonuc = new int[4, 1]{
    {0},
    {0},
    {0},
    {0}
};

int[,] new_kubit = new int[4, 1]{
    {0},
    {0},
    {0},
    {0}
};

```

Tablo 3.8’de “Hadamard” değişkeni Hadamard kapısının 2x2 matris formunu göstermektedir. İhtiyaca bağlı olarak 2x2 “Unitary” formda olan diğer kapılar da aynı yöntemle oluşturulabilir. “Identity” isimli değişken 2x2 boyutunda birim matristir. Birim matrisler çarpım işleminde etkisiz eleman görevi gören matrislerdir. Aynı sonucun korunması istenen durumlarda bu matristen yararlanılacaktır.

“Tag1”, “Tag2”, “Tag3” ve “Tag4” isimleriyle gösterilen matrisler ise kapıların bırakıldığı soketlere karşılık gelen değerleri tutacak değişkenlerdir. Hangi kapı ilgili sokete bırakılırsa ona karşılık gelen “TagX” matrisi o değeri alacaktır. “hat0” ve “hat1” isimli 2x1 boyutundaki matrisler ise hattın başlangıcından hat sonundaki ölçme işlemine kadar kübitin evresini tutacak olan değişkenlerdir. Bu değişkenlerin taşıdığı değere göre ölçme sonucu üretilir. “QubitLine0” ve “QubitLine1” değişkenleri ise rassal olarak üretilen kübitleri tutan matris değişkenleridir.

IBM kuantum besteleycisi simülatöründe daha önce de söylendiği üzere ilgili hat üzerinde ölçme işlemi sonucunda, $|1\rangle$ değeri elde edilmektedir. İki kübitlik bir sistemde bunu gerçekleştirmek için “Measure0” ve “Measure1” 2x2 boyutlu matris değişkenleri tasarlanmıştır. İki kübitlik bir sistemde $|00\rangle + |01\rangle + |10\rangle + |11\rangle$ evreleri olduğundan, “Measure0” operatörü 0’ıncı kübitin “1” olduğu $|01\rangle$ ve $|11\rangle$ evrelerini, “Measure1” operatörü ise 1’nci kübitin “1” olduğu $|10\rangle$ ve $|11\rangle$ evrelerini seçer. Her iki ölçme işlemi de uygulanırsa ikisinin kesişimi olan $|11\rangle$ evresi seçilir.

Tablo 3.9’da işlemlerin nasıl yapıldığı gösterilmiştir. Dikkat edileceği üzere işlemler “MonoBehaviour” tipindeki sınıf dosyasının “Update()” fonksiyonu altında yapılmaktadır. Bunun nedeni, hatların üzerinde yapılacak her değişimin izlenmesi ve yeni

sonucun güncellenerek ekrana aktarılmasının sağlanmasıdır.

Tablo 3.9 Çok kubitli ölçme uygulamasında matris işlemleri

```
private void Update()
{
    foreach (GameObject game in Capsules)
    {
        game.gameObject.SetActive(false);
    }
    if (!SocketMultiFirstTag.firstTag && !SocketMultiThirdTag.thirdTag)
    {
        Tag1 = Identity;
        Tag3 = Identity;
    }
    else if (SocketMultiFirstTag.firstTag && !SocketMultiThirdTag.thirdTag)
    {
        Tag1 = Hadamard;
        Tag3 = Identity;
    }
    else if (!SocketMultiFirstTag.firstTag && SocketMultiThirdTag.thirdTag)
    {
        Tag1 = Identity;
        Tag3 = Hadamard;
    }
    else if (SocketMultiFirstTag.firstTag && SocketMultiThirdTag.thirdTag)
    {
        Tag1 = Hadamard;
        Tag3 = Hadamard;
    }
}

for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        toplam = toplam + Tag1[i, j] * QubitLine0[j, 0];
    }
    hat0[i, 0] = toplam;
    toplam = 0;
}

for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        toplam = toplam + Tag3[i, j] * QubitLine1[j, 0];
    }
    hat1[i, 0] = toplam;
    toplam = 0;
}

for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        sonuc[toplam, 0] = hat0[j, 0] * hat1[i, 0];
        toplam++;
    }
}
```

```

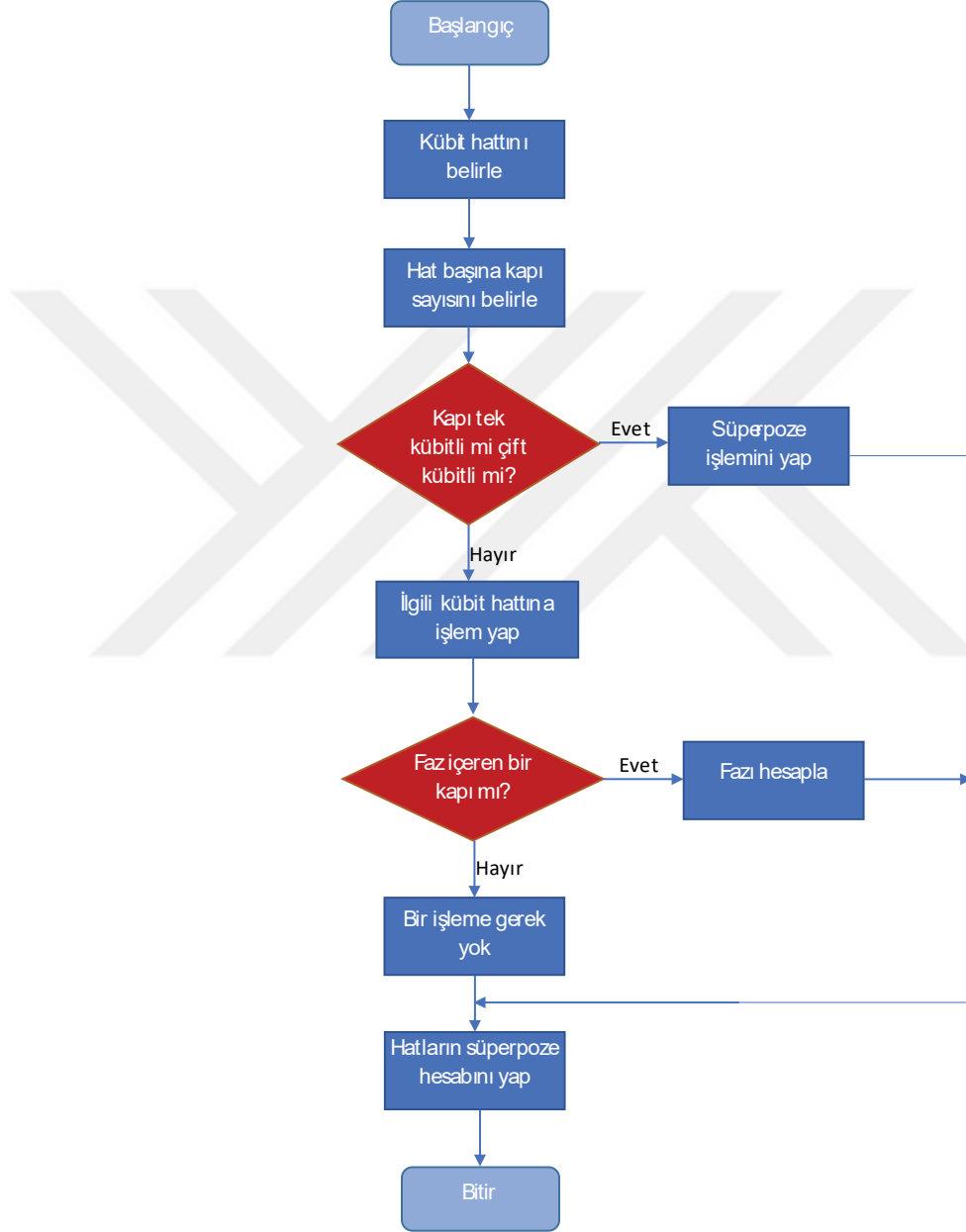
    }
    toplam = 0;
    if (SocketMultiFirstTag.firstTag && SocketMultiSecondTag.secondTag)
    {
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                toplam = toplam + Measure0[i, j] * sonuc[j, 0];
            }
            new_kubit[i, 0] = toplam;
            toplam = 0;
        }
        sonuc = new_kubit;
    }
    toplam = 0;
    if (SocketMultiThirdTag.thirdTag && SocketMultiFourthTag.fourthTag)
    {
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                toplam = toplam + Measure1[i, j] * sonuc[j, 0];
            }
            new_kubit[i, 0] = toplam;
            toplam = 0;
        }
        sonuc = new_kubit;
    }
    toplam = 0;
    for (int i = 0; i < sonuc.Length; i++)
    {
        if (Mathf.Abs(sonuc[i, 0]) == 1)
        {
            Capsules[i].gameObject.SetActive(true);
            stateMessage[i].text = states[i];
        }
        if (sonuc[i, 0] == -1)
        {
            stateMessage[i].text = "e<sup>i\u003C0</sup>" + states[i];
        }
    }
}

```

İlgili kod incelenecek olursa, öncelikle “TagX” isimli matris değişkenlere hangi matris değerlerinin karşılık geleceği belirlenmektedir. “Tag1” ve “Tag3” matris değişkenleri “Hadamard” kapılarını almaktadır. Eğer “Hadamard” kapısı yerleştirilmezse etkisiz eleman olarak “Identity” operatörüne eşitlenmektedirler. 2’ye 2 “for” döngüleriyle “hat0” ve “hat1” matrislerinin ölçme işlemi öncesi değerleri elde edilmektedir. Her iki hatta da “Hadamard” ve “Measurement” kapılarının olması durumunda 4’e 4 “for” döngüleriyle, Bloch küresi üzerinde sonucun gösterimini sağlayacak olan 4x1 boyutlu “sonuc” matris

değişkeni hazırlanır.

Serbest uygulama kapsamında kullanıcıların belli kapıları sıralı şekilde kullanabilmesi için bir yaklaşım geliştirilmiştir. Bu kapsamda ortaya konan yaklaşımın temel işleyiş algoritması Şekil 3.18’de gösterildiği gibidir.



Şekil 3.18 Çok kapılı bir tasarımda işlem akışı

Tek kübitli kapılardan PauliY kapısı -i ve i karmaşık genliklerini içermektedir. Dolayısıyla $e^{i3\pi/2}$ ve $e^{i\pi/2}$ ’lik faz farkları oluşur. Ayrıca PauliZ kapısında ve Hadamard kapısında -1 değeri bulunur ve bu değer $e^{i\pi}$ ’lik bir faz farkı oluşturur. Eklenen her kapıda

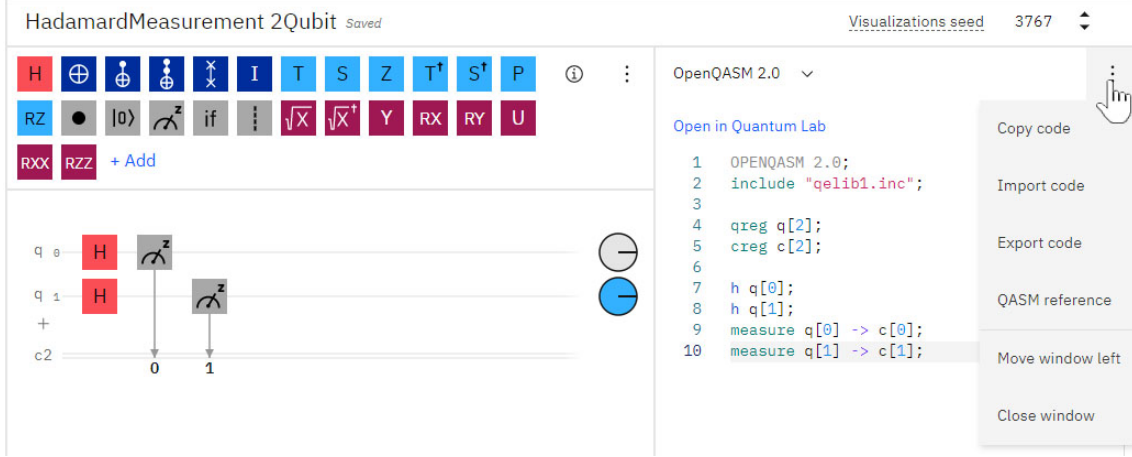
faz işlemlerinin de gerçekleştirilmesi gerekmektedir. Kapılar 2x2 ve 4x4 tamsayı matris düzeninde hazırlandığından faz işlemleri işleme alınan kapının gerçekte içerdiği faz durumu hesaba katılarak gerçekleştirilir. Bunun için 2 kübitlik bir sistemin 4 evresinin her biri için faz operatörü bulundurulur. Nihai işlem gerçekleştirildiğinde “Bloch” küresinde ilgili evrenin bilgisine faz operatörünün de değeri metin bilgisi olarak eklenir.

3.3. QASM Dosyasının Üretilmesi

VR ortamda dizilen kapıların, IBM kuantum besteleyicisinin metin temelli dosya türü olan QASM dosyası biçiminde üretilmesi için C# diliyle dosya işlemleri yönetimi gerçekleştirmek gerekir. Bunun için “System.IO” kütüphanesi çağrılarak Unity platformunda dosya giriş/çıkış işlemleri etkinleştirilir.

C# dilinde dosyaya yazmak için “StreamWriter” nesnesi kullanılır. Bu nesnenin temel olarak 10 kadar Yapıcı/Kurucu (Constructor) yöntemi vardır. Bu uygulamada “StreamWriter(string path, bool append)” yapıcı yöntemi kullanılmıştır. Oluşturulacak metin dosyasının yolu “String” veri tipinde olan “path” isimli parametreyle belirlenir. “Boolean” tipindeki “append” isimli parametreyle verinin dosyaya ne şekilde yazılacağı belirtilir. Eğer “true” ise yeni veri öncekinin devamına yazılır, “false” ise önceki verinin üzerine yazılır. Eğer belirtilen yolda dosya mevcut değilse bu parametre dikkate alınmaz ve yeni bir dosya oluşturulur.

Şekil 3.19’da IBM kuantum besteleyicisinde 2 kübitlik bir tasarım ekranı ve sağ tarafında QASM çıktısı gösterilmiştir. Bu çıktı, “.qasm” uzantısıyla metin formatında bir dosya olarak ekranda görülen satır düzeninde bilgisayara kaydedilebilir. Bunun için şekilde görülen menü ekranındaki “Export code” komutu seçilir.



Şekil 3.19 Örnek bir besteleyici ekranı ve QASM çıktısı

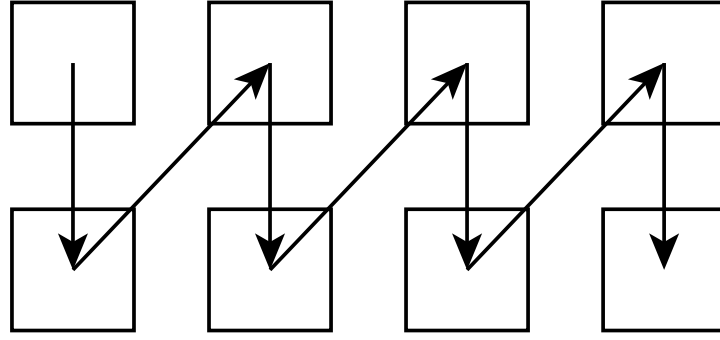
Unity platformunda, kullanıcının verilen kapıları kullanması suretiyle böyle bir dosyalama işlemini VR ortamda yapması mümkündür. Böylece oluşturulan metin dosyası, şekilde görülen menü ekranındaki “Import code” komutuyla sanal gerçeklik ortamında seçilerek, besteleyici ekranına yükleme işlemi yapılabilir.

IBM kuantum standart işlem kütüphanesinin kullanılması için gerekli kapı modellerinin yüklenmesi talimatını belirten “include qelib1.inc” satırı, ilk başlık (header) satırı olarak qasm dosyasında yer alır. Ardından tasarımı yapılacak modelde kaç kübit kullanılacaksa, kübit sayısını ve ölçme işlemi için kaçınıcı bitlerden ölçme işlemi yapılabileceğini belirten satırlar yazılır. Şekil 3.19’daki örnekte bu satırlar “qreg q[2]” ve “creg c[2]” olarak görülmektedir. Tablo 3.10’da örnek olarak “circuit.qasm” dosyası için başlık satırlarının oluşturulması gösterilmiştir.

Tablo 3.10 QASM dosyası için başlık satırlarının oluşturulması

```
string path = "C:/Users/HAKANGENC/Downloads/circuit.qasm";
StreamWriter writer = new StreamWriter(path, false);
writer.WriteLine("OPENQASM 2.0;");
writer.WriteLine("include \"qelib1.inc\";");
writer.WriteLine("qreg q[2];");
writer.WriteLine("creg c[2];");
```

IBM kuantum besteleyicisinde kapılar sıralı şekilde yerleştirilmektedir ve her bir kapının çıktıya etkisi Şekil 3.20’de gösterildiği gibi bir mantıkla işlemektedir.



Şekil 3.20 Kuantum kapılarının hiyerarşik olarak işleme sırası

Bu sıralı mantığa göre işlem yapılabilmesi için, tamsayı matris formatında tanımlanmış hücrelere karşılık gelen kuantum kapılarının etiket değerleri string formatında tanımlanmış bir “List” değişkenine atanır. Böylece “List” formatında tanımlanmış dizinin ilgili indis numarasına göre kapı etiketinin kod karşılığı yerleştirilir.

İlgili hücreye kapının yerleştirilmesi ya da o hücreden kapının çıkartılması durumu VR ortamda XRSocketInteractor aracıyla gerçekleştirilmiştir. İlgili hücrede bir kuantum kapısının olmaması durumunda, hücrenin matris değeri “Identity” isimli birim matrisine eşitlenmektedir. Böylece matematiksel operasyonda o hücrenin bir etkisi olmamaktadır.

Kullanıcının ekranda ilgili kapıyı seçmesi sonucu, önceden tanımlanmış kapı etiketine göre metin dosyasına kodun eklenmesi gerekir. Böyle bir uygulama örneği Tablo 3.11’de gösterilmiştir. Bu işlemler ilgili program kodunda “Update()” yöntemi içinde gerçekleştirilir. Böylece kullanıcının uygulamayı çalıştırmasına kadar yaptığı tüm değişiklikler güncel olarak revize edilmiş olur.

Tablo 3.11 QASM dosyasına seçili kapı etiketine göre ilgili satırın yazılması

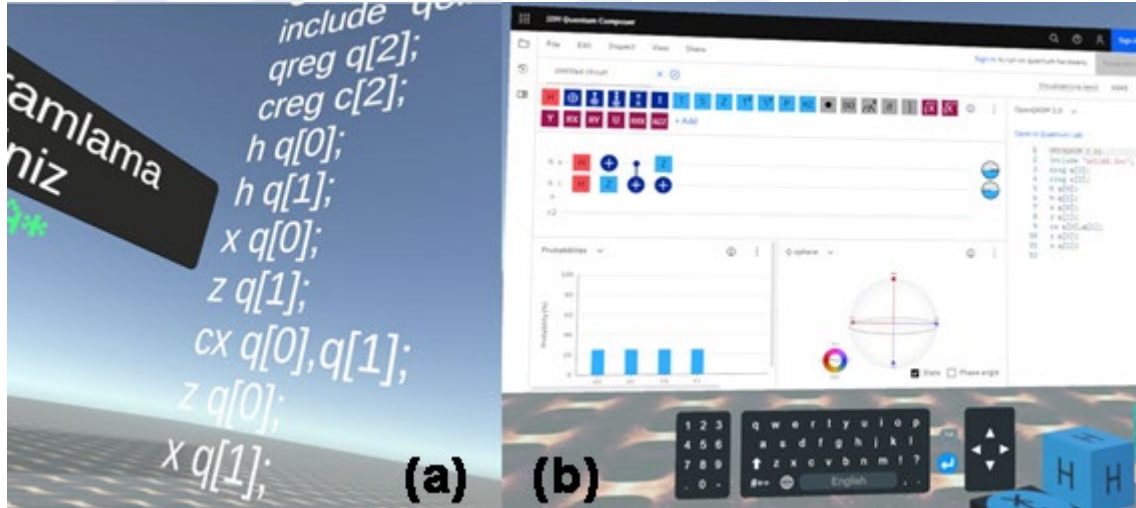
```
List<string> kod = new List<string>() { "", "", "", "", "", "", "", "" };
if (SocketAllFirstTag.karsilas == "PauliX")
{
    Tag1 = PauliX;
    kod[0] = "x q[0];\n";
}
```

Tüm hücreler için taranıp, etiket değerlerine göre “List” tipindeki değişkenin ilgili indis numarasına atama yapılır. VR eğitim platformunda “RUN” butonuna basılınca, içinde bir kuantum kapısı barındıran hücrelerin kod karşılıkları, Tablo 3.12’de gösterilen tarama uygulamasıyla başlık satırlarından sonra eklenir.

Tablo 3.12 Hücrelere karşılık gelen kapı etiketlerinin kod karşılıklarının eklenmesi

```
foreach(string s in kod)
{
    if(s != "")
    {
        writer.Write(s);
    }
}
```

Kullanıcı sanal ekranda gözüken “RUN” komutuna tıkladığında, “circuit.qasm” dosyasının içeriği sanal gerçeklik ekranında bir Canvas yüzeyinde gösterilir. Aynı zamanda sanal gerçeklik web tarayıcısında boş bir besteleyici alanı açılır. Böylece kullanıcının ürettiği dosyayı web tarayıcısında yükleyerek gerçek zamanlı olarak işletmesi sağlanır. İki kübitlik örnek bir uygulamanın VR ortamında elde edilen kod çıktısı ve besteleyiciye yüklendikten sonra elde edilen çıktısı Şekil 3.21’de gösterilmiştir.



Şekil 3.21 Örnek bir kod çıktısının (a) VR ortamda yazdırılması ve (b) besteleyiciye yüklenmesi

Çalıştır komutuyla birlikte daha önceden tanımlanmış adres yoluna kaydedilen qasm metin dosyası, besteleyiciye yüklendiğinde gerçek zamanlı olarak derleme işlemi yapılır.

4. SONUÇ VE TARTIŞMALAR

Mekanik bilgisayarlardan lambalı bilgisayarlara ve ardından yarı iletken mimarisine sahip büyük ölçekli entegre devre sistemlerine uzanan süreçte, süper iletken kuantum bilgisayarlar yeni bir çıkış açan teknoloji olarak ortaya çıkmaktadır. Dolayısıyla böyle bir teknolojinin yeni başlayanlara mevcut en güncel araçlarla öğretilmesinin anlamlı olacağı düşünülmektedir.

Bu araçla sanal gerçeklik ortamında kuantum programlama yapmayı sağlayan çok disiplinli bir çalışma gerçekleştirilmiştir. Uygulama kapsamında VR platformlarının kullanımı, Unity ile 3D ortam tasarımı ve nesne kontrolü, C# dili ile betik dili programlama, temel kuantum mekanikleri ve lineer cebir bilgisi, kuantum bilgisayar programlama bilgisi gibi birbirleriyle doğrudan ilişkili olmayan çok sayıda disiplin tek bir çalışmada entegre şekilde sunulmuştur.

Sürekli güncellenen ve neredeyse her güncellemeden sonra temel “XR Plug-in Management” araçlarında veya “Render Pipeline” araçlarında bazı düzeltmeler gerektiren Unity gibi bir programda, böyle bir uygulamayı tutarlı bir şekilde çalıştırmak oldukça güç olmuştur. Ancak tüm bu güçlüklerin üstesinden gelmek için gerekli bilgi ve deneyim kazanılmıştır.

Kuantum bilgisayarlarının ve işlemcilerinin geliştirilme süreciyle birlikte artık herkes tarafından erişilebilen web tabanlı platformlar ile kuantum bilişiminde de hızlı ilerleme devam etmektedir. Yakın zamana kadar, makine öğrenmesi ve yapay zekâ algoritmaları gibi klasik bilgisayar işlemcileriyle gerçekleştirdiğimiz üst seviye işlemlerin artık kuantum transfer öğrenme algoritmaları gibi kuantum mekaniği ilkelerini kullanacak şekilde uyarlandığını görmekteyiz.

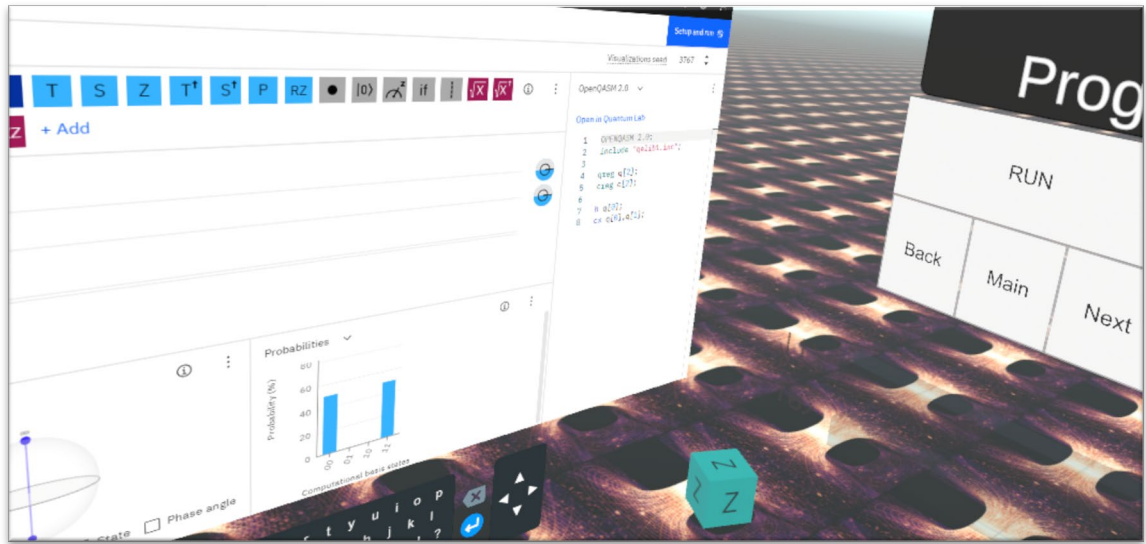
Bu çalışma kapsamında VR Kuantum Programlama adlı araç tasarlanmıştır. Tasarlanan araç, kullanıcının Oculus VR başlığına sahip IBM kuantum web besteleyicisinde gezinmesine olanak tanır. Ayrıca, kullanıcının kuantum kapıları hakkındaki yardım aracına çok esnek bir şekilde erişmesini sağlar. Bu aracın kuantum bilgisayarları programlamayı kolaylaştırması ve öğrenmeyi kolaylaştırıcı olması beklenmektedir.

Yapılan ayrıntılı literatür taramasında böyle bir uygulama tasarımına rastlanmamıştır. Tasarım bu yönüyle kuantum bilişiminin eğitimi açısından mühendislik eğitimi

disiplininde özgün ve yeni bir yaklaşım sunmaktadır. Bilgisayar programcılığının, robotik kodlama, scratch benzeri puzzle mimarili kodlama, modüler kodlama, vb. farklı yaklaşımlarla uygulamalarının karşımıza çıktığı günümüzde bilgisayar programcılığı açısından da özgün bir yaklaşım sunmaktadır. Bu çalışma kapsamında elde edilen bir kısım bilimsel çıktılar SCİexpanded indekste taranan “Computer Applications in Engineering Education” dergisinde yayınlanmıştır.

Tez çalışmasının başlatılmasından yaklaşık 2 yıl sonra, IBM firmasınca kuantum mekaniğinin ve kuantum kapıları ekseninde kuantum bilgisayar programlama ilkelerinin öğretilmesi için “Quantum Games” adı altında 3 boyutlu oyun yaklaşımlı bir eğitim uygulaması sunulmuştur. Ayrıca 2020 tarihi itibarıyla aralarında ülkemizin de olduğu bazı ülkelerde kuantum mekaniği davranışlarını öğretmek amacıyla kuantum oyunları tasarımı yarışları düzenlenmeye başlamıştır. Ülkemizde de ilk olarak 2020 ve ardından 2021 yıllarında, QTurkey takımı koordinatörlüğünde QOyun’2020 ve QOyun’2021 Quantum Hackathonu yarışma etkinlikleri gerçekleştirilmiştir.

Platform içinde gömülü web tarayıcısı ile Şekil 4.1’de görüldüğü gibi IBM kuantum besteleyicisinde kodlar arasında gezinmek ve kurulan devrelerin QASM koduna erişmek mümkündür. VR platformun sunduğu klavye eklentisi ile QASM koduna müdahalede bulunup kod değiştirilebilir ve güncel sonuç besteleyicide görülebilir.



Şekil 4.1 Gömülü web tarayıcısı ile kuantum besteleyicisinde gezinti

Besteleyici sayfasındaki kod sisteme indirilebilirken karşı tarafa da VR platform içinden

kod göndermek mümkündür. Bu özellik, VR temelli web tarayıcısının dosya yönetim temel dosyalarında yapılan güncelleme sayesinde etkin hale getirilmiştir. Böylece kullanıcı belli sınırlar içinde kalmadan istediği düzenlemeyi yapabilmekte ve elde edilen kodu IBM kuantum besteleycisi içinden “import” komutuyla içeri yüklemek suretiyle daha esnek bir derleme işlemi yapabilmektedir.

Çalışmanın, Unity yazılımı ile VR ortamındaki tasarımında Alex Okita’nın (Okita, 2020) ve Jonathan Linowes’in (Linowes, 2020) kitaplarından etkin şekilde yararlanılmıştır. Kuantum mekaniklerinin davranışlarının yeni başlayanlara göre de son derece uygun bir dil ve seviyede izah edildiği, kuantum bilişiminin ve kuantum kapıların temellerinin ilgili herkesin rahatlıkla anlayabileceği şekilde anlatıldığı, “Quantum Computing for the Quantum Curious” (Hughes ve ark., 2021) isimli kitaptan da çalışma sürecinde etkin şekilde yararlanılmıştır.

Tez kapsamında işletilen kuantum kapıların fiziksel davranışlarına ve matematiksel modellerine ancak tam hakimiyet sağlandığında sanal gerçeklik ortamına uyarlanabilmektedir. Bu çalışma kapsamında kuantum kapıların davranışlarına mantıksal olarak tam hakimiyet sağlanmıştır.

5. ÖNERİLER

Preskill'in de (2012) belirttiği üzere, azalan hata oranları ile kuantum bilgisayarların her geçen gün daha etkin kullanılacağı görülmüştür. Standart kuantum algoritmalarının ötesinde, ciddi düzeyde hesaplama avantajları sağlayabilen algoritmaların geliştirildiği günümüzde, kuantum bilgisayarların programlanması konusunda etkin mühendislik çalışmalarına ağırlık verilmelidir. Dolayısıyla bu çalışmada, üç boyutlu tasarım ortamlarının sağladığı avantajlar kullanılarak bu sürece katkı sağlanmıştır.

Elde edilen tecrübe ışığında içeriğin daha geniş bir çalışma ekibiyle güçlendirilmesi anlamlı olacaktır. Özellikle kuantum bilişiminin temelleri ve Unity platformunda 3 boyutlu VR temelli tasarım gerçekleştirme süreçleriyle çok yoğun şekilde zaman geçirilmiştir. Ayrıca VR tasarım 2 kübitli analiz gerçekleştirmeye uygun olacak şekilde tasarlanmıştır. Bloch küresi ile anlık sonucu görme uygulaması da en fazla 2 kübit üzerinden gerçekleştirilmiştir. Matris matematiğini daha yüksek kübitler için uygulayabilecek bir tasarım yaklaşımı uygulanabilir. Yüksek kübitli bir modelin matematiksel analizinin VR ortama aktarımı kuşkusuz iyi bir ekip çalışması gerektirecektir.

Söz konusu çalışma farklı disiplinler arası eşgüdümü gerektirmektedir. Tüm bu yaklaşım içerisinde, kuantum mekanikleri, kuantum bilişimi, sanal yazılım geliştirme, hedef kullanıcı odaklı içerik geliştirme, vb. süreçler tarafımdan hazırlanmıştır. Böyle bir çalışmanın, disiplinler arası eşgüdüm ile farklı uzmanların birlikte çalışabileceği daha profesyonel bir yaklaşımla ticarileştirilebilir bir ürüne dönüştürülmesi muhtemeldir.

KAYNAKLAR

Makale:

1. Abhijith, J., Adedoyin, A., Ambrosiano, J., Anisimov, P., Bärtschi, D., Casper, W., Chennupati, G., Coffrin, C., Djidjev, H., Gunter, D., Karra, S., Lemons, N., Lin, S., Malyzhenkov, A., Mascarenas, D., Mniszewski, S., Nadiga, B., O'Malley, D., Oyen, D., Pakin, S., Prasad, L., Roberts, R., Romero, P., Santhi, N., Sinitsyn, N., Swart, P.J., Wendelberger, J.G., Yoon, B., Zamora, R., Zhu, W., Eidenbenz, S., Coles, P.J., Vuffray, M., Lokhov, A.Y., (2020). Quantum Algorithm Implementations for Beginners, arXiv:1804.03719v2 [cs.ET].
2. Acar, E., Yılmaz, İ., (2021). COVID-19 detection on IBM quantum computer with classical-quantum transfer learning, Turkish Journal of Electrical Engineering & Computer Sciences, 29, 46-61.
3. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brando, F.G.S.L., Buell, D.A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Heibegger, S., Harrigan M.P., Hartmann, M.J., Ho, A., Hoffman, M., Huang, T., Humble, T.S., Isakov, S.V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P.V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandra, S., McClean, J.R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M.Y., Ostby, E., Petukhov, A., Platt, J.C., Quintana, C., Rieffel, E.G., Roushan, P., Rubin, N.C., Sank, D., Satzinger, K.J., Smelyanskiy, V., Sung, K.J., Trevithick, M.D., Vainsencher, A., Villalonga, B., White, T., Yao, Z.J., Yeh, P., Zalcman, A., Neven, H., Martinis, J.M., (2019). Quantum supremacy using a programmable superconducting processor, Nature, 574, 505-511.
4. Aviles, F.R., Aviles-Cruz, C., (2018). Handheld augmented reality system for resistive electric circuits understanding for undergraduate students, Computer Applications in Engineering Education, 26, 602-616.
5. Aviles-Cruz, C., Villegas-Cortez, J., (2019). A smartphone-based augmented reality system for university students for learning digital electronics, Computer Applications in Engineering Education, 27, 615-630.

6. Azuma, R. T., (1997). A survey of augmented reality, *Presence: Teleoperators and Virtual Environments*, 6, 355–385.
7. Azuma, R.T., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B., (2001). Recent Advances in Augmented Reality, *Computer Graphics and Applications*, IEEE, 21, 34-47.
8. Behera, B., Seth, S., Das, A., Panigrahi, P., (2019). Demonstration of Entanglement Purification and Swapping Protocol to Design Quantum Repeater in IBM Quantum Computer, *Quantum Information Processing*.
9. Bennett, H., Bernstein, E., Brassard, G., Vazirani, U., (1997). Strengths and Weaknesses of Quantum Computing, *SIAM Journal on Computing*, 26(5), 1510-1523.
10. Coiera, E., (2003). Interaction design theory. *International Journal of Medical Informatics*, 69(2-3), 205–222.
11. Christopoulou, E., Xinogalos, S., (2017). Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices, *International Journal of Serious Games*, 4, 21-36.
12. Deutsch, D., Jozsa, R., (1992). Rapid solution of problems by quantum computation, *Proceeding of Royal Society A*, 439(1907), 553-558.
13. Ekert, A., Jozsa, R., (1996). Quantum computation and Shor's factoring algorithm, *Reviews of Modern Physics*, 68(3), 733-753.
14. Feynman, R. P., (1982). Simulating Physics with Computers, *International Journal of Theoretical Physics*, 21(6/7), 467-488.
15. Feynman, R.P., (1986). Quantum Mechanical Computers, *Foundations of Physics*, 16(6), 507-531.
16. Grover, L.K., (1996). A fast quantum mechanical algorithm for database search, *Annual Acm Symposium on Theory of Computing*, 212-219.
17. Harrow A. W., Montanaro A., (2017). Quantum computational supremacy, *Nature*, 549, 203-209.
18. Hewett, T., (1999). Human-Computer Interaction and Cognitive Psychology in Visualization Education, in *Handbook of Research on Human Cognition and Assistive Technology: Design, Accessibility and Transdisciplinary Perspectives*.

19. Kanamori, Y., Yoo S.-M., Pan W.D., Sheldon, F.T., (2006). A short survey on quantum computers, *International Journal of Computers and Applications*, 28, 228-233.
20. Ladd, T.D., Jelezko, F., Laflamme, R., Nakamura, Y., Monroe, C., O'Brien, J.L., (2010). Quantum computers, *Nature*, 464(7285), 45-53.
21. Lindgren, R., Tscholl, M., Wang, S., Johnson, E., (2016). Enhancing learning and engagement through embodied interaction within a mixed reality simulation, *Computer Applications in Engineering Education*, 95, 174–187.
22. Mehlenbacher, B., (2002). Assessing the usability of on-line instructional materials, *New Directions for Teaching and Learning*, 91-98.
23. Moreno-Pineda, E., Godfrin, C., Balestro, F., Wernsdorfer, W., Ruben, M., (2017). Molecular Spin Qudits for Quantum Algorithms, *Chemical Society Reviews*, 47.
24. Nasar, A.A., (2016). The history of algorithmic complexity, *The Mathematics Enthusiast*, 13(3), 217-241.
25. Preskill, J., (2012). Quantum Computing and the Entanglement Frontier, arXiv:1203.5813v3.
26. Sharma, K., Thakur, S., Kalia, A., Thakur, J., Kumar, S., (2014). Emerging Cloud Computing Paradigm Vision, Research Challenges and Development Trends, *International Journal of Research in Engineering and Technology (IRJET)*, 3(5), 892-899.
27. Shor, P.W., (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal on Computing*, 26(5), 1484-1509.
28. Sim, S., Cao, Y., Romero, J., Johnson, P.D., Aspuru-Guzik, A., (2018). A framework for algorithm deployment on cloud-based quantum computers, *Quantum Physics*, Cornell University.
29. Soeparno, H., Perbangsa, A.S., (2021). Cloud Quantum Computing Concept and Development: A Systematic Literature Review, *Procedia Computer Science*. 179 (2021), 944-954.
30. Tansuwannont T., Limkumnerd S., Suwanna S. ve Kalasuwan P., (2019). Quantum Phase Estimation Algorithm for Finding Polynomial Roots, *Open Physics*, 17(1), 839-849.

31. Tiruneh, D.T., De Cock, M., Weldeclassie, A.G., Elen, J., Janssen, R., (2017). Measuring Critical Thinking in Physics: Development and Validation of a Critical Thinking Test in Electricity and Magnetism, *International Journal of Science and Mathematics Education*, 15, 663–682.
32. Zhukov, A.A., Pogosov, W.V., Lozovik, E., (2017). Modeling Dynamics of Entangled Physical Systems With Superconducting Quantum Computer, *arXiv:1710.09659*.

Bildiri:

1. Andrews, A., (2011). Serious Games for Psychological Health Education, *International Conference on Virtual and Mixed Reality (VMR 2011)*, 9-14 July, Orlando, FL, USA.
2. Bichsel B., Baader M., Gehr T. ve Vechev M., (2020). Silq: A High-Level Quantum Language with Safe Uncomputation and Intuitive Semantics, *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI'20)*, 15-20 June, London, UK.
3. Bellemans, M., Lamrnens, D., De Sloover, J., De Vleeschauwer, T., Schoofs, E., Jordens, W., Van Steenhuyse, B., Mangelschots, J., Selleri, S., Hamesse, C., Fréville, T., Haeltermann, R., (2020). Training Firefighters in Virtual Reality, *2020 International Conference on 3D Immersion (IC3D)*, 15 December, Brussels, Belgium.
4. Campos, P., Pessanha, S., (2011). Designing Augmented Reality Tangible Interfaces for Kindergarten Children, *International Conference on Virtual and Mixed Reality (VMR 2011)*, 9-14 July, Orlando, FL, USA.
5. de Sá, M., Carriço, L., Duarte, L., Reis, T., (2008). A mixed-fidelity prototyping tool for mobile devices, *AVI'08: Proceedings of the working conference on advanced visual interfaces*, Napoli, Italy.
6. Farias, L. Dantas, R. ve Burlamaqui, A., (2011). Educ-AR: A tool for assist the creation of augmented reality content for education, *2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS 2011)*, 19-21 September, Ottawa, ON, Canada.

7. Fragomeni, G., Lackey, S.J., Champney, R., Salcedo, J.N., Serge, S., (2015). Training Effectiveness Evaluation: Call for Fire Trainer – Augmented Virtuality (CFFT-AV), International Conference on Virtual, Augmented and Mixed Reality (VAMR 2015), 2-7 August, Los Angeles, CA, USA.
8. Genç, H.H., Aydın, S., Erdal, H., (2020). Using of Cloud-Based Quantum Computer Programming Platforms for High Precision Floating Point Calculations, 4th International Science Congress; 19-20 June, Kocaeli, Turkey.
9. Liu, X., Zhang, J., Hou, G., Wang, Z., (2018). Virtual Reality and Its Application in Military, 2nd International Symposium on Resource Exploration and Environmental Science, IOP Conference Series: Earth and Environmental Science, 28-29 April, Ordos, China.
10. Nascimento, A.M., Queiroz, A.C.M., Vismari, L.F., Bailenson, J.N., Cugnasca, P.S., Camarga, J.J.B., de Almeida, J.R., (2019). The Role of Virtual Reality in Autonomous Vehicles' Safety, 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), 9-11 December, San Diego, CA, USA.
11. Shor P.W., (1994). Algorithm for quantum computation: Discrete logarithm and factoring, Proc. 35th IEEE Annual Symposium on Foundations of Computer Science, 20-22 November, Santa Fe, NM, USA.
12. Thöner, M., Kuiper, A., (2015). Delta Global Illumination for Mixed Reality, International Conference on Virtual, Augmented and Mixed Reality (VAMR 2015), 2-7 August, Los Angeles, CA, USA.
13. Wu, Y., Zhang, Y., Shen, J., Peng, T., (2013). The Virtual Reality Applied in Construction Machinery Industry, International Conference on Virtual, Augmented and Mixed Reality (VAMR 2013), 21-26 July, Las Vegas, NV, USA.

Kitap:

1. Ferrone, H., (2019). Learning C# by Developing Games with Unity 2019, 4th Edition, Packt Publishing Ltd., Birmingham.
2. Hughes, C., Isaacson, J., Perry, A., Sun, R.F., Turner, J., (2021). Quantum Computing for the Quantum Curious, Open Access eBook, Springer, Cham.
3. Johnson, J.L., (2003). Distance Education. 1st Edition, Teachers College, Columbia University, New York.

4. Kolko, J. (2011), 'Exposing the Magic of Design a Practitioner's Guide to the Methods & Theory of Synthesis', Reprint Edition, Oxford University Press, USA.
5. Linowes, J., (2020). Unity 2020 Virtual Reality Projects, 3rd Edition, Packt Publishing Ltd., Birmingham.
6. Ma, D., Gausemeier, J., Fan, X., Grafe, M., (2011). Virtual Reality & Augmented Reality in Industry, The 2nd Sino-German Workshop, Springer-Verlag, Berlin.
7. Okita, A., (2020), Learning C# Programming with Unity 3D, 2nd Edition, CRC Press/Taylor & Francis Group, Boca Raton, FL.
8. Ömer B., (2009). Structured Quantum Programming, Institute for Theoretical Physics Vienna University of Technology.
9. Sharp, H., Yvonne, R., Jennifer, P., (2019). Interaction Design Beyond Human-Computer Interaction. 5th Edition. John Wiley & Sons, Inc, Indianapolis, IN.
10. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, N., (2018). Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th Edition. Pearson Education Limited.
11. Williams C.P., (2011). Exploration in Quantum Computing, 2nd Edition, Springer-Verlag, London.

Tezler:

1. Byrne, C., (1996). The Use of Virtual Reality as an Educational Tool. PhD thesis, University of Washington.
2. Ciocîrlan, D., (2014). On the Adiabatic Model of Quantum Computation. Bachelor of Computer Science and Engineering Department Thesis of Faculty of Automatic Control and Computers University POLITEHNICA of Bucharest.
3. Padilha, D., (2014). Solving NP-Hard Problems on an Adiabatic Quantum Computer. Bachelor of Aerospace Engineering Thesis of School of Mechanical and Manufacturing Engineering UNSW Sydney.

Raporlar:

1. Haas, J.K., (2014). A History of the Unity Game Engine, An Interactive Qualifying Project. WORCESTER POLYTECHNIC INSTITUTE.

Elektronik Yayınlar:

1. Lewis, J. R., IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. Url: <https://garyperlman.com/quest/quest.cgi>. (24.07.2021).
2. Lund, A. M., Measuring Usability with the USE Questionnaire. Url: <https://garyperlman.com/quest/quest.cgi?form=USE>. (24.07.2021).
3. Dwyer, B., (2020). Fun with the IBM Quantum Experience Circuit Composer. https://www.researchgate.net/publication/341539446_Fun_with_the_IBM_Quantum_Experience_Circuit_Composer





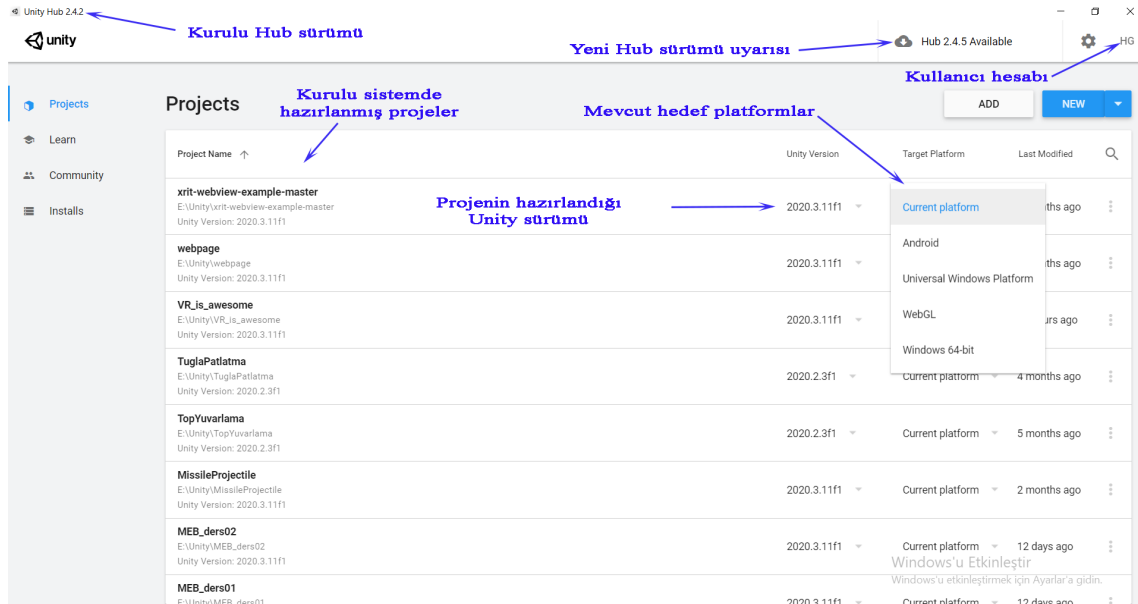
EK 1: Unity Programının Kurulumu Hakkında

Unity farklı kurulum alternatifleri sunmaktadır. Bu kurulum yöntemleri şunlardır:

- Unity Hub ile yükleme,
- Komut satırı ile yükleme,
- Çevrimdışı manuel yükleme.

Unity Hub, Unity programını kurmanın, projeler oluşturmanın ve Unity deneyiminizi yönetmenin birincil yoludur. Farklı sürüm kurulumlarınızı, hesaplarınızı, lisanslarınızı ve projelerinizi yönetmek için merkezi bir konum sağlar. Unity Hub, tüm Unity projelerinizi ve kurulumlarınızı yönetmek için kullanabileceğiniz bir yönetim aracıdır. Unity Editor'ün birden çok kurulumunu ilişkili bileşenleriyle birlikte yönetmek, yeni projeler oluşturmak ve mevcut projeleri açmak için Hub'ı kullanabilirsiniz.

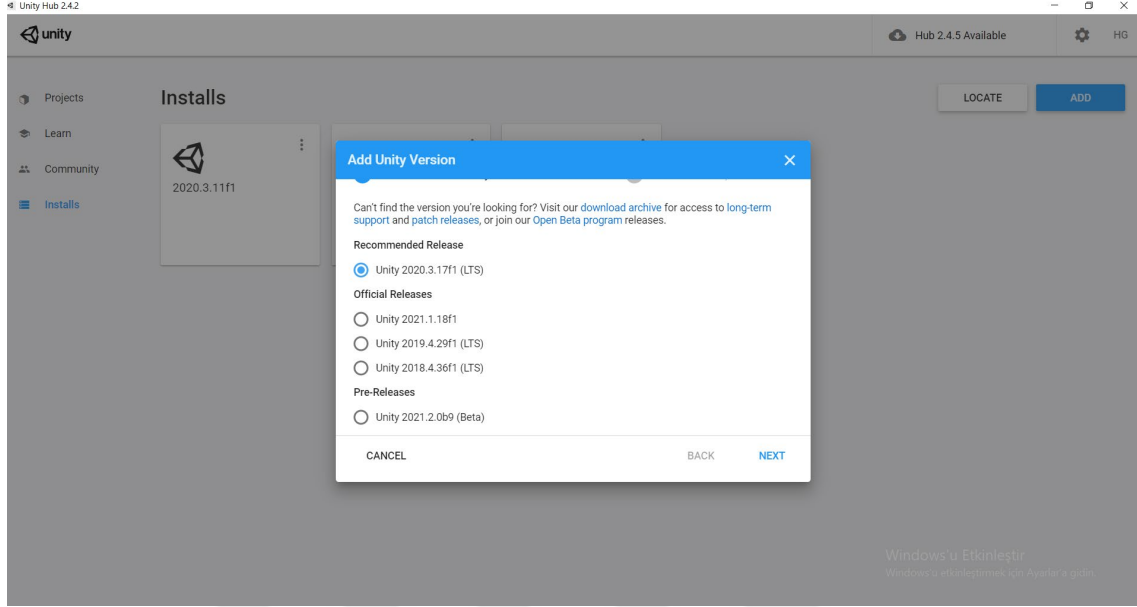
Hub aracılığıyla EK 1-Şekil 1'de görüldüğü üzere, proje dosyalarınızı, proje dosyanızın Unity sürümünü, projenin tasarlandığı platformu ve dönüştürülebileceği diğer hedef platformları, kullanıcı hesabınızı, Hub'ın sürümünü, projenizin en son ne zaman düzenlendiğini görebilirsiniz. Ayrıca “Installs” menüsünden sisteminize yüklü Unity sürümlerini de görebilirsiniz.



EK 1-Şekil 1 Unity Hub Projeler sayfası

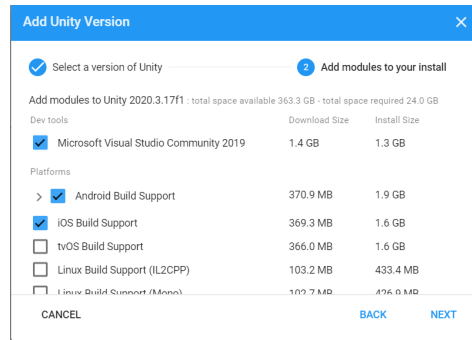
Windows, macOS ve Linux işletim sistemleri için bağımsız Unity Hub aracı Unity'nin resmî web sitesinden indirilebilir. Unity Hub aracı kurulduktan sonra Installs → ADD

seçeneğiyle EK 1-Şekil 2’de gösterildiği gibi yüklenmek istenen bir sürüm belirlenerek kurulum işlemi yapılır.



EK 1-Şekil 2 Unity Hub ile isteğe bağlı bir Unity versiyonu yüklemesi

Sürüm belirlendikten sonra “NEXT” butonuyla bir sonraki adıma geçilir ve seçilen sürüme göre kurulabilecek modüllerin seçim işlemi yapılır. Android ve iOS temelli akıllı cihazlar için uygulama geliştirme durumuna göre ilgili modüllerin seçimi bu kısımda yapılır.



EK 1-Şekil 3 Unity versiyonuna göre kurulabilecek modüllerin seçimi

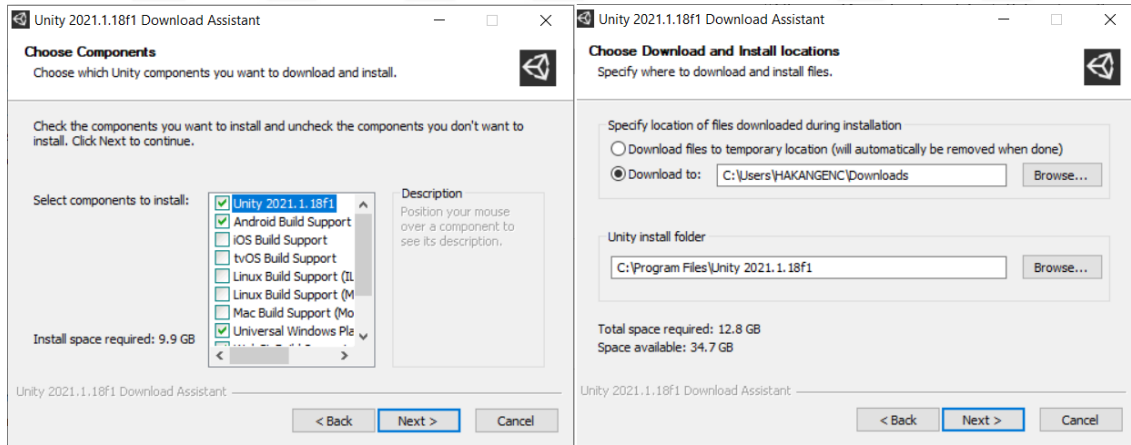
EK 1-Şekil 3’te gösterilen modül seçimi işlemi kurulum bittikten sonra da yapılabilir. “Installs” menüsünde ilgili sürümün üzerine gelip sağ tıkladığınızda “Add Modules” seçeneğiyle yüklenmemiş modülleri yükleyebilirsiniz. Belli bir sürüme göre tasarlanmış proje dosyasını yeni bir sürüme yükseltmek mümkündür. Ancak bu durumda dikkatli olunmalıdır. Sürüm yükseltme sonrası çeşitli nesnelerin yer aldığı “Prefabs”, “Scripts”, “Materials”, “Models” gibi varlıklarda (assets) yaşanacak değişimler projenin eksik

alışmasına ya da hi alışmamasına neden olabilir. Bazı durumlarda kullanılmış paketlerin yeni sürümlerinde “Script” kütüphanesinde önemli deėişiklikler olabilmektedir. Bu durumda proje dosyası alışmayacaktır. Dolayısıyla güncelleştirmeler öncesinde proje klasörünün bir yedeėinin alınmasında fayda vardır.

Bir Unity projesi klasörü, Unity Hub aracının “Projects” menüsündeyken “ADD” butonuna tıklanėında ıkan dosya gezgini penceresi yardımıyla seilerek proje dosyaları listesine eklenebilir. Bu özellik bir başka kaynaktan gelen Unity projesinin kolayca kullanılabilir olmasını saėlar.

Unity komut satırından da yüklenebilir. Bunun için,

- Unity yükleme arşivi (Unity download archive) sitesinden istenen bir sürüm seilir ve yükleyici dosyası indirilir.
- Yükleyci dosyası alıştırılır ve indirme asistanı penceresi açılır (Bkz. EK 1-Şekil 4). “Choose Components” penceresinde yüklenmesi istenen bileşenler seilir.
- “Choose Download and Install locations” penceresinden dosyaların indirileceėi ve yükleneceėi konum seilir.
- Kurulum bitirilir.



EK 1-Şekil 4 Unity yükleme asistanı

Kullandėınız işletim sistemi için komut satırı programı alıştırılır ve dosyaların indirildiėi klasöre gidilir. Klasörün içinde Windows işletim sistemi için “install.bat”, Linux ve OS X işletim sistemleri için “install.sh” yığın dosyası bulunur. Komut satırından bu dosya alıştırılarak kurulum gerçekleştirilir.

Unity için bir diėer kurulum seeneėi ise çevrimdışı olarak da kurulum yapabilmeyi

sağlayan seçenektir. Bunun için yine Unity yükleme arşivi sayfasına gidilir ve Unity Editor 64-bit seçeneği ile kurulum paketi indirilir. Bu dosya internet bağlantısının olmadığı bir durumda da çalıştırılarak kurulum yapılabilir.



EK 2: Oculus Rift ve XR Management Araçlarının Unity 3D'ye Kurulumu

Oculus donanımının bilgisayara kurulumu öncesinde akıllı telefonda “Oculus Companion App” uygulaması kurulur ve Settings → More Settings → Developer Mode ayarlarından geliştirici modu açılır. Bu süreçte bir Oculus hesabı oluşturulur. Oculus pek çok etkinliğin çevrimiçi yürütülebilmesi için kullanıcılardan Facebook hesabı ister. Ancak bir Google hesabıyla da hesap oluşturulabilir. Bağımsız olarak çalıştırılacak uygulamalar için bir sosyal medya hesabı şart değildir.

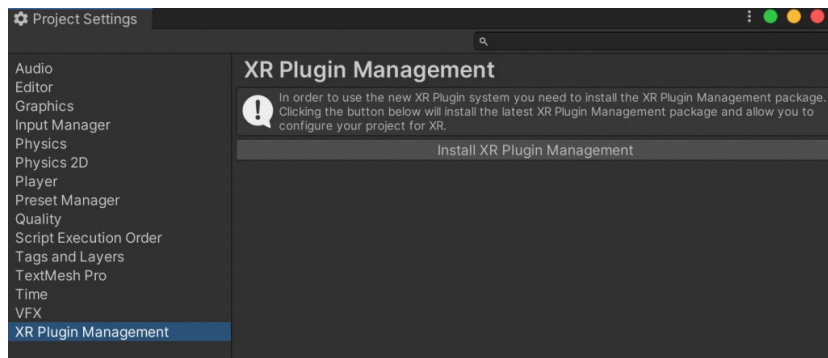
Bu işlem yapıldıktan sonra etkinleştirmek için kurulumu yapılacak cihaz bilgisayara uygun USB kablosuyla bağlanır. PC için de Oculus uygulaması kurulur. Kurulum işlemleri sonrasında Unity’de yeni bir proje dosyası oluşturulur.

Oluşturulan yeni proje dosyası açıldıktan sonra File → Build Settings menüsüne tıklanır. Ardından;

- Platform “PC, Mac & Linux Standalone” seçeneğiyle bağımsız masaüstü moduna ayarlanır.
- “Target Platform” seçeneği altında Windows ya da diğer işletim sistemi seçilir.

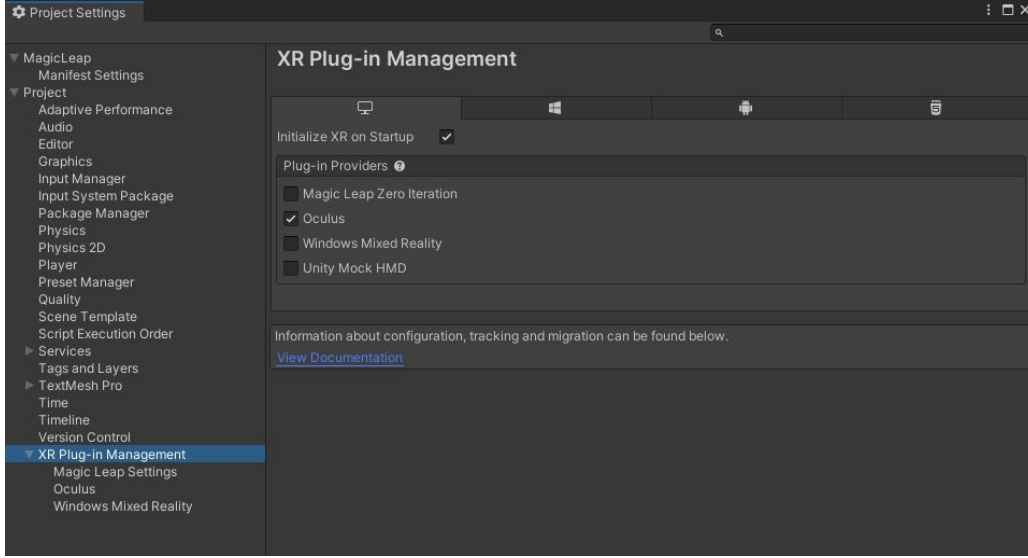
İlk VR uygulamasını tasarlamadan önce VR desteğinin etkinleştirilmesi gerekir. Bunun için,

- Edit → Project Settings menüsüne gidilir.
- “XR Plugin Management” seçeneğine tıklanır ve “Install XR Plugin Management” ile kurulum yapılır.



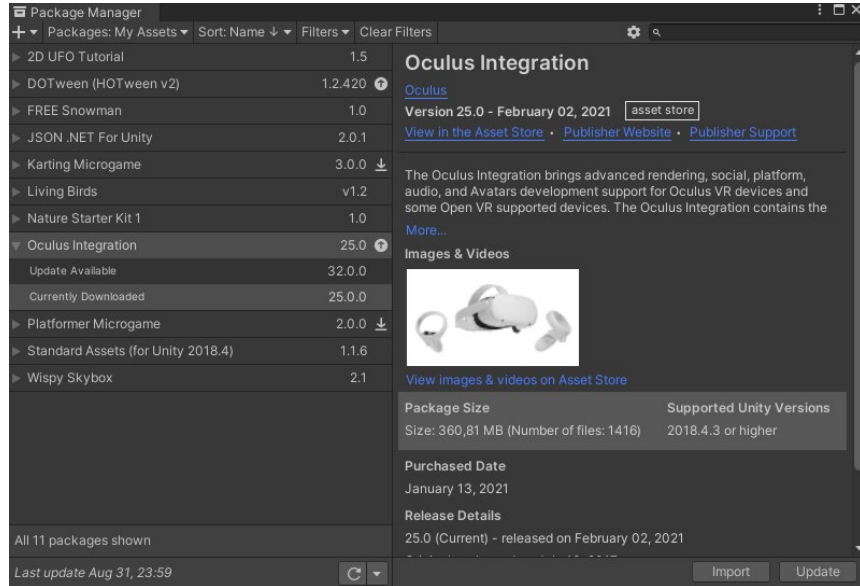
EK 2-Şekil 1 XR Plugin yöneticisinin kurulumu

- Ardından Windows sekmesine tıklanılır ve Unity’de VR desteği için Oculus seçeneğine tıklanarak Oculus XR eklentileri yüklenir.



EK 2-Şekil 2 Oculus eklentilerinin yüklenmesi

Bundan sonra Unity projesi açıkken Window → Asset Store menüsünden açılan mağaza sayfası yardımıyla “Oculus Integration” eklentisinin yüklenmesi gerekir. Unity mağazasından bir eklenti ya da paket yüklenebilmesi için bir Unity hesabı oluşturulmalıdır. “Oculus Integration” paketi bulunduğunda Unity penceresinde aç seçeneğiyle “Package Manager” penceresinin açılması sağlanır. Paket yöneticisi içinde “Oculus Integration” eklentileri yüklenir ya da yüklüyse güncelleştirmeler yapılır.



EK 2-Şekil 3 Unity'de paket yöneticisi