

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

**SANAL GİYSİ DENEME KABİNİ UYGULAMASI İÇİN  
KINECT İLE İNSAN MODELİ OLUŞTURMA VE  
KONTROLÜ**

**YÜKSEK LİSANS TEZİ**

**Muhammed KOTAN**

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM  
MÜHENDİSLİĞİ**

**Tez Danışmanı : Doç. Dr. Cemil ÖZ**

**Temmuz 2014**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ


**SANAL GİYSİ DENEME KABİNİ UYGULAMASI İÇİN  
KINECT İLE İNSAN MODELİ OLUŞTURMA VE  
KONTROLÜ**

**YÜKSEK LİSANS TEZİ**

**Muhammed KOTAN**

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM  
MÜHENDİSLİĞİ**


**Bu tez 10/07/2014 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.**

  
**Prof. Dr. Nejat YUMUŞAK**

**Jüri Başkanı**

  
**Prof. Dr. İsmail Hakkı  
CEDİMOĞLU**

**Üye**

  
**Doç. Dr. Cemil ÖZ**

**Üye**

## **TEŐEKKÜR**

Lisans ve yksek lisans eđitimim boyunca bilgi ve birikimlerini benimle paylaŐan ve tez alıŐmamın her aŐamasında tavsiyeleri ile yolumu aydınlatan deđerli danıŐman hocam Do. Dr. Cemil ÖZ'e, uygulama geliŐtirme ve test aŐamalarında bana sabırla yardım eden ve desteklerini esirgemeyen deđerli ev arkadaŐlarım ve dostlarıma sonsuz teŐekkrler.

Ayrıca tm yaŐamım boyunca maddi ve manevi arkamda duran, dua ve desteklerini esirgemeyen annem Muazzez KOTAN, babam Yunus KOTAN ve kardeŐlerim BuŐra KOTAN, Kevser KOTAN ve Nidanur KOTAN'a en özel ve iten Őkran duygularım ile...

# İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ .....	viii
ÖZET .....	xi
SUMMARY .....	xii
BÖLÜM 1.	
GİRİŞ .....	1
BÖLÜM 2.	
SANAL GERÇEKLIK, 3B MODELLEME ve OYUN MOTORLARI .....	6
2.1. Sanal Gerçeklik.....	6
2.1.1. Sanal gerçeklik donanımları.....	7
2.1.2. Sanal gerçeklik uygulama alanları .....	10
2.1.2.1. Eğitim alanında sanal gerçeklik .....	10
2.1.2.2. Oyun ve eğlence alanında sanal gerçeklik .....	11
2.1.2.3. E-ticaret alanında sanal gerçeklik .....	12
2.2. 3B Modelleme .....	13
2.2.1. Model tasarımlar .....	13
2.2.2. Modelleme işlemleri.....	14
2.2.3. 3B modellemenin avantajları .....	16
2.3. Oyun Motorları .....	16
2.3.1. Oyun motorlarının temel amaçları .....	19

### BÖLÜM 3.

SİSTEM İÇERİSİNDE KULLANILAN DONANIM ve YAZILIMLAR .....	20
3.1. Sistemde Kullanılan Donanımlar ve Özellikleri.....	20
3.1.1. MS Kinect .....	20
3.1.1.1. Kinect sensörü temel ve teknik özellikleri .....	21
3.1.1.2. MS Kinect görüntü izleme ve iskelet takip sistemleri ...	23
3.1.1.3. Kinect SDK ve diğer kütüphaneler .....	25
3.1.2. Sistemin geliştirilmesi ve test aşamasında kullanılan bilgisayarların teknik özellikleri.....	26
3.2. Sistemin Geliştirilmesinde Kullanılan Yazılımlar ve Özellikleri .....	27
3.2.1. MakeHuman .....	27
3.2.2. Blender modelleme yazılımı .....	29
3.2.3. Unity 3D.....	31
3.2.4. C# Programlama Dili .....	32

### BÖLÜM 4.

3B SANAL İNSAN MODELLEME ve GERÇEK ZAMANLI KONTROLÜ ....	34
4.1. MakeHuman Yazılımı ile Gerçekleştirilen İşlemler .....	37
4.1.1. 3B sanal insan modellerin oluşturulması .....	37
4.1.2. Modellerin rig yapısının belirlenmesi ve blender exchange (.mfx) dosyalarına dönüştürülmesi.....	38
4.2. Blender 3D ile Modellerin Düzenlenmesi .....	39
4.2.1. Mesh uyumsuzluklarının giderilmesi .....	39
4.2.2. UV Mapping özelliği ve giysi dokularının ayarlanması .....	40
4.2.3. Yüz ifadeleri oluşturma .....	41
4.3. Unity 3D Oyun Motoru ile Arayüz ve Script Tasarımı.....	42
4.3.1. Modelleme tabanlı işlemler .....	42
4.3.1.1. Sanal kabin bileşenlerinin oluşturulması .....	42
4.3.2. Script tabanlı işlemler .....	43
4.3.2.1. Kinect cihazı algılanması .....	46
4.3.2.2. Matematiksel formüller ve kod içerisinde kullanımı .....	46
4.3.2.3. Kullanıcı ve model entegrasyonu .....	49
4.3.2.4. Diğer bazı oluşturulan scriptler ve özellikleri .....	51

4.3.3. Uygulamanın Farklı Platformlara Göre Derlenmesi .....	56
BÖLÜM 5.	
SONUÇLAR VE ÖNERİLER .....	58
KAYNAKLAR.....	61
ÖZGEÇMİŞ .....	65

## SİMGELER VE KISALTMALAR LİSTESİ

API	:Application Programming Interface (Uygulama Programlama Arayüzü)
B2C	: Business to Costumer (Firmadan müşteriye)
CMOS	:Complementary Metal Oxide Semiconductor (Bütünleyici metal oksit yarı iletken)
CPU	: Central Processing Unit (Merkezi İşlem Birimi)
DLL	: Dynamic Link Library (Dinamik Link Kütüphanesi)
E-ticaret	: Elektronik ticaret
FBX	: Autodesk FBX Interchange File
FPS	: Frame per second (Saniyedeki çerçeve sayısı)
GB	: Gigabyte
GHz	: Gigahertz
GL	: Graphics Library (Grafik Kütüphanesi)
GUI	: Graphical User Interface (Grafiksel Kullanıcı Arayüzü)
HD	: High definition (Yüksek çözünürlük)
HMD	: Head/Helmet Mounted Display
IR	: Infrared (Kızılötesi)
KHz	: Kiloherz
KÖ	: Kızılötesi
LED	: Light Emitting Diode (Işık Yayan Diyot)
MHX	: Blender Exchange File
MS	: Microsoft
NUI	: Natural User Interface (Doğal Kullanıcı Arayüzü)
OM	: Oyun Motoru
OS	: Operating System (İşletim Sistemi)
PC	: Personal Computer (Kişisel Bilgisayar)

RAM	: Random Access Memory (Rastgele Eriřimli Bellek)
RGB	: Red Green Blue (Kırmızı Yeřil Mavi)
SDK	: Software Development Kit
SG	: Sanal Gerçeklik
SSD	: Solid State Disk (Katu halli disk)
USB	: Universal Serial Bus
V	: Volt
VGA	: Video Graphics Array
VRAM	:Virtual Random Access Memory (Sanal Rastgele Eriřimli Bellek)
WPF	: Windows Presentation Foundation
XML	: Extensible Markup Language (Geniřletilebilir İřaretleme Dili)
YGK	: Yazılım geliřtirme kiti
2B	: 2 boyutlu
2D	: 2 dimensional
3B	: 3 boyutlu
3D	: 3 dimensional



## ŞEKİLLER LİSTESİ

Şekil 1.1. Güncel bir çalışmadan alınmış fizik tabanlı giysi simülasyonu.....	3
Şekil 1.2. Uygulama arayüzünden örnek bir görünüm .....	5
Şekil 2.1. Sanal gerçeklik sistemi mimarisi .....	7
Şekil 2.2. HMD örneği .....	8
Şekil 2.3. Başlık ekran örneği 1 .....	8
Şekil 2.4. Başlık ekran örneği 2 .....	9
Şekil 2.5. DataGlove örneği .....	9
Şekil 2.6. Virtuix omni donanım birimi .....	10
Şekil 2.7. Sanal gerçeklik futbol uygulaması.....	12
Şekil 2.8. Utah çaydanlık modeli 3B grafik eğitiminde en çok kullanılan modellerden biridir .....	13
Şekil 2.9. Yüz için örnek bir 3B poligonal modelleme .....	14
Şekil 2.10. Render işlemi a)Render edilmemiş model b)Render işlemi (hesaplamalar) c) Render edilmiş model .....	15
Şekil 2.11. Unity 3D oyun motoru arayüzünden örnek bir sahne tasarımı .....	17
Şekil 2.12. Need for Speed oyunu için tasarlanmış bir yarış pisti .....	17
Şekil 2.13. Çok gelişmiş oyun motorları gerçekçi görünümler elde etmenizi sağlar	18
Şekil 3.1. Xbox 360 MS Kinect sensörü .....	20
Şekil 3.2. Kinect fiziksel ve pratiksel hareket izleme sınırları .....	22
Şekil 3.3. Kinect donanımsal bileşenleri .....	23
Şekil 3.4. İnsan iskelet yapısı üzerinde Kinect cihazının izlediği bölgeler.....	24
Şekil 3.5. Kinect hareket algılama algoritması .....	25
Şekil 3.6. MakeHuman 1.0 alpha 7 örnek kullanıcı arayüzü .....	28
Şekil 3.7. MakeHuman 1.0 alpha 7 model giysi ekranından bir görünüm.....	28
Şekil 3.8. MakeHuman 1.0 alpha 7 model saç tasarım ekranından bir görünüm .....	29
Şekil 3.9. Blender arayüzü ve örnek bir tasarım .....	30

Şekil 3.10. Unity Asset Store üzerinden sağlanan AngryBots isimli demo projeden örnek sahne görünümü .....	32
Şekil 4.1. Uygulamaya ait Akış Diyagramı.....	36
Şekil 4.2. Arayüz üzerinden erkek modelin yüz şeklinin ayarlanması .....	37
Şekil 4.3. Uygulama içerisindeki kadın modellerden birine ait mesh yapısı .....	38
Şekil 4.4. Örnek model iskelet yapısı.....	39
Şekil 4.5. Giysi uyumsuzluklarının giderilmesi.....	40
Şekil 4.6. UV Mapping ile ayakkabı nesnesinin doku düzenlemesi .....	41
Şekil 4.7. Modelin yüzünde değişik ifadeler oluşturulması .....	41
Şekil 4.8. Sanal kabin tasarım aşamasından bir görünüm.....	43
Şekil 4.9. Oyun nesnelerinin (game objects) yer aldığı Hiyerarşi sekmesi .....	43
Şekil 4.10. Uygulamaya ait Sınıf Diyagramı .....	45
Şekil 4.11. Kinect kontrolünü sağlayan kod bloğundan bir görünüm.....	46
Şekil 4.12. Parametrik olarak girilen eksen noktalarının eksenlere göre farkının Vector3 değişkenlerinde tutulması .....	47
Şekil 4.13. Parametrik olarak alınan iki eklem noktasının eksenlere göre farkını döndüren fonksiyon bloğu.....	47
Şekil 4.14. Yumuşatma Filtresinin uygulandığı fonksiyon .....	48
Şekil 4.15. İnsan modelin arayüz üzerinde y eksenine göre döndürülmesini sağlayan kod bloğu.....	48
Şekil 4.16. Yüz ifadesinin daha net görüntülenmesini sağlayan kamera konum ve rotasyon değişikliği komutları.....	49
Şekil 4.17. Kullanıcı ve model eklemlerinin eşleştirilmesi ve model eklemlerinin dizi elemanlarına atanması .....	50
Şekil 4.18. Arayüz üzerinde ilişkilendirilen model eklemlerini diziye atayan kod bloğu parçası .....	50
Şekil 4.19. Uygulama scriptlerinin yer aldığı Unity 3D script sekmesi.....	51
Şekil 4.20. Kameranın yüze yaklaşması ve ifadenin daha net görüntülenmesi .....	52
Şekil 4.21. Temel fonksiyonların yer aldığı ChangingRoom.cs adlı script içerisinde bir görünüm .....	53
Şekil 4.22. Yüz ifade değişikliğini sağlayan aşırı yüklenmiş fonksiyonlar .....	54
Şekil 4.23. Kullanıcı beden ölçümlerinin ekran üzerinde yansıtılması.....	55
Şekil 4.24. Script arayüzü üzerinde istenilen sayıda model uygulamaya eklenebilir	55

Şekil 4.25. Kullanıcı hareketlerini aynen uygulayan gülümseyen bir avatar .....	56
Şekil 4.26. Unity ile geliştirilen uygulamaların farklı platformlar için derlenmesi ...	57
Şekil 4.27. Kinect bağlı arayüzden genel bir görünüm .....	57
Şekil 5.1. Giysi ile model uyumsuzluğu .....	60

## ÖZET

Anahtar kelimeler: Sanal kabin, Sanal gerçeklik, Kinect, İnsan-Bilgisayar etkileşimi, Oyun motorları, 3B Modelleme

Bu çalışmada gerçek zamanlı 3B model kontrol yaklaşımı ve kullanıcıların sanal bir ayna önünde giysileri deneme olanağı bulabilecekleri sanal bir giyinme odası tasarlanması amaçlanmıştır. Oluşturulan sanal kabin içerisinde kullanıcının sanal temsilcisi olan 3B insan modeli oluşturulur. Sisteme entegre olan MS Kinect vb. yardımcı bir donanım kullanılarak, el hareketleri ile ekrandaki giysi listesinden kullanıcı tercih ettiği giysiyi seçer. Daha sonra ise bu seçilen 3B giysi sanal ayna uygulaması üzerinden 3B model üzerine otomatik olarak giydirilir. Seçilen 3B giysilerin model üzerinde tam ve uygun bir şekilde giydirilmesi için yerleştirme, ölçüm alma ve döndürme gibi işlemlerde insan eklemlerinin 3B koordinat bilgileri kullanılmıştır.

Geliştirdiğimiz algoritma ile kullanıcının vücut ölçümleri dikkate alınarak uygun olan small, medium, large veya xlarge giysi türü otomatik olarak seçilmekte ve bu bilgi ekran üzerinde görüntülenmektedir. Kullanıcı ve model arasındaki uyumsuzlukları önlemek ve giysilerin renk uyumunu anlayabilmek amacıyla ten rengi seçme özelliği ek olarak sanal kabin üzerinde sunulmuştur. İsteğe bağlı ayna seçenekleriyle de giysiler ve model üzerinde farklı bakış açılarından görüntü sağlanabilmektedir.

Bu çalışmada, kullanıcı hareketlerini izlemek ve bu hareketlerin 3B model üzerinde uygulanması, eklem koordinatlarının belirlenmesi, kıyafetlerin kullanıcıya göre boyutlandırılması ve insan vücudu ile giysiler arasında etkili efektler oluşturabilmek için Microsoft Kinect SDK ile bize sunulan özelliklerden ve Unity 3D ile oyun motorlarının sağladığı hızlı hesaplayabilme yeteneklerinden yararlanılmıştır.

Geliştirilen arayüz ile gerçekçi bir izlenim oluşturulmuş ve sistem çalışması farklı vücut ölçülerine sahip insanlar üzerinde test edilmiştir. Bu uygulama ile kişilerin fiziksel olarak giysileri denemelerinin zorluğu ve deneme kabinleri önünde oluşan uzun kuyruklar düşünülmüş bu sebeple sanal giysi deneme kabini uygulaması üzerinden online alışverişlere katkı sağlanması amaçlanmıştır. Uygulama Kinect desteklidir ve uygulama başlatıldığında sistem için gerekli bazı veriler otomatik olarak bu cihaz üzerinden elde edilmektedir. Bununla birlikte Kinect cihazı bulunmayan sistemlerde, uygulama grafik arayüzündeki düğmeler kullanılarak seçimler yapılabilir.

# **MODELLING AND CONTROL OF A HUMAN MODEL USING KINECT FOR A VIRTUAL DRESSING ROOM APPLICATION**

## **SUMMARY**

Key Words: Virtual try-on, Virtual reality, Kinect for Windows, Human-Computer interaction, Game engines, 3D Modelling

This thesis proposes a real time 3D virtual model controlling approach and a virtual dressing room application to enable users to try virtual garments on in front of a virtual mirror. A virtual representation of the user appears in a virtual changing room. The user's hand motions select the clothes from a list on the screen. Afterwards the selected virtual clothes appears on a humanoid model in the virtual mirror. For the purpose of aligning the 3D garments and shoes with the model, 3D locations of the joints are used for positioning, scaling and rotating.

By using our developed algorithm, small, medium, large or xlarge garment size is selected automatically and this information is shown on the screen. Then, we apply skin colour detection to handle the unwanted occlusions between the user and the model. Some optional mirror selection buttons make it possible to have multiple viewing angles on the model.

In this study, we benefit from the Microsoft Kinect SDK in order to follow the user's movements, move the avatar, read 3D information of joints' positions, coordinate the suitable clothe try-ons and provide depth sort effect for the human body and clothes. In order to use the rapid calculation attributes of game engines, we used Unity 3D Game Engine.

By developed interface of the application, the system works accurately and it is tested by different users. With this application it is aimed to contribute to online shopping and reduce the loss of time in the store shopping and effort of trying garments on in fitting rooms. When the application is started, the system uses Kinect and reads user's information. However, in systems without the Kinect device, application choices can be made using the buttons on the graphical interface.

## **BÖLÜM 1. GİRİŞ**

Başlangıçta sadece birkaç kurumu birbirine bağlamak amacıyla tasarlanan İnternet, artık milyonlarca kişinin bağlandığı dünya çapında devasa bir yapıya kavuşmuştur. İletişim teknolojileri ve İnternet kullanımı üzerindeki hızlı gelişmeler dünya üzerinde yeni bir ekonomik düzenin yaygınlaşacağını ortaya koymaktadır. Buna bağlı olarak; coğrafi sınırlar ortadan kalkmış, satıcı ile alıcının buluştuğu yer olan pazarlar farklı bir boyut kazanmış ve satıcılar tüm dünyayı müşteri kabul ederek, ticari faaliyetlerini web tabanlı sistemlere kaydırmışlardır [1].

Elektronik ticaret, internet çağının en önemli özelliklerinden biri haline gelmiştir. Online alışveriş, son kullanıcıların gereksinim duydukları mal ve hizmetleri internet üzerinden satın almak üzere işletmelerle bağlantı kurmaları şeklinde e-ticaretin firmadan tüketiciye (B2C) gerçekleştirilen boyutudur [2]. Araştırmacılar tüketicilerin satın alma tercihlerini ve karar verme süreçlerini, önerilen çeşitli modeller yardımıyla sürekli olarak tahmin etme ve açıklama girişiminde bulunmuşlardır [3].

Çevrimiçi müşterilerin sayısındaki artış, internet kullanıcılarının sayısındaki artıştan daha fazladır ve bu da, artık daha çok kullanıcının internetten alışveriş yapmaya sıcak baktığını göstermektedir. İnternet üzerinden alışveriş hacminin 2015 yılı itibariyle 119 milyar dolara ulaşacağı ABI Araştırma Şirketi tarafından ifade edilmiştir [4].

E-ticari Web siteleri mevcut her ürün için detaylı metinsel açıklamalar ve durağan bazı resimler içeren online bir katalog türü gibi görünmektedir. Bu yaklaşım ise alışkın olduğumuz alışveriş yönteminden bir hayli farklıdır. Müşteriler daima nesnelere ile etkileşimde bulunmak ve analiz etmek isteyeceklerdir. Basit bir katalog

ise interaktif hareket ve üç boyutlu bir görünüm gerektiren ürünler için yeterli olmayacaktır [5] [6].

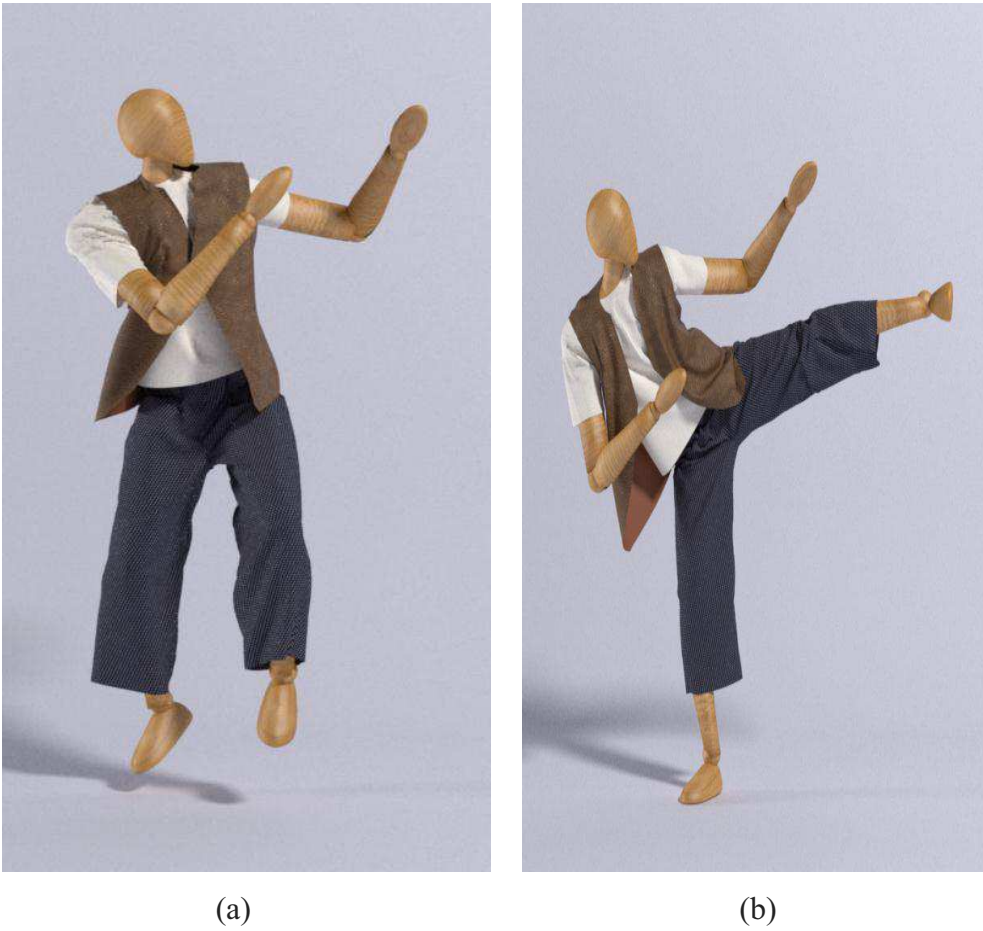
İnternet üzerinden alışveriş müşteriler için birçok avantajlar sunmaktadır. Bu avantajların başında ise karşılaşılan mağaza kalabalığından kurtulma ve kuyrukta bekleme gibi problemler, fiyatların genellikle düşük olması, alışverişin tüketiciye rahatlık ve kolaylık sağlaması, yedi gün yirmi dört saat alışveriş imkanı ve geniş bir ürün çeşidi sunması gibi nitelikler gelmektedir. Online alışverişin bu avantajlarına rağmen tüketicileri bu kanaldan alıkoyan bazı caydırıcı sebepler de mevcuttur. Bu sebeplerden bazıları istenilen ürünü deneme ve ürünün gerçek boyutlarını görme isteği gibi imkanların sanal alışverişlerde bulunmamasıdır [7].

Eurostats 2010 raporuna göre ise internet kullanıcılarının en çok satın aldığı ürün ve hizmetler %51'lik oranla seyahat ve konaklama hizmetleri, %46'lık oranla giyim ve spor ürünleri, %37'lik oranla ev ihtiyaçlarıdır [8]. Özellikle giyim sektörü üzerinde sayılan dezavantajlardan bazılarını ortadan kaldırmak ve kullanıcıya kıyafet deneme imkanı sunabilmek için online alışverişe destek sağlamak adına görüntü işleme konulu projeler olan sanal ayna sistemlerinden faydalanılabilir. Bu sistemler ile kullanıcılar fiziksel bağlantı sağlamadan ve zaman kaybı olmaksızın kıyafet deneme imkanı bulabilmektedirler.

Son zamanlarda literatürde sanal ayna çalışmaları yer almaya başlamıştır. Örneğin Wang ve ekibi online çanta satışı için bir sanal ayna sistemi üzerinde çalışmışlardır [9]. Zhou, takım arkadaşlarıyla birlikte ekran önünde kıyafet denenebilen bir sanal ayna sistemi oluşturmuştur [10]. Işıkdoğan, Kinect cihazı kullanarak gerçek zamanlı sanal giyinme odası uygulaması tasarlamıştır [11]. Giovanni ve arkadaşları ise iskelet takibinde kullanılan Kinect for Windows SDK ve OpenNI SDK'larını performans yönünden karşılaştırmış ve yine Kinect sensor ile HD kamera kullanarak sanal giyinme odası sistemi tasarlamışlardır [12]. Bu çalışmalar ile birlikte online alışveriş dışında da eğitim amaçlı kullanılan sanal ayna sistemleri de mevcuttur. [13] [14].

Sanal ayna sistemlerinin çoğunda 2B giysiler, kamera ile ekranda beliren kullanıcı üzerine oturtulmaktadır. Yalnızca 3B bir ortam ve gerçek 3B nesnelere gerçekçi bir

şekilde alışveriş ortamı oluşturabilirler [15]. Sanal gerçeklik teknolojisi e-ticareti geliştirmek için gerekli değerler arasına eklenebilir. SG son zamanlarda uygulamalar içerisinde büyük bir kullanım alanı bulmasına rağmen (örneğin, eğitim, simülasyon, tıp ve eğlence) e-ticaret deneyimini inceleyen daha heyecan verici bir sanal gerçeklik amaçlayan uygulama sayısı pek azdır. Gerçekçi bir 3B arayüz, ürünler ile etkileşim sağlanan, diğer insanlarla tanışma sunulan (temsilci avatarlar ile), seslerin duyulabildiği vb. bir alışveriş merkezine kullanıcıyı dahil edebilir ve böyle bir yöntem ile insanlar gerçeğine benzer deneyimler edinebilirler [15].



Şekil 1.1. Güncel bir çalışmadan alınmış fizik tabanlı giysi simülasyonu [16]

Bu çalışmanın amacı, dinamik olarak biçimlendirilen ve vücut hareketleri ile kontrol edilebilen 3B bir insan modeli oluşturmaktır. Sanal bir giyinme kabini oluşturularak ürünler kullanıcı tercihine göre bu kabin içerisinde dinamik olarak gösterilmiştir ve kullanıcının kendini gerçekçi sentetik bir ortam içerisinde hissetmesi sağlanmıştır. Kullanıcının, giysileri kendi ölçülerine sahip bir 3B insan model üzerinde

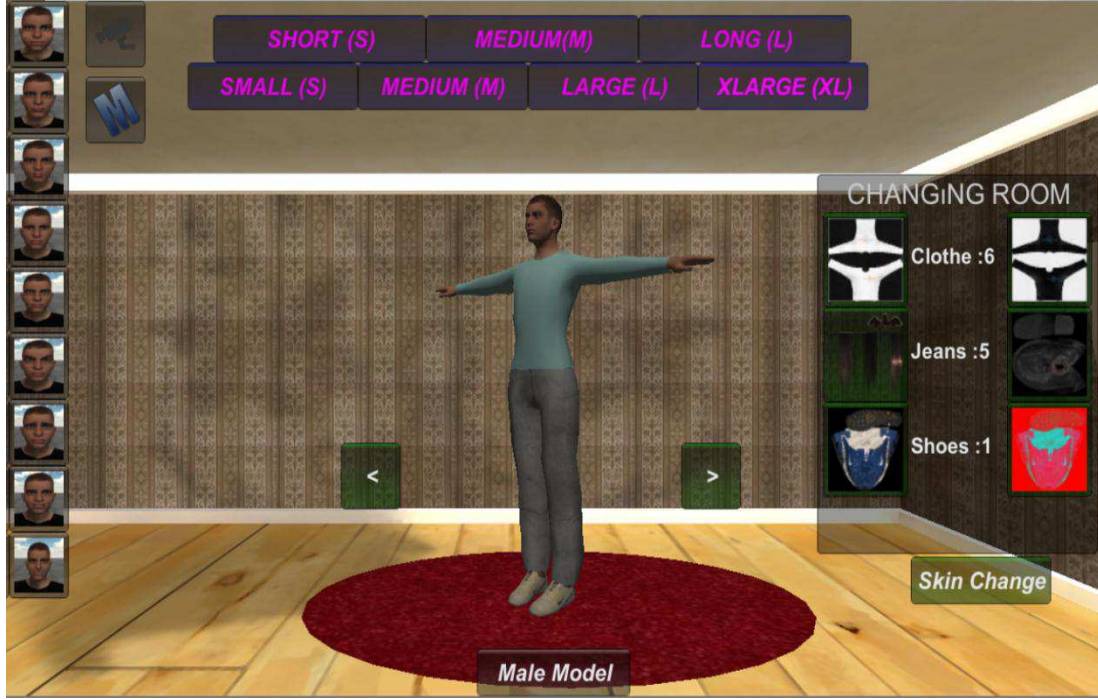


görebilmesi sağlanmıştır. Uygulama içerisinde Kinect sensöründen yararlanılmıştır. Kinect; RGB kamera, derinlik algılayıcı kameralar ve mikrofonlardan oluşan bir sensördür. Ayrıca insan iskeletine ait 20 adet eklem noktası bu sensörler ile belirlenebilmekte ve hareketleri izlenebilmektedir. Tez çalışmasında kullanıcı hareketlerini izlenmesinde, bu hareketlerin model üzerinde eş zamanlı gerçekleştirilmesinde, kullanıcı eklem noktaları arasındaki uzaklığın hesaplanmasında ve beden ölçüsünün hesaplanması algoritmalarında Kinect verilerinden yararlanılmıştır. Program geliştiriciler için iskelet beden takip yöntemleri içeren uygulama arayüz programları geliştirilmesi üzerinde yoğun çalışmalar mevcuttur. Güçlü ve Pratik iskelet takip algoritmalarından dolayı tez uygulamasında MS Research Kinect SDK kullanımı tercih edilmiştir.

Uygulama ve kod tasarımı Unity 3D oyun geliştirme motoru üzerinde C# programlama dili kullanılarak yazılan kodlar ile gerçekleştirilmiştir. Uygulama içerisindeki yaklaşımımız aşağıdaki gibi özetlenebilir:

- a) 3B giysileri sanal kabin içerisine yerleştirmek
- b) Sensörler üzerinden alınan verilere göre 3B insan modeli boyutlandırmak
- c) Seçilen giysileri model üzerine giydirmek
- d) Ten rengi ve cinsiyet seçimi
- e) Modeli isteğe bağlı olarak döndürme
- f) Model üzerinde farklı bakış açılarından görüntü almak için aynalar kullanmak

Sistemin çalışması esnasında kullanıcı Kinect sensörlerinin karşısına geçer ve sensörler tarafından algılanmayı bekler. Uygulama kullanıcı ile model eklemlerini eşleştirerek modelin kullanıcı hareketlerini taklit etmesini sağlar. Ardından kullanıcı el hareketleri ile kabin üzerinde mevcut giysi kataloğundan tercih ettiği giysiyi seçer ve seçilen giysi model üzerine otomatik olarak giydirilir. Uygulama arayüzünden örnek bir görünüm Şekil 1.2’de gösterilmiştir.



Şekil 1.2. Uygulama arayüzünden örnek bir görünüm

İkinci bölümde geliştirme ve tasarım aşamalarında kullanılan sanal gerçeklik, 3B modelleme ve oyun motoru teknolojileri hakkında bilgi verilmiştir.

Üçüncü bölümde uygulamada kullanılan yazılım ve donanımlar anlatılmıştır. Uygulamada donanım olarak kullanılan MS Kinect for Windows sensörü tanıtılmış ve sistem geliştirilme ve test aşamalarında kullanılan bilgisayarların özellikleri sunulmuştur. Yazılım kısmında ise modelin tasarlandığı MakeHuman, model üzerinde düzenlemelerin yapıldığı, doku ve desenlerin eklendiği Blender 3D ve uygulama arayüzünün geliştirildiği Unity3D Oyun motoru anlatılmıştır. Kodlarının geliştirildiği C# programlama dili hakkında ise kısa bir bilgi verilmiştir.

Dördüncü bölümde bu tez çalışması kapsamında sistemin tasarım ve uygulama aşamaları üzerinde durulmuş ve geliştirilen algoritmalar açıklanmıştır.

Beşinci bölümde ise bu çalışmanın sonuçlarına değinilmiştir.

## **BÖLÜM 2. SANAL GERÇEKLİK, 3B MODELLEME ve OYUN MOTORLARI**

Bu bölümde sanal gerçeklik, 3B modelleme ve oyun motoru teknolojileri hakkında bilgi verilecektir.

### **2.1. Sanal Gerçeklik**

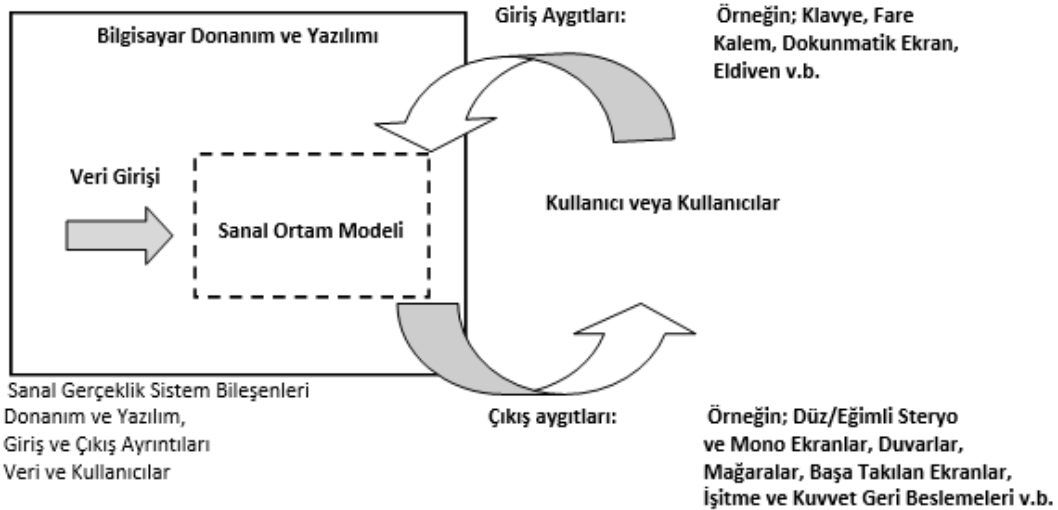
Sanal gerçeklik, bilgisayar ortamında oluşturulan 3 boyutlu resimlerin ve animasyonların teknolojik araçlarla insanların zihinlerinde gerçek bir ortamda bulunma hissini vermesinin yanı sıra, ortamda bulunan bu objelerle etkileşimde bulunmalarını sağlayan teknoloji olarak tanımlanabilir. Bu şekilde algılanması sağlanan ortama ise sanal ortam denilir. “Yapay gerçeklik”, “sanal dünyalar”, “sanal ortamlar” gibi farklı isimler de verilmektedir.

Sanal gerçeklik ortamı, gözlemcinin gerçek ortamdaki zihinsel olarak koparak yapay üç boyutlu dünyanın (simülasyonun) içine girdiği, orada bulunma, dolaşma, nesnelerin yerlerini ve özelliklerini değiştirme gibi çeşitli etkileşimlerde bulunduğu ve bu etkileşimler sonucu gerçek dünyadaki gibi duyuşsal tepkiler aldığı bir ortamdır [17].

SG’yi diğer uygulamalardan ayıran temel üç özelliği vardır [18]. Kullanıcılara gerçekmiş hissi vererek dinamik bir ortamda karşılıklı etkileşim sağlanan bir benzetim modelidir. Katılımcılara gerçekmiş hissi vermesi belki de bu özelliklerden en önemlisidir. SG sistemlerinde temel amaç kişinin gördüğü görüntüyü gerçek zannetmesi ve kendisini bu görüntüye kaptırmasıdır. Kullanıcı, bilgisayarların oluşturmuş olduğu ortamda istediği yere gidebilmeli, yani kontrolün kendi elinde olduğunu hissetmelidir. Bu, ancak karşılıklı etkileşimle sağlanabilir. Sanal dünya oluşturmak için kullanılan cihazlar, insanın beş duyusunu uyaracak şekilde tasarlanmıştır. SG uygulamaları, minimum seviyede SG gözlüklerini içermelidir.

Böylece, kullanıcı gideceği noktayı ve baktığı doğrultuyu etkileşimli olarak belirleyebilecektir. Bir takım pozisyon izleyici aletlerin bu tür gözlüklerle kombine edilmesiyle bilgisayar tarafından oluşturulmuş ortamlarda yürüme sağlanabilir. Görme duyusu, şüphesiz sanal ortamdaki objelerin yerlerini değiştirme, dokunma, fiziksel özelliklerini hissetme ve çevredeki sesleri işitme duyularını da kapsayacak şekilde genişletilebilir [19]. Bu gibi olanaklar, DataGloves olarak adlandırılan özel eldivenleri, üç boyutlu ses kavramını ve benzeri teknolojik aletleri gündeme getirmektedir [20]. Ses, inandırıcılık açısından önemi göz önüne alınarak her yönden farklı derinliklerde verilebilir.

Bir SG sisteminde, gerçek zamanda hareket edebilmek için güçlü bilgisayarlara gerek vardır. Burada bilgisayarlar, birer görüntü yaratıcı (Reality Engine) olarak görev yapacaktır. Objelere ait üç boyutlu geometrilere ilişkin veri tabanlarına ulaşmaya olanak tanıyacak ve şu ana kadar bahsi geçen tüm duyu algılama aletlerini ve donanımlarını koordine edecek uygulama programlarını çalıştıracaktır [19].



Şekil 2.1. Sanal gerçeklik sistemi mimarisi [21]

### 2.1.1. Sanal gerçeklik donanımları

Sanal gerçeklik kullanıcılar vücutlarına giydikleri özel amaçlı aygıtlar ile veya çevre birim aygıtlarını kullanarak sanal ortamda etkileşim halinde olurlar.

Head mounted display veya Helmet mounted display, her ikisinin kısaltılmışı olarak HMD, tek bir göz (monocular HMD) veya her iki göz önünde küçük bir optik ekranı bulunan görüntüleme cihazıdır.



Şekil 2.2. HMD örneği [22]

Başa takılan ekranlar ile kullanıcının baş hareketleri algılanarak baş hareketleri ile döndürme algılandığı esnada görüntünün değişmesi sağlanır. Her göz için ayrı bir göz ünitesi ve seslerin algılanmasını sağlayan bir hoparlör bulundurlar.



Şekil 2.3. Başlık ekran örneği 1 [22]

Sanal gerçeklik uygulamaları genellikle sadece görsel ve işitselken yeni tür bazı prototipler ile koku, nem ve dokunmak gibi duylara da hitap edilebilecektir. Ortam ses düzeyi istenildiği gibi ayarlanabilen kask aynı zamanda burun altına yerleştirilen tüplerle ortama uygun kokuları da kimyasal karışımlarla verebilecek şekilde tasarlanmaktadır.



Şekil 2.4. Başlık ekran örneği 2 [23]

Parmakların hareketlerini algılamak için çeşitli sensörler barındıran veri eldivenleri kullanılır. Eldivenin pozisyon ve yönünü takip etmek için çeşitli aygıtlar kullanılabilir.



Şekil 2.5. DataGlove örneği [22]

Vücut kitleri ve sensörleri sayesinde bedensel hareketler sanal ortama aktarılmaktadır. Günümüzde oyun sektöründe kullanılmaktadır. Şekil 2.6'da gösterilen yeni nesil Virtuix Omni cihazıyla kullanıcının beden hareketleri ile oyun karakterinin kontrolü sağlanmıştır. Kullanıcının sanal dünyada kendi ayakları ile dolaşmasına olanak sağlayan bu cihaz, sanal dünyada oturarak elde edilemeyecek deneyimler kazanmanıza yardımcı olabilir [24].



Şekil 2.6. Virtuix omni donanım birimi [24]

### 2.1.2. Sanal gerçeklik uygulama alanları

SG'nin ilk uygulama bulduğu alanları video oyunları ve eğlence dünyası oluşturmaktadır. Pek çok bilim kurgu filminde SG teknolojisinin değişik unsurları önemli roller üstlenmiştir. Genel prensiplerine bakıldığı zaman ise her türlü veri ve uygulama için avantajları bulunmaktadır. Sanal gerçekliğin içerisinde bulunduğu uygulama alanları çok olmakla birlikte bu alanlardan birkaç tanesini alt başlıklar halinde inceleyebiliriz.

#### 2.1.2.1. Eğitim alanında sanal gerçeklik

Örgün eğitim sistemleri SG'den birçok yarar sağlayabilirler. Özellikle matematik, fen, tıp, askeri ve havacılık eğitiminde kullanılması, eğitimin kalitesi açısından son derece önemlidir. SG sayesinde öğrencilerin, sadece bilimsel gerçekleri daha hızlı ve iyi öğrenmeleri değil aynı zamanda deneyerek gerçek deneyim edinmeleri de sağlanabilmektedir.

Sanal gerçeğin eğitimdeki kullanımına yönelik çok miktarda çalışmalar yapıldığını görmekteyiz [25]. Bu çalışmalara örnek olarak;

- i. Özel Eğitim alanında motorlu tekerlekli sandalye kullanan çocukların karşılaşılabilecekleri tehlikeli durumları sanal gerçeklik ortamı yoluyla yaşamaları sağlanarak eğitimleri gerçekleştirilmektedir.
- ii. Mimarlık eğitimi alanında bina içerisinde engelli merdivenlerinin nasıl olması gerektiğinin test edilmesi sağlanmaktadır.
- iii. Tarih eğitimi alanında tarih odası adı verilen sanal gerçeklik ortamları ile öğrenciler tarihi olaylara tanıklık edebilmektedirler.
- iv. Fen ve Matematik alanında Kimya öğrencileri 3 boyutlu moleküllerin özelliklerini inceleyebilmekte, Fizik bölümünde fizik kurallarına aykırı durumlar oluşturulabilmekte ve buradaki etkiler gözlemlenebilmektedir. Matematik alanında ise görsel olarak anlaşılması zor olan grafik ve denklemler daha kolay anlaşılır bir hale getirilmeye çalışılmaktadır.
- v. Tıp Eğitimi alanında sanal kadavra yoluyla tıp eğitimi alan öğrenciler kadvralar üzerinde sayısız denemeler yapabilmektedirler.
- vi. Askeri ve Havayolu Endüstrisi alanında pilot eğitimlerinde sanal gerçeklik simülatörleri kullanılmaktadır. Aynı şekilde askeri amaçlı birçok sanal savaş oyunları tasarımı yapılmaktadır [25].

### **2.1.2.2. Oyun ve eğlence alanında sanal gerçeklik**

Oyun dünyası sanal gerçeklikle daha çekici bir hal almıştır. Pek çok bilim kurgu filminde de sanal gerçeklik teknolojisinden yararlanılmıştır. 3D gözlükler ve hareketli koltuklar ile izleyiciler kendilerini filmin içerisindeymiş gibi hissetmektedirler.





Şekil 2.7. Sanal gerçeklik futbol uygulaması [26]

### 2.1.2.3. E-ticaret alanında sanal gerçeklik

Elektronik ticaret siteleri, SG tekniklerini bir pazarlama aracı olarak kullanabilmektedir. Gözlük ve algılayıcılar sayesinde kullanıcıların sanal ortamda daha gerçekçi ve eğlenceli bir sörf yapabilmelerine olanak sağlayan bu sistemin zamanla elektronik sanal market uygulamalarında da yaygın olarak kullanılması beklenmektedir. Tabii ki burada en belirleyici faktörler; internet bağlantısının hızı, SG kullanımına uygun bilgisayar sistemlerinin kalitesi ve maliyetlerdir. Bu yüzden günümüzde SG sistemlerinin internet tabanlı uygulamalarda kullanımı, oyun endüstrisinde olduğu kadar yaygın değildir [19].

Mevcut bazı online alışveriş siteleri üzerinde kullanıcılar web kamerası üzerinden sanal giyinme deneyimi yaşayabilmekte [27] veya kameralarını bir ayna gibi kullanarak seçilen gözlüğün kendileri üzerinde nasıl durduğunu test edebilmektedirler [28].

Bu tezin de konusunu oluşturduğu müşterinin istediği ürünü fiziksel olarak giymeden bilgisayar ekranında kendi beden ölçülerinde mevcut bir insan model üzerinden deneme ve görme imkanı sunulması, sanal gerçekliğin e-ticaret alanındaki örnekleri arasında sunulabilir.

## 2.2. 3B Modelleme

3 boyutlu modelleme bilgisayar ortamında bir nesnenin matematiksel olarak vektörel bir formatta oluşturulmasıdır. Ortaya çıkan ürün model alınan nesnenin geometrik bir temsilidir ve 3 boyutlu modeldir. 3D model olarak anılır. Ancak sadece modelin oluşturulması, amaç görselleştirme olduğu zaman yeterli olmaz. Modelin görselliği, model görselleştirme (Rendering) diye adlandırılan bir süreçten daha geçirilerek artırılır veya model kullanılarak çeşitli animasyonlar oluşturulabilir [29]. Modeller aynı zamanda 3D Yazıcılar kullanılarak fiziksel olarak da elde edilebilirler.

Modeller otomatik veya manuel olarak oluşturulabilirler. Bir veri setinden oluştukları için (noktalar ve diğer bilgiler) 3B modeller elle, algoritmik olarak (prosedürel modelleme) veya tarayıcı ile taranarak oluşturulabilirler.

### 2.2.1. Model tasarımlar

3B modeller genel anlamda 3B Grafiklerin bulunduğu her yerde kullanılırlar. Bu modeller 3 boyutlu uzayda üçgen, çizgi, kavisli yüzey vb. çeşitli geometrik şekiller ile birbirine bağlanmış noktalar kümesini kullanan 3B bir nesneyi temsil ederler.



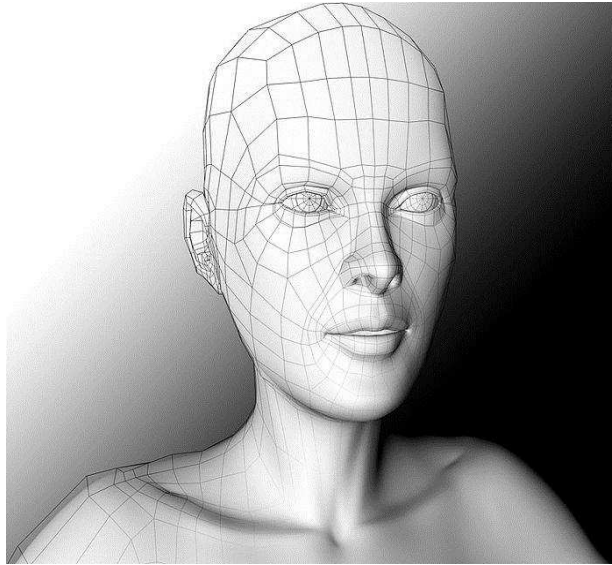
Şekil 2.8. Utah çaydanlık modeli 3B grafik eğitiminde en çok kullanılan modellerden biridir [30]

Günümüzde 3B modeller geniş bir yelpazede kullanılmaktadır. Film endüstrisinde, animasyonlu veya gerçek yaşamlı filmlerin karakter ve nesnelere olarak kullanılırlar. Oyun sektöründe oyun içerisindeki varlıklar olarak kullanılırlar. Tıp alanında

organların detaylı modelleri şeklinde kullanılırlar. Bilim alanında kimyasal bileşenlerin geniş ve detaylı modelleri olarak kullanılırlar [30]. Mimari alanda Mimari Model Yazılımları içerisinde çizilen bina ve arazilerin gösteriminde kullanılırlar. Mühendislik alanları içerisinde diğer kullanımlarla beraber araç, cihaz ve yapıların tasarımlarında kullanılırlar. Son zamanlarda ise yer bilimciler de 3B jeolojik modeller inşa etmekte kullanmaya başlamışlardır.

### 2.2.2. Modelleme işlemleri

Bir modelin temsilinde kullanılan üç popüler yöntem mevcuttur. Bunlar: Poligonal, Curve ve Digital Sculpting yöntemleridir. Bu yöntemlerden poligonal modelleme, 3 boyutlu uzayda köşe (vertice) denilen noktalar ve çizgiler ile bağlanarak poligonal yüzey oluştururlar. Bugünkü modellerin büyük bir çoğunluğu esnek oldukları ve bilgisayarlar tarafından hızlı işlenebildikleri için dokulu poligonal modeller olarak tasarlanmaktadır. Ancak poligonlar düzlemsel yapıya sahiptirler.



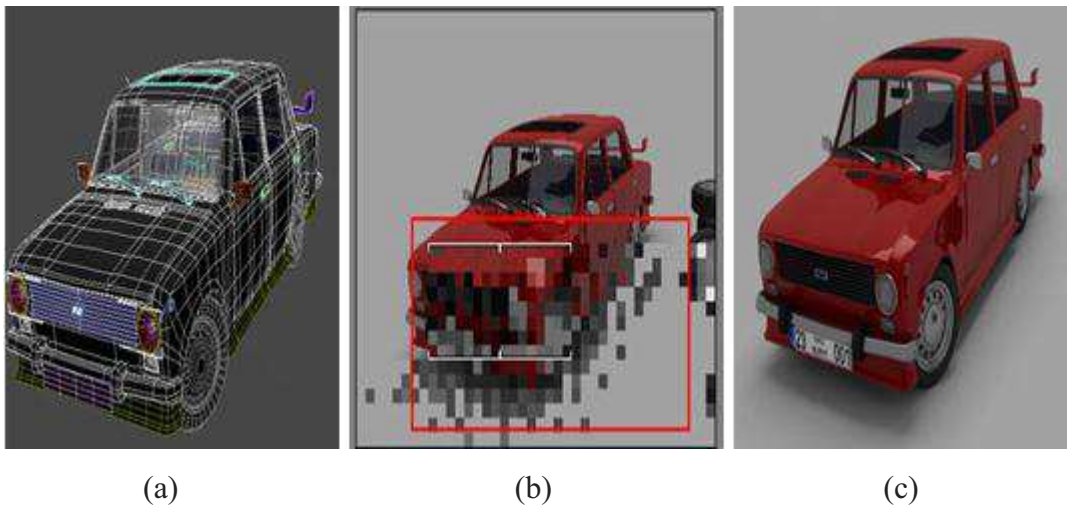
Şekil 2.9. Yüz için örnek bir 3B poligonal modelleme [30]

*Kaplama:* 3 boyutlu bir model oluşturulduğunda genellikle renksiz ve modeli oluşturan geometrik şekillerden oluşurlar. Modeli gerçeğe uygun hale getirebilmek için üzerine desen ve doku kaplamaları yapılır [29].

*Aydınlatma* : Model üzerinde desen ve doku kaplamaları yapıldıktan sonra ortamdaki ışık kaynakları ayarlanarak görüntüyü oluşturacak elemanların parametreleri ayarlanır [29].

*Render*: 3B alanında en çok karşımıza çıkan terimlerden biridir. “sunmak”, “derlemek” manasına gelen render, nesnelerin tüm hesaplamalarının yapılarak son gösterim aşamasına gelmesi demektir. Sahneler çizim aşamasındayken belirgin bir biçimde gerçek görüntüyü yansıtamazlar. Render işlemi sürekli hesaplamalar gerektiren bir işlemdir. Bazı sahnelerin render işlemi saatleri bulabilir ve saatler süren bir sahnenin gerçek zamanlı olarak hem çizilmesi hem hesaplanması oldukça maliyet gerektirir [29].

Render sahnedeki objelerin, ışıkların ve yansımalarının birbirinden etkilenerek ortama yayılan ışık kümelerinin hesaplanmasıdır. Render motoru başta sahnedeki objeleri, ışık kaynaklarını ve efektleri tespit eder ve hesaplamaya başlar. Örnek olarak sağ taraftan vuran bir ışık kaynağının nereye ışığını göndereceğini ve sonunda nerelerde gölge ve bu ışığın parlaklık oluşturacağını hesaplar ve çizimi sunuş haline getirir [29]. Şekil 2.10, bu aşamalar için bir örnek temsil etmektedir.



Şekil 2.10. Render işlemi a) Render edilmemiş model b) Render işlemi (hesaplamalar) c) Render edilmiş model [29]

### 2.2.3. 3B modellemenin avantajları

*Esnelik*, deęişikliklerin daha çabuk render işlemine tabi tutulmasıyla görüntüleme animasyon eklenmesi ve açıları deęiştirme yeteneęi sağlaması.

*Render(Hesaplama) kolaylığı*, mental bir görselleştirme ya da tahmine dayalı yöntemler yerine otomatik hesaplama ve foto gerçekçi efektler işleme kolaylığı.

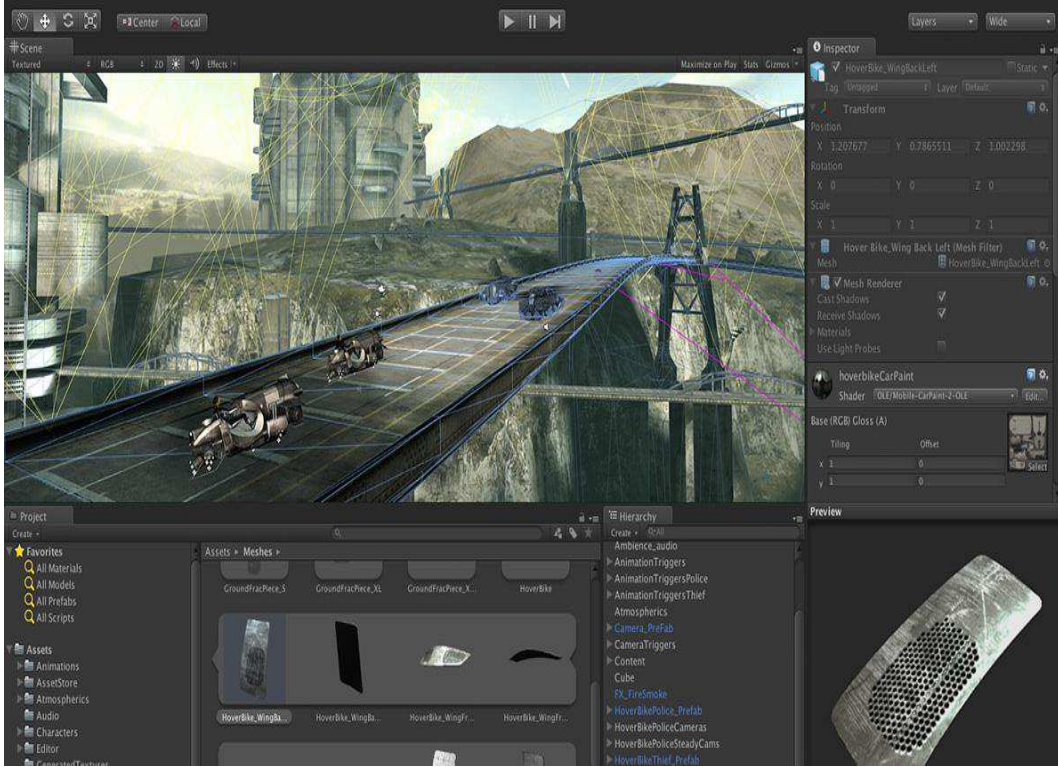
*Eksiksiz fotogerçekçilik*, yanlış yerleştirme, abartıya kaçma veya görsel bir efekti eklemeyi unutmak gibi insan hatalarını azaltması [31].

### 2.3. Oyun Motorları

Oyun motoru, bilgisayar video oyunlarının ve gerçek zamanlı grafik işleyen dięer interaktif uygulamaların çekirdek yazılım bileşenidir. En temelde yer alan birçok teknolojileri sağlar, geliştirme aşamasını kolaylaştırır ve genellikle oyunların oyun konsolları ve Linux, Mac OS X ve Windows gibi işletim sistemleri vb. bir çok platform üzerinde çalışmalarını sağlar.

Başka bir ifadeyle oyun motorları, geliştirilen oyunun tüm yazılımsal ve donanımsal ilişkilerini düzenleyen, oyunların alt yapısını oluşturan görsel ve işitsel kabiliyetleri ile yapay zeka gibi kabiliyetleri içerisinde barındıran, kullanıcılarına kişiselleştirme imkanlarını sağlayan yazılımlardır [32].

Bu yazılımlar, çok sayıda kütüphane dosyalarını içeren yapılardan oluşurlar. Bu kütüphanelerin içinde bilgisayar programlama dilleri ile tanımlanmış fonksiyonlar ve sınıflar gibi çok sayıda veri bulunmaktadır. Oyun tasarımcısı oyun motorunu kullandığında, başkası tarafından önceden tanımlanmış olan bu verileri kullanarak oyununu geliştirir [33]. Oyun motorları, önceden kodlanarak tanımlanmış olan verilerle tekrar kod yazmayı ortadan kaldırarak önemli ölçüde zaman tasarrufu sağlar.



Şekil 2.11. Unity 3D oyun motoru arayüzünden örnek bir sahne tasarımı [34]

Oyun motorları tekrar kullanılabilir yazılım bileşenlerine ek olarak görsel geliştirme araçları da sağlarlar. Bu araçlar genellikle bir geliştirme ortamı ile birlikte sunulur ve daha kolay, hızlı oyunlar geliştirilmesini sağlarlar.



Şekil 2.12. Need for Speed oyunu için tasarlanmış bir yarış pisti [35]

Bir oyun motorundan en çok beklenen şey grafik ve performanstır. Ayrıca oyun motorları fiziksel fonksiyonlar da sunarlar. 3 boyutlu oyunlarda varillerin devrilmesi, yuvarlanması, karakterinizin arazi üzerinde duruşu bile bir takım fizik fonksiyonları sonucu oluşur. Çok gelişmiş oyun motorları gerçekçi fizik özellikleri sağlamaktadır. Örneğin duvara gelen merminin duvardaki betonu parçalaması gibi gerçekçi fiziksel özellikler sağlayacaktır. Performans, en çok beklentiyi oluşturan etkidir. Bir oyun motorunun iyi bir optimizasyon ve performansa sahip olması gerekmektedir [36].



Şekil 2.13. Çok gelişmiş oyun motorları gerçekçi görüntüler elde etmenizi sağlar [37]

Oyun motorlarına bazen “oyun orta katman yazılımı” (game middleware)’da denmektedir. Oyun motorları genellikle gerçek zamanlı grafiksel gereksinimleri olan demo sürümleri piyasaya sürmek, görsel mimari, eğitim simülasyonları ve ortam modelleme gibi diğer uygulama türleri için de kullanılırlar [31].

Oyun motoru kullanılarak oyunlar geliştirmek ‘Doom’ oyunundan sonra popüler olmaya başlamıştır. Oyun geliştiriciler var olan oyunların çekirdeklerini değiştirerek oyunlara yeni modeller, senaryolar, ses efektleri ekleyebildikleri gibi oyunun model, senaryo ve ses efekti gibi unsurlarını da değiştirebilirler. Örneğin ‘Counter Strike’, ‘Team Fortress’, ‘TacOps’, ‘Strike Force’ ve ‘Quake Soccer’ oyunları temelde

'Quake' oyun motorunu kullanan, birçok farklı oyun motoru ile geliştirilmişlerdir [38].

Bilinen en popüler oyun motorlarına:

- BlitzTech
- CryEngine 3
- GameMaker
- HeroEngine
- Infernal Engine
- Unity 4
- Unreal Engine 4

örnek olarak verilebilir.

### **2.3.1. Oyun motorlarının temel amaçları**

OM temel amaçları arasında şu özellikler sayılabilir:

- a) Temel teknolojiler,
- b) 2B ve 3B grafikler,
- c) Bir fizik motoru ve çarpışma algılayıcı (ve çarpışma tepkisi),
- d) Ses özellikleri,
- e) Script geliştirme,
- f) Animasyon oluşturma,
- g) Yapay zeka,
- h) Ağ yapılandırması,
- i) Giriş, çıkış işlemleri, bellek yönetimi, thread işlemi ve sahne grafiği sağlamak.
- j) Geliştirme işlemini kolaylaştırmak.
- k) Birçok platform üzerinde çalışmak [31].



## **BÖLÜM 3. SİSTEM İÇERİSİNDE KULLANILAN DONANIM ve YAZILIMLAR**

### **3.1. Sistemde Kullanılan Donanımlar ve Özellikleri**

#### **3.1.1. MS Kinect**

Kinect, Microsoft tarafından geliştirilmiş olan hareket algılayıcı bir sensördür. Kinect'in göze çarpan en büyük özelliği insan hareketlerini algılaması olmuştur. Yani oyuncu hiç bir kontrol aracı kullanmadan hareketleri ile oyunu oynayabilmektedir. Microsoft 2010 yılında bu teknolojiyi piyasaya sürmüştür.



Şekil 3.1. Xbox 360 MS Kinect sensörü [39]

Microsoft tarafından “You are the controlller” [39], “Kontrol sizsiniz” sloganıyla duyurulan bu teknoloji, kullanıcılara önceki aşına oldukları gerçek zamanlı etkileşim deneyimlerinden farklı olarak, herhangi bir aygıta dokunmaksızın doğal bir etkileşim yöntemi sağlamıştır.

### 3.1.1.1. Kinect sensörü temel ve teknik özellikleri

Kinect cihazı, içerisinde bir RGB kamera, 3B derinlik algılayıcı sensörler ve dört adet mikrofon bulundurmaktadır. Ayrıca alt kısımda yukarı ve aşağı yönlü dönmesini sağlayan bir eğim motoru bulunmaktadır. RGB ve derinlik kameralarının her ikisi de 640x480 piksel çözünürlükte olup saniyede 30 frame (FPS) işlemektedir.

Kinect teknik özellikleri aşağıdaki gibi listelenmektedir [40];

RGB kameranın özellikleri ;

- 1.3 megapixel renkli kamera
- Micron MT9M001
- IR (Infrared) geçiren filtre ile donatılmış
- 32-bit renk ve 30 frame/sn
- 640 x 480 pixel resim çözünürlüğüne sahiptir.

Sensör;

- Renk ve derinlik algılama lensleri
- Ses mikrofon düzeni
- Sensör ayarlaması için tilt motoru

Görüş Alanı;

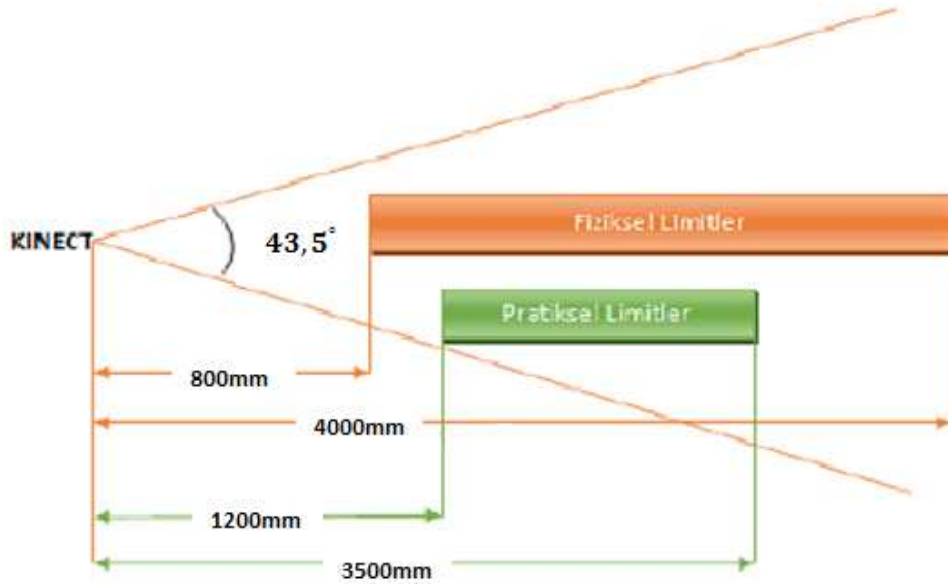
- Yatay görüş alanı: 57 derece
- Dikey görüş alanı: 43 derece
- Fiziksel Tilt alanı: 27 derece
- Derinlik sensörü alanı: 1.2m - 3.5m

Data Akışı;

- 320x240 16-bit derinlik - 30 frame/sn
- 640x480 32-bit renk - 30 frame/sn
- 16-bit audio - 16 kHz

Derinlik sensörü Kızılötesi(IR) ışınlar yayan bir IR projektör ve bu ışınları alıp her bir pikselin 3B derinlik verisini oluşturan toplayıcı metal oksit yarı iletken bir sensörün birleşiminden oluşmaktadır. KÖ ışınlarla derinlik verisini anlama özelliği cihazı ışık koşullarından bağımsız hale getirerek karanlıkta bile çalışabilir kılmaktadır. Ancak, kızılötesi bozulmalarından dolayı gün ışığında çalışamazlar.

Cihaz, yatayda 57.5 derecelik ve dikeyde 43.5 derecelik fiziksel görüş alanı ile birlikte +/- 27 derecelik yukarı aşağı eğim özelliği bulunmaktadır. Şekil 3.2’de Kinect sensörünün algılama sınırları gösterilmektedir.



Şekil 3.2. Kinect fiziksel ve pratiksel hareket izleme sınırları [41]

Kinect'in 0.8 m ile 4 m bir görüşü olmasına rağmen pratikte en iyi sonuçları 1.2 m ile 3.5 m arasında görülmektedir. Aygıt tarafından ayırt edilebilen kullanıcı iskeleti tespit edilip izlendiği için 1.2m-3.5m optimal aralıktır, bu aralığın dışında güvenilir olmayan sonuçlar üretilebilir.

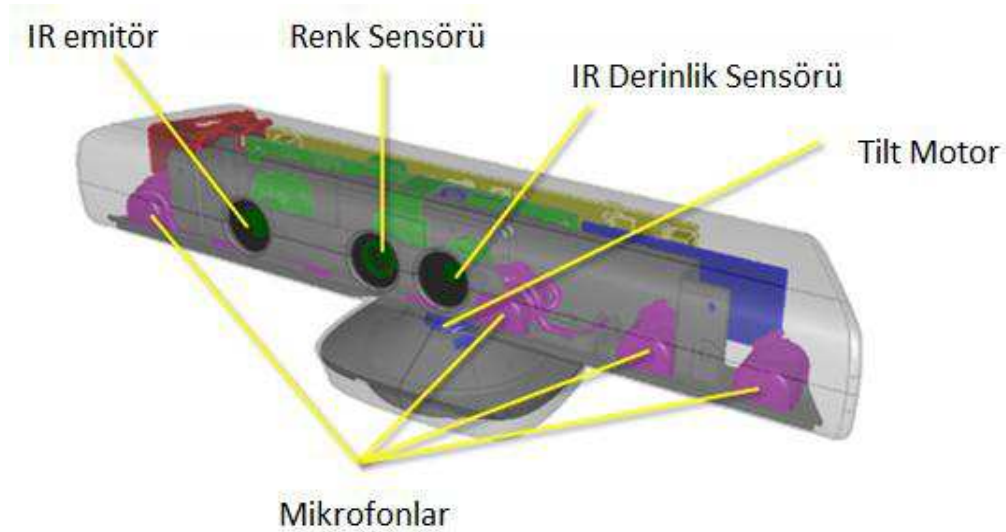
Kinect sensörü görüş alanı içerisinde bulunan 2 ayrı kullanıcının aktif hareketlerini izleyebilir. Bununla birlikte 6 kullanıcıya kadar algılayabilme özelliğine sahiptir. Kinect'in içerisinde bulunan mikroişlemcide insanın yapabileceği birçok hareketler tanımlıdır. Mikroişlemci içerisindeki yazılım, gerçekleştirilen durumu veya hareketi

tanımlar ise, buna karşılık gelen özel kodları bilgisayar sistemine yollar. Eğer insanın yaptığı bir hareket Kinect sistemi tarafından algılanmazsa veya tanımlı değilse, Kinect insanın son yaptığı harekette sabit kalır [40].

Kinect cihazının bir bilgisayarla entegre çalışabilmesi için USB (Universal Serial Bus) ile bilgisayara bağlanması ve ayrıca harici bir kaynaktan 5V güç alması gerekmektedir [40]. Cihaz görüntüleri resimler halinde ilettiği için uygulamalar içerisinde görüntülere çok rahat ulaşılır.

### 3.1.1.2. MS Kinect görüntü izleme ve iskelet takip sistemleri

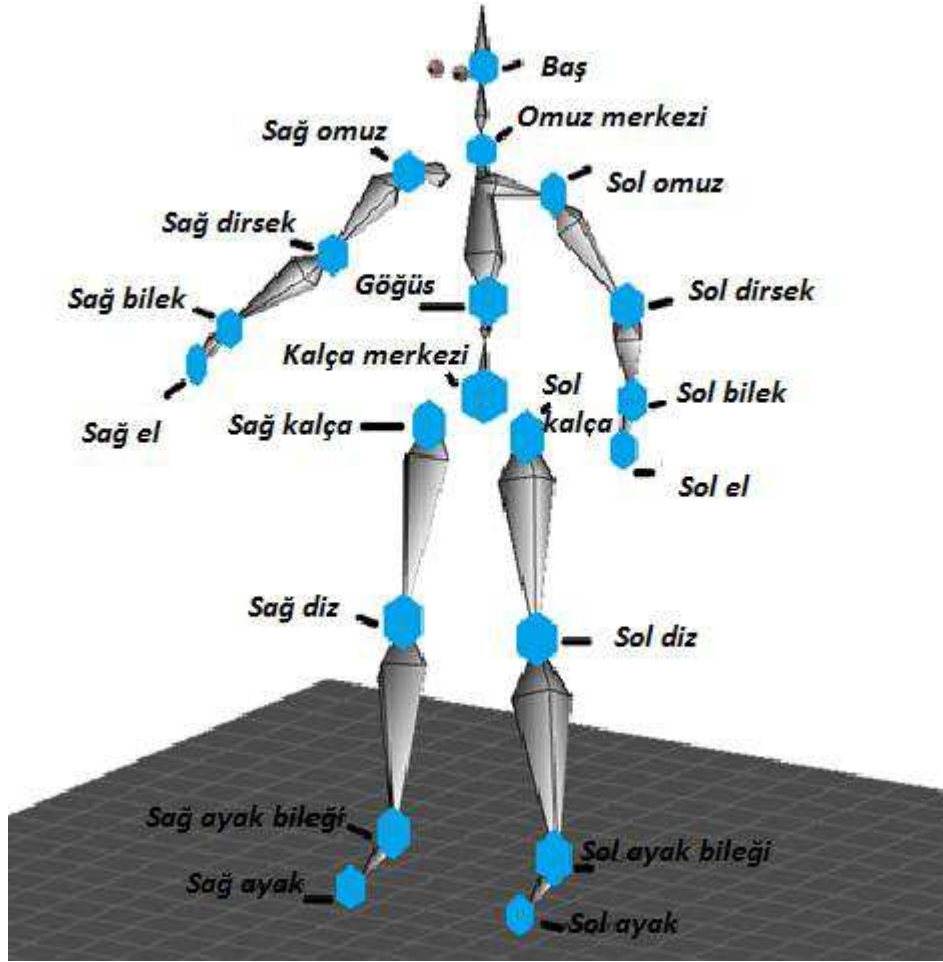
Kinect üzerinde bulunan soldaki bölme lazer projeksiyonu yaparken, sağdaki kızılötesi sensör bu ışınların gidiş - geliş süresini hesaplayarak her bir noktanın mesafesini bildirmektedir. Kinect içerisindeki yazılım ise bu veriler ışığında iskelet yapısını hesaplamakta ve bunu XBox'a yada bilgisayara göndermektedir[40].



Şekil 3.3. Kinect donanımsal bileşenleri [42]

Kinect'in ortasında bulunan göz ise 640 x 480 çözünürlüğünde 30 FPS (Frame Per Second) bir VGA (Video Graphics Array) kameradır. Yakalanan görüntü, saniyede 30 resim olarak uygulamaya iletilmektedir [40].

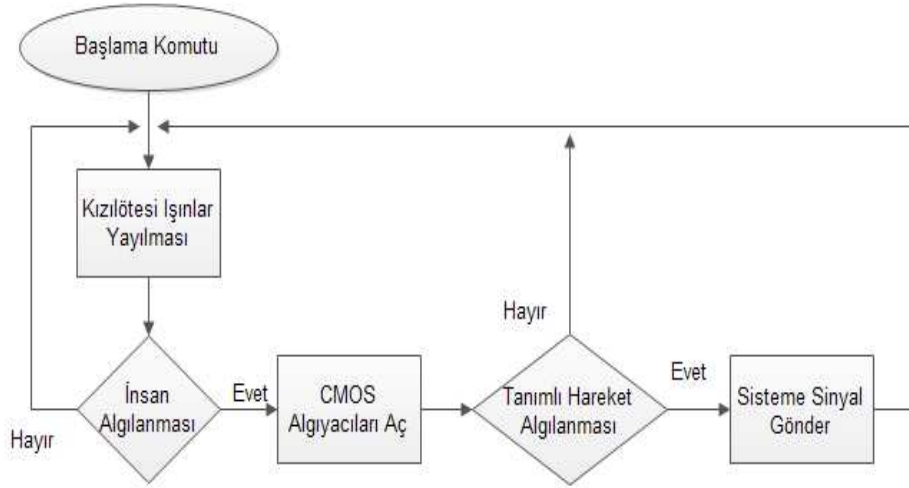
Kinect üzerinde bulunan kızılötesi kamera sayesinde insanın hareketli eklemleri algılanabilmekte ve izlenebilmektedir. Kinect kameraları aynı anda 2 farklı kişinin 20 farklı bölgesini aktif olarak algılayıp izleyebilmektedir. Tez içerisinde kullanılan model iskelet yapısı üzerinden bu bölgeler Şekil 3.4’de gösterilmiştir.



Şekil 3.4. İnsan iskelet yapısı üzerinde Kinect cihazının izlediği bölgeler

Kinect’e hareket algılama komutu verilmesi ile beraber IR (Infrared) kamera ortama kızılötesi ışın yapmaktadır. İnsanın el, kol, baş ve ayaklarının bulunabileceği noktalara yayılan kızılötesi ışınlar ile mevcut noktalarda insan algılanırsa CMOS (Complementary Metal Oxide Semiconductor) algılayıcılar açılarak hareket tanımlanmaya başlamaktadır. Kinect içerisindeki ROM’da kayıtlı olan hareket algılanırsa sistem bilgisayar sistemine uyarı kodu göndermektedir. Gerçekleştirilen

hareket, tanımlı hareketler arasında yok ise Kinect sistemi beklemeye almaktadır [40]. Şekil 3.5’de hareket algılama algoritması gösterilmiştir.



Şekil 3.5. Kinect hareket algılama algoritması

### 3.1.1.3. Kinect SDK ve diğer kütüphaneler

Kinect sensörü ile uygulama geliştirmek için Kinect SDK, OpenNI, OpenKinect gibi çeşitli kütüphaneler bulunmaktadır. Bunlar içerisinde en çok kullanılan iki tanesi Kinect SDK ve OpenNI kütüphaneleridir[43].

Kinect SDK: MS, programcılara “Kinect for Windows SDK” adı altında yayınlanan ticari olmayan bir yazılım geliştirme kiti (SDK) ile Kinect’in tüm özelliklerini kullanabilen oyun ve uygulamalar geliştirme olanağı sunmaktadır. Bu YGK ile piksellerin derinlik bilgisini veren derinlik sensöründen, piksel renk bilgisini veren renk kamerasından ve yönüyle birlikte çevreden algılanan ses bilgisini veren mikrofon setinden elde edilen ham verileri kullanıcıya sunar. Ayrıca cihaz içerisindeki algoritma ile elde edilen iskelet takibini de erişilebilir hale getirir. SDK, Natural User Interface API ve Audio API gibi programlama arayüzleri de (API) bulundurur. Natural User Interface API sensörlerin aktif edilmesi ve tilt(eğim) motorun kontrol edilmesi gibi genel birtakım fonksiyonları içerir. NUI API kendi içerisinde programcıya derinlik ve görüntü bilgisi içeren kameralara erişim sağlayan

NUI Image Camera API ve iskelet izleme fonksiyonları sunan NUI Skeleton API gibi alt arayüzler bulundurulur. Audio API ise programcıya mikrofon dizi bilgisine erişim sağlar [41]. Windows altyapısı ile çalışan programcılar için sıklıkla tercih edilir. C++, C# ve Visual Basic yazılım dilleri ile birlikte kullanılabilir [43].

Bu tez kapsamındaki hareket izleme yapısının daha net anlaşılması için Kinect çalışma mekanizması aşağıdaki gibi sıralanabilir [41]:

- 1- Kullanıcı Kinect cihazının önünde durur.
- 2- Derinlik verisi ile kullanıcının nokta kümesi oluşturulur.
- 3- Cihaz kullanıcı iskeleti için bir tahminde bulunur.
- 4- Kullanıcı vücut bölümleri farklı güvenilirlik seviyeleriyle tahmin edilir.
- 5- Kullanıcının duruşuna en yakın iskelet seçilir.

### **3.1.2. Sistemin geliştirilmesi ve test aşamasında kullanılan bilgisayarların teknik özellikleri**

Tez uygulaması 2 ayrı bilgisayar sistemi üzerinde gerçekleştirilmiş ve test edilmiştir. Bu sistemlere ait teknik özellikler aşağıdaki gibidir:

#### 1- ASUS N56VZ:

İşlemci	: Intel® Core™ i7 3610QM işlemci
İşletim Sistemi	: Windows 8
Bellek	:16 GB SDRAM, DDR3 1600 MHz SDRAM
Ekran	: 15.6" 4:3 Wide View Angle /Full HD (1920 x 1080)/HD (1366 x 768) LED Arkadan Aydınlatma
Grafik	: NVIDIA® GeForce® GT 650M bellekli 2GB DDR3 VRAM
Depolama	: 1TB

## 2- Lenovo Yoga 2 Pro

İşlemci	: Intel® Core™ i7-4500U İşlemci
İşletim Sistemi	: Windows 8.1 Pro
Bellek	: 8GB DDR3L 1600 MHz,
Ekran/Çözünürlük	: 13,3" yüksek çözünürlüklü QHD+ (3200 x 1800) 10 noktalı çoklu dokunmatik teknolojili IPS geniş görüntüleme açılı ekran
Grafik	: Entegre Intel® HD grafik 4400
Depolama	: 256GB SSD

### 3.2. Sistemin Geliştirilmesinde Kullanılan Yazılımlar ve Özellikleri

#### 3.2.1. MakeHuman

MakeHuman, grafiksel bir kullanıcı arayüzü ile sanal insan modeller oluşturmayı kolaylaştırmak için tasarlanmış bir yazılımdır. Grafiksel Kullanıcı Arayüzü genellikle GUI (Graphical User Interface) olarak geçmektedir. MakeHuman takımının bu geniş 3B modelleme dünyası üzerindeki bu özel konu üzerinde odaklanmasının sebebi, bu alanda mümkün olan en iyi kaliteyi yakalayabilmek ve kullanım kolaylığı sağlamaktır. Asıl ulaşılmak istenen hedef yalnızca birkaç mouse tıklamasıyla hızlı bir şekilde birçok gerçekçi sanal insanlar oluşturabilmek, oluşturulan bu modelleri render edebilmek veya diğer projelerde kullanılmak üzere istenilen formatta export edebilmektir [44]. MakeHuman 1.0 alpha 7 versiyonundan örnek bir arayüz Şekil 3.6'da gösterilmektedir.





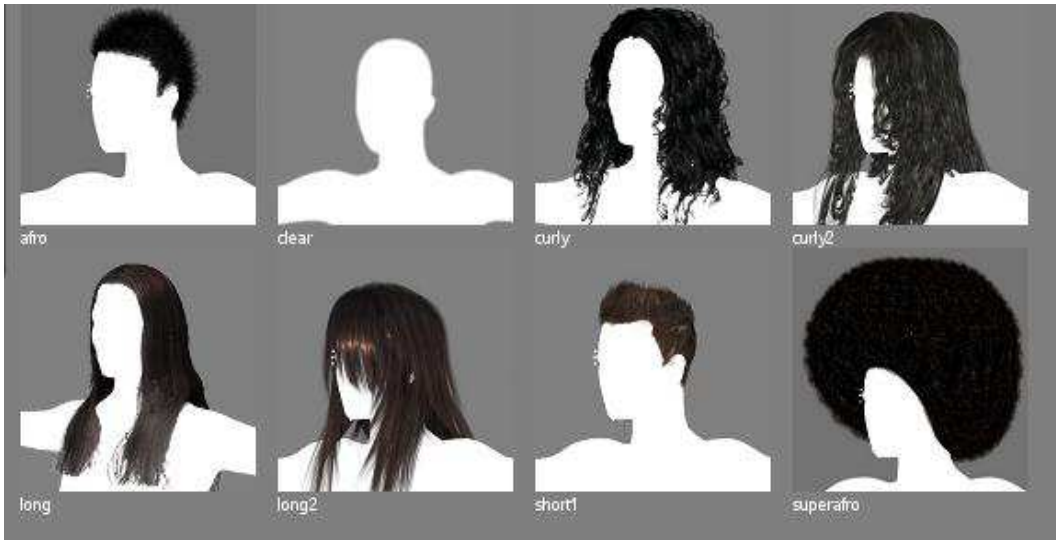
Şekil 3.6. MakeHuman 1.0 alpha 7 örnek kullanıcı arayüzü

Modeller, benzersiz 3B insan karakterler oluşturabilmek için farklı insan özelliklerini bir yerde toplayan kontrol manipülasyonlarıyla oluşturulurlar. Bu kontrol edilebilir nitelikler iki başlık altında toplanmıştır: makro ve detaylar. Makro kısmı genellikle cinsiyet, yaş, yükseklik, en ve etnik köken gibi genel insan karakteristiklerini hedef alan kısımdır. Detaylar kısmı ise gözün şekli veya parmak uzunluğu gibi daha düşük seviye detaylar üzerine odaklanarak daha farklılaştırılmış karakter oluşturulmasını sağlar [44].



Şekil 3.7. MakeHuman 1.0 alpha 7 model giysi ekranından bir görünüm

MakeHuman projesi, gerçekçi sanal insanlar oluşturmak için gereken her şeyin yönetimini sağlayan bir araç sağlamaya çalışmaktadır. Bu proje henüz oluşturulmamış olan veya poz, animasyon döngüleri, yüz ifadelerini yönetmek, saç ve elbiseler gibi daha gelişiminin ilk aşamalarında olan bazı araçları da içermektedir. MakeHuman kütüphaneleri ile kullanıcılar farklı pozlar, animasyon döngüleri, yüz ifadeleri, saç, ayakkabı ve elbiseleri görüntüleyebilmekte ve model üzerine aktarabilmektedirler. Yazılım ayrıca modellerin üzerlerinde daha ayrıntılı işlemlerin yapılabileceği diğer yazılımlara (Blender 3D gibi) aktarılmasını gerçekleştiren araçlar da sunmaktadır [44]. Bu aktarım özelliği ile tez için oluşturulan modeller Blender 3D programına aktarılmış ve üzerlerinde Bölüm 4’de gösterilen değişiklikler uygulanmıştır.



Şekil 3.8. MakeHuman 1.0 alpha 7 model saç tasarım ekranından bir görünüm

### 3.2.2. Blender modelleme yazılımı

Blender ilk olarak 1993 yılında geliştirilmeye başlanmış ve 1994 yılında kararlı sürümü çıkmış olan 2D ve 3D içerik oluşturulmasını sağlayan bir yazılımdır. Blender açık kaynak kodludur ve platformdan bağımsız olarak çalışmaktadır. Fiziksel benzetimlerin modellenmesini sağlayacak Bullet Physics isimli bir fizik motoruna sahiptir [45]. Blender’in güncel sürümü 2.71’dir. Geliştirici topluluğunun yanında,

oldukça geniş bir kullanıcı kitlesi de vardır. Blender ile yapılabilecek işlemler aşağıda sıralanmıştır [45]:

- Karakter modelleme (Character modelling)
- Kemik ekleme (Rigging)
- Katı modelleme (Solid modelling)
- Animasyon (Animation)
- Grafik gerçekleştirme (Graphic rendering)
- Işın izleme gerçekleştirme (Raytrace rendering)
- Fizik ve parçacık (Physics and particles)
- Gölgeleme (Shading)
- Resim ve kompozisyon (Image and composition)
- Kamera ve hareket izleme (camera and motion tracking)

Blender'in bu işlevleri yanında geniş eklenti yelpazesi sayesinde birçok işlem gerçekleştirilebilir. Blender 32 ve 64 bit sistemlerde çalışabilmektedir. İleri seviye kullanıcıları uygulamalarını özelleştirmek ve özel amaçlı araçlar geliştirmek için Python dilinde script tasarımını sağlayan Blender API'lerini kullanırlar. Blender ile platformlar arasında geçiş sağlanabilir ve Linux, Windows ve Macintosh sistemlerde oldukça performanslı çalışabilir. Arayüzü OpenGL kullanmaktadır [46].



Şekil 3.9. Blender arayüzü ve örnek bir tasarım [47]

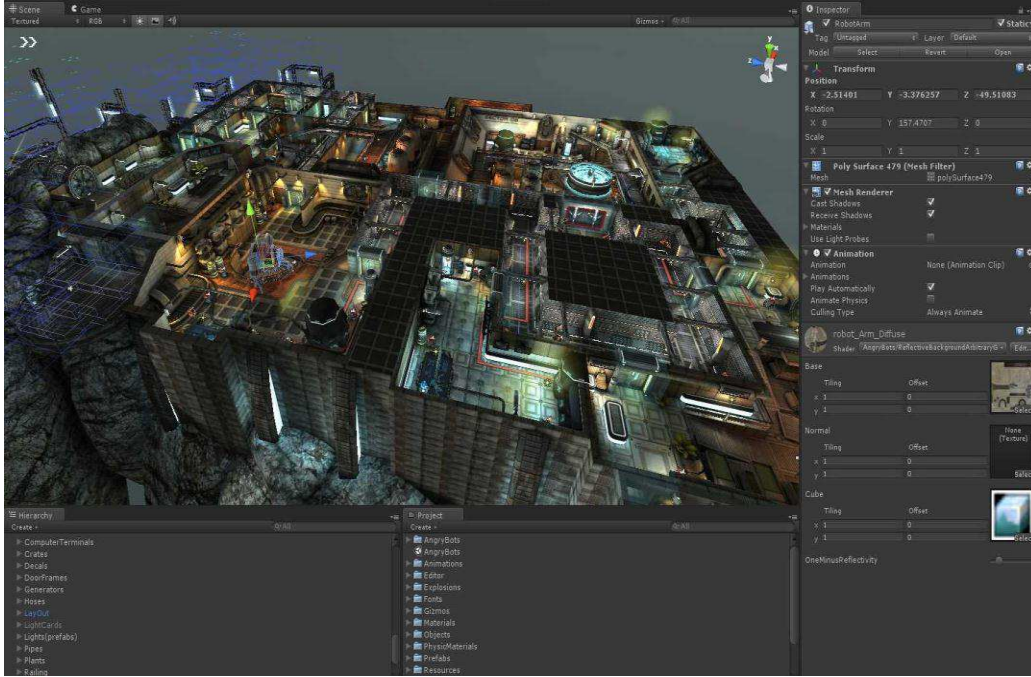
Fotoğereçkeçi render işlemleri sađlamak isteniyorsa bu tür render işlemleri birçok işlem gerektirdiđi için güçlü bir işlemci (CPU) ve RAM bulundurmanız tavsiye edilmektedir. Ancak çok gereçkeçi render uygulanmayacaksa veya daha çok kaba taslak çalıřmalar yapılacaksa çok güçlü sistemler bulundurmanız gerekmez. Blender Kuruluđu tarafından önerilen minimum gereksinimler [46]:

- 3 tuřlu Mouse
- 16 MB RAM Open GL Grafik kartı
- 300 MHz CPU
- 128 MB RAM
- 1024 x 768 piksel 16 bit ekran
- 20 MB disk alanı

### 3.2.3. Unity 3D

Unity 3D, bir oyun geliřtirme platform olup getirdiđi en önemli yeniliklerden birisi geliřmiř özelliklere sahip 3B oyunların bilgisayara kurulmadan oynanmasını sađlamak olmuřtur. Unity 3D motorunu kullanan oyunlar, Unity Web Player eklentisi sayesinde hiçbir kurulum işlemi olmadan web tarayıcı üzerinden çalıřabilmektedir. Bu özellik ile Unity ile geliřtirilen oyunlar düşük özelliklere sahip bilgisayarlarda rahatlıkla oynatılabilmektedir [48].

Unity'nin oyun yapımcılarına sađladıđı başka bir kolaylık da Unity ile geliřtirilen bir oyunun herhangi bir altyapı deđiřikliđine gerek olmadan farklı platformlara (PC, Mac, Web, iOS, Android, Windows Phone) uygun olarak derlenebilmesidir [48].



Şekil 3.10. Unity Asset Store üzerinden sağlanan AngryBots isimli demo projeden örnek sahne görünümü

Unity ile grafik ve kod birlikte çalışmaktadır. Bu çalışma mantığı geliştiriciye esneklik sağlamakta, geliştirme süresini kısaltmaktadır. Unity 3D şu anda 3 farklı skript dili yapısını (Java Script, C# ve Boo) desteklemektedir.

### 3.2.4. C# Programlama Dili

C#, C/C++ ve Java dillerinden türeyen, güçlü, basit, esnek, tip güvenli, modern ve Microsoft.NET platformu için sıfırdan geliştirilmiş tek programlama dilidir. Bilgisayarlar ilk çıktıkları günden bu yana değişik düzeyde birçok programlama dili kullanılarak programlanıp insanlara faydalı araçlar haline getirilmiştir.

Örnek olarak Delphi yüksek düzeyli diller arasındayken, C düşük düzeyli diller arasındadır. C# orta düzeyli diller grubuna girer. C# ile hem alt düzey hem de üst düzey programlar yazılabilir [49].

Java dilinin piyasaya sürülmesi ve C/C++ gibi dillerin nesne yöneliminde zayıf kalması ile Microsoft atağa geçmiş ve .NET ile uyumlu en gözde dillerden biri olan C# ortaya çıkmıştır [49].

C# programlama dilinin diğer programlama dillerinden ayıran bazı özellikler aşağıdaki gibi sıralanabilir [40].

- Öğrenilmesi kolaydır.
- Yüksek verim. Program yazarken oluşan hataların önüne geçilmiştir.
- Nesne yönelimli programlamayı destekler. Bu sayede büyük ölçekli projeler hızlı bir şekilde geliştirilebilir.
- Güç ve kolaylık arasında denge. Hızlı ve güçlü programlar geliştirmek için uygundur.
- XML desteği.
- Yönetilmeyen kod. C# ile yazılım geliştirirken gerekli olmasa da işaretçiler kullanılır. Bu tür program kodlarına yönetilmeyen kodlar denir.
- Modern bir dil. İnternet çağının gerektirdiği tüm özellikleri desteklemesinin yanında hızlı yazılım geliştirmeye olanak tanır [50].

C# programlama dili bilgisayar dünyasında her platformda çalışacak yazılımlar geliştirilmek için kullanılabilir. C# ile yapılabilecek bazı işler şunlardır [50].

- C#, Windows uygulaması gerçekleştirme, Asp.Net uygulaması geliştirme.
- Windows için gelişmiş, güçlü, hızlı ve güvenli program yazma.
- Web form uygulamaları geliştirme.
- Web üzerinden servis veren program parçacıkları yazma.
- Mobil uygulamalar geliştirme.
- .Net desteği olan diller için güçlü, hızlı ve esnek DLL (Dynamic Link Library) bileşenleri yazma.
- WPF uygulamaları geliştirme.
- Dokunmatik donanım kontrolü için kolay kütüphane barındırma ve yazma

## **BÖLÜM 4. 3B SANAL İNSAN MODELLEME ve GERÇEK ZAMANLI KONTROLÜ**

Tez uygulaması kapsamında bir erkek ve bir kadın olmak üzere 3B iki adet model oluşturulmuştur. Oluşturulan bu modellerde MakeHuman programının sanal insan modelleme üzerine sağladığı kolaylık ve özelliklerden faydalanılmıştır. MakeHuman ile oluşturulan erkek ve kadın modeller Blender 3D programı ile üzerlerinde mesh ve doku düzenlemeleri yapılarak Unity 3D oyun motoruna aktarılmıştır. Ayrıca kullanıcılar için sanal bir giyinme kabini arayüzü tasarlanmıştır. Uygulamada kişi iskeletini takip etmek büyük önem taşımaktadır. Uygulamada kişi kullanılacak ekranın önüne geçer, Kinect cihazı üzerinden program içerisinde geliştirilen algoritma ile kişinin beden ve uzunluk ölçüsü alınır. Alınan ölçümlere göre kişi beden ölçüleri otomatik olarak ekranda yansır ve oluşturulan 3B model bu ölçülere göre ekran üzerinde boyutlandırılır.

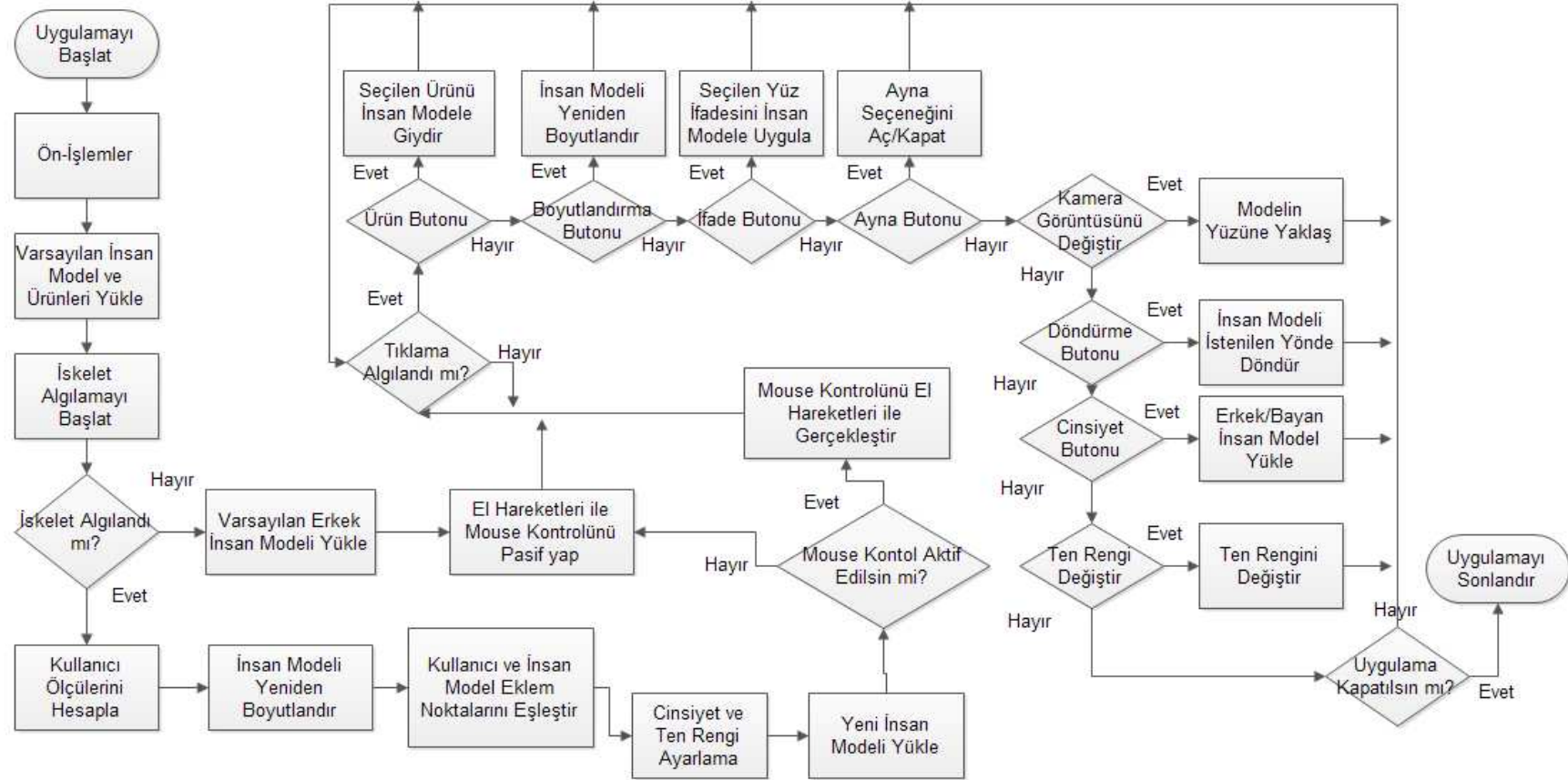
Kinect SDK'nın getirdiği özelliklerden faydalanılarak kişi eklem noktaları ile sanal modelin eklemeleri ilişkilendirilerek modelin kullanıcı ile eş zamanlı olarak aynı hareketleri yapması sağlanır. Ayrıca kullanıcı sağ eliyle mouse kontrolünü de isteğe bağlı olarak gerçekleştirebilmektedir. Kinect sensör ile kullanıcı algılandığı zaman model ile kullanıcı otomatik olarak ilişkilendirilir ve kullanıcı el hareketleriyle sanal kabin üzerinden kıyafet seçimi yapabilir. Seçilen kıyafetler belirlenen beden ölçülerine göre model üzerine giydirilir. Model ve giysi uyumunun rahat bir şekilde izlenmesi açısından arayüz üzerinde döndürme butonları bulunmaktadır. Kullanıcı bu butonlar ile modeli ekranda döndürebilir ve farklı açılardan görüntü sağlayabilir. Modellerin en önemli kısımlarından biri yüz olduğu için gülümseyen, üzgün, kızgın vb. bazı yüz ifadeleri de eklenmiştir. Ortamda Kinect sensörü algılanmadığı durumda uygulama mouse vb. bir giriş donanımı ile manuel olarak da kullanılabilir.

Geliştirilen arayüzde en önemli amaçlardan bir tanesi online alışverişlere katkı sağlamaktır. Sanal kabin çalışması ile birlikte kullanıcılar kendi beden ölçülerindeki

modeller üzerinde giysileri 3B olarak deneme imkanı bulacak bununla beraber herhangi bir fiziksel uğraş veya kabin sırası beklemek gibi zaman kaybı yaşamayacaklardır.

Uygulamanın geliştirme aşamaları 3 alt başlık halinde anlatılmaktadır. Birinci aşamada MakeHuman üzerinde yapılan işlemler, ikinci aşamada Blender 3D üzerinde gerçekleştirilen düzenlemeler, üçüncü aşamada ise Unity 3D oyun motoru ile yapılan tasarımlar ve Unity ile birlikte yüklenen MonoDeveloper eklentisi üzerinde yazılan script kodların işlevleri anlatılacaktır. Uygulamaya ait akış diyagramı Şekil 4.1’de sunulmaktadır.



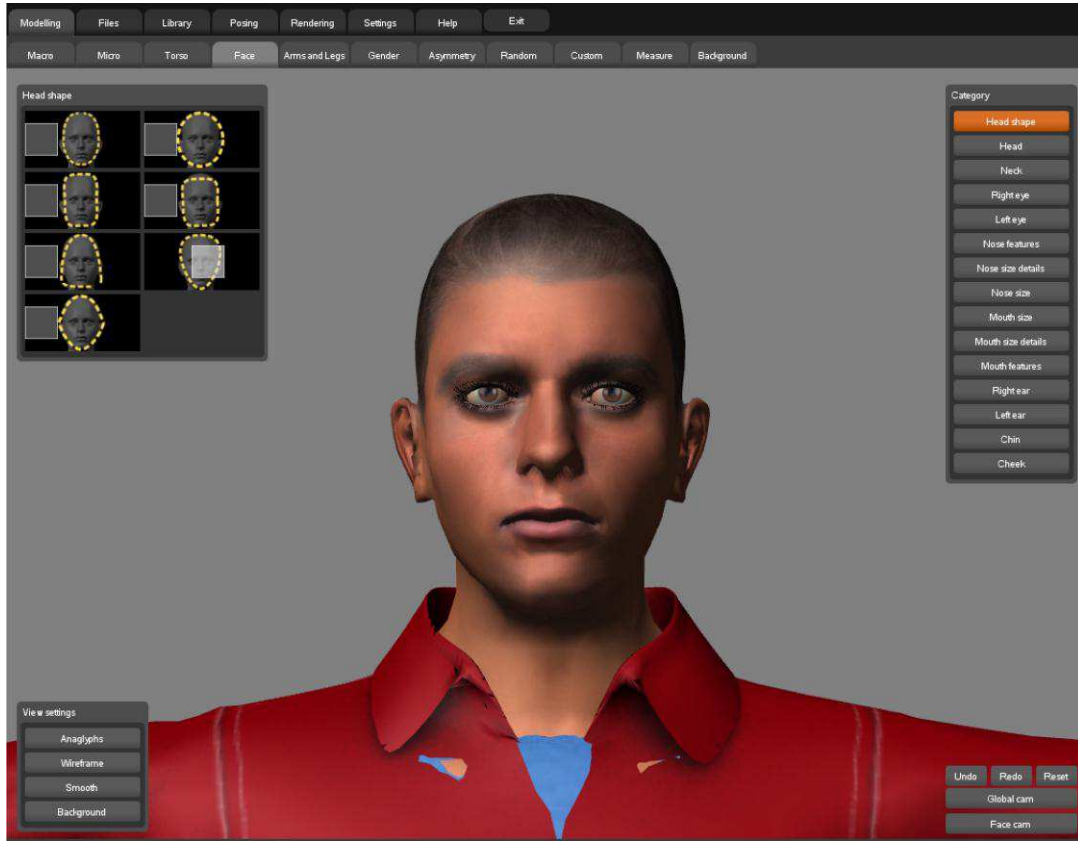


Şekil 4.1. Uygulamaya ait Akış Diyagramı

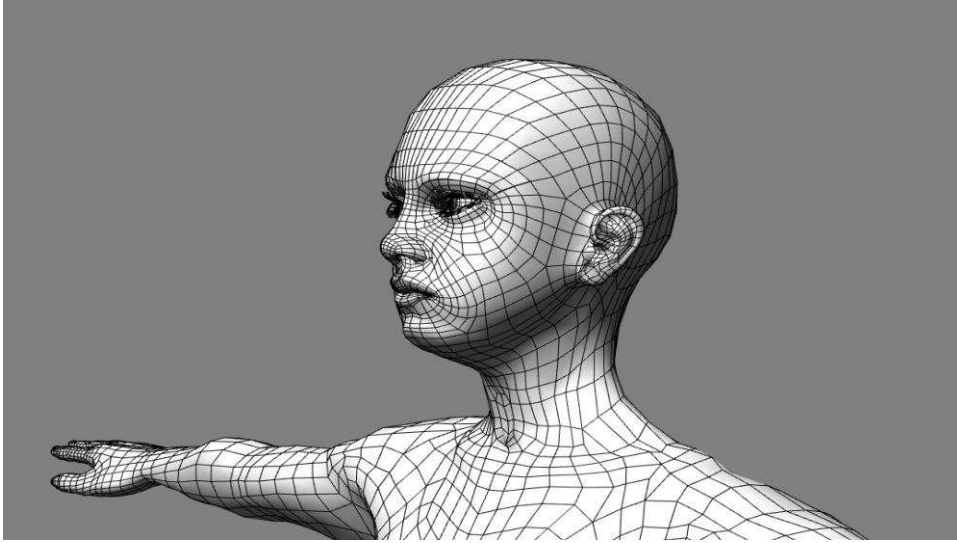
## 4.1. MakeHuman Yazılımı ile Gerçekleştirilen İşlemler

### 4.1.1. 3B sanal insan modellerin oluşturulması

3B modeller başlangıçta varsayılan beden ve uzunluk ölçüleriyle MakeHuman açık kaynak kodlu yazılımı ile oluşturulmuştur. Modellerin kol, bacak, gövde ve yüz şekil ve ölçüleri yine aynı yazılım üzerinden belirlenmiştir. Modellere kullanıcının kendi yüz şekil ve ölçülerinin aktarılması bu çalışmanın ileriki aşamalarını kapsamaktadır. İskelet yapısı uygulama için gerekli karmaşıklık düzeyinde sunulan alternatifler arasından seçilmiştir. Şekil 4.2 ve 4.3, modellerin oluşum aşamalarından görüntüler içermektedir.



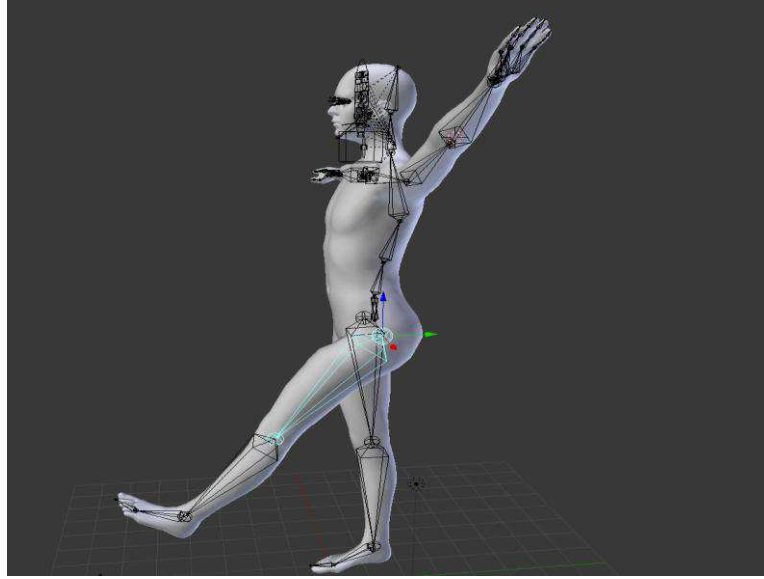
Şekil 4.2. Arayüz üzerinden erkek modelin yüz şeklinin ayarlanması



Şekil 4.3. Uygulama içerisindeki kadın modellerden birine ait mesh yapısı

#### **4.1.2. Modellerin rig yapısının belirlenmesi ve blender exchange (.mfx) dosyalarına dönüştürülmesi**

MakeHuman çeşitli uzantılarda modellerimizi kaydetmemize olanak sunar. Bunlardan en çok kullanılanlar ise .obj ve .mfx uzantılı olanlarıdır. Bu uzantılar ile modeller 3B modelleme yazılımlarına aktarılarak üzerlerinde daha detaylı işlemler ve farklı uygulamalar gerçekleştirilebilir. Bu uzantılardan .mfx, Blender içerisinde ilgili kütüphane aktif edildiği zaman Blender 3D arayüzüne aktarılabilmekte ve üzerinde değişiklikler yapılabilmektedir. Şekil 4.4, model iskelet (rig) yapısından bir görünüm sunmaktadır.



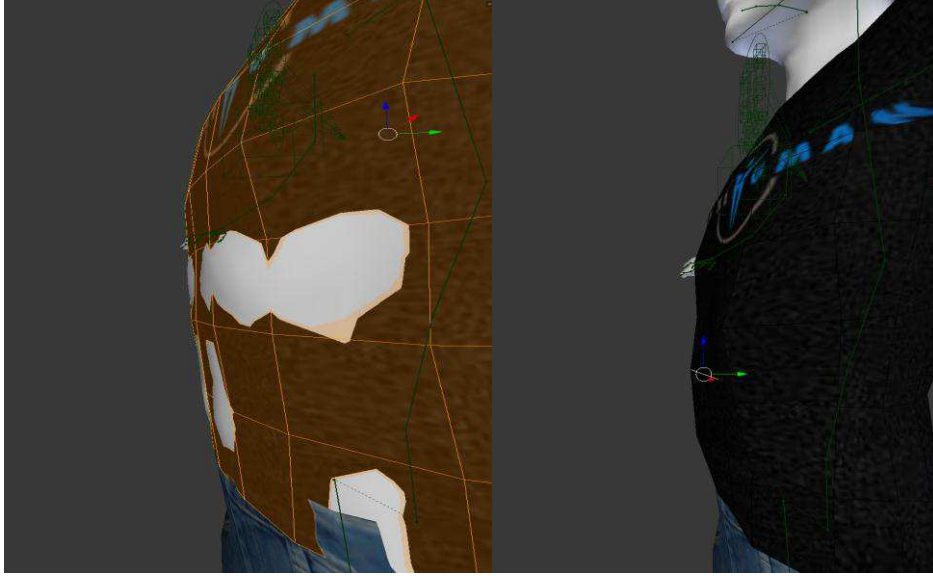
Şekil 4.4. Örnek model iskelet yapısı

## 4.2. Blender 3D ile Modellerin Düzenlenmesi

Bu aşamada Blender 3D üzerinde gerçekleştirilen işlemler anlatılacaktır.

### 4.2.1. Mesh uyumsuzluklarının giderilmesi

MakeHuman açık kaynak kodlu ve sürekli geliştirilmekte olan bir yazılımdır. Bu sebeple bazı durumlarda elbiselerin model üzerinde tam oturmaması, saç şekillerindeki uyumsuzluk gibi birtakım sorunlar oluşabilmektedir. Kullanıcılar burada oluşturdukları modelleri farklı bir 3B modelleme yazılımına aktararak üzerlerinde daha detaylı düzenlemeler yapabilmektedirler. Şekil 4.5’de bazı uyumsuzluklar ve düzeltilmeleri örneklenmiştir.

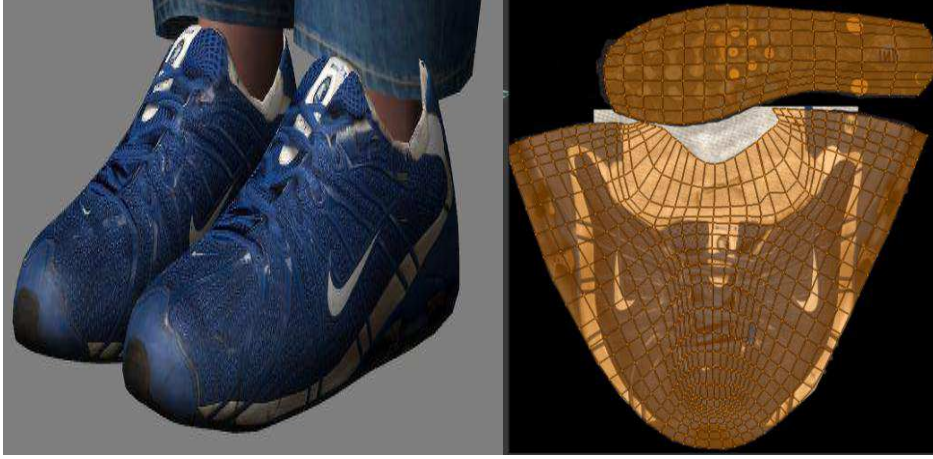


Şekil 4.5. Giysi uyumsuzluklarının giderilmesi

#### 4.2.2. UV Mapping özelliği ve giysi dokularının ayarlanması

“UV Mapping” özelliği, 3B bir nesneye 2B bir dokunun haritalanmasının en esnek yöntemlerinden biridir. Bu işlem 3B tasarımın (X, Y ve Z) alınarak 2 boyutlu (X ve Y, veya isimde gösterildiği gibi U ve V) düz görüntüler halinde açılmasıdır. Görüntü üzerindeki renk ve desenler mesh üzerine haritalanmış olan renk ve desenlerdir ve mesh yüzlerine denk gelen renk ve desenleri gösterir. Prosedürel materyal ve dokuların sağlayamayacağı kalitede gerçeklik ve Vertex Boyama özelliğinden daha detaylı nesnelere oluşturmak için bu özellik kullanılabilir.

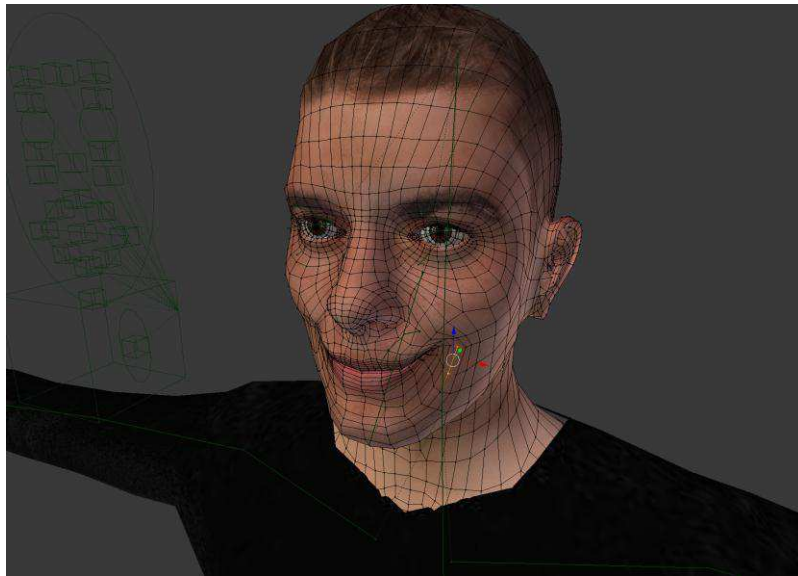
Bu özelliğe haritalama denmesinin sebebi nesneyi harita gibi 2 boyutun yani düzlemin üzerine açarak doğru bir şekilde düzenlemektir. Böylece istenilen doku mevcut nesne üzerine doğru şekilde oturtulur. Şekil 4.6’da ayakkabı nesnesinin UV mapping özelliği ile dokularının ayarlanması gösterilmektedir. Tez içerisinde MakeHuman ile blender programına aktarılan model ve giysilerin doku ayarlanması bu özellik ile sağlanmıştır.



Şekil 4.6. UV Mapping ile ayakkabı nesnesinin doku düzenlemesi

### 4.2.3. Yüz ifadeleri oluşturma

Blender 3D, basit bir mesh üzerinde shape keys (şekil anahtarları) özelliğini kullanarak çeşitli farklı şekiller oluşturmanızı ve animasyonlar oluşturabilmenizi sağlar. Shape key özelliğinin birçok kullanım amacı vardır. Ama özellikle karakter animasyonlarında ve bu tez içeriğinde de olduğu gibi yüz ifadeleri oluşturmak için kullanılmaktadır. Nesnenin Basis olarak adlandırılan genel bir şekli üzerinden başlanıp istenilen sayıda farklı şekiller oluşturulabilir. Şekil 4.7’de Blender edit mod üzerindeyken “ShapeKey” özelliği kullanılarak gülümseme yüz ifadesi eklenmesi gösterilmektedir.



Şekil 4.7. Modelin yüzünde değişik ifadeler oluşturulması

### **4.3. Unity 3D Oyun Motoru ile Arayüz ve Script Tasarımı**

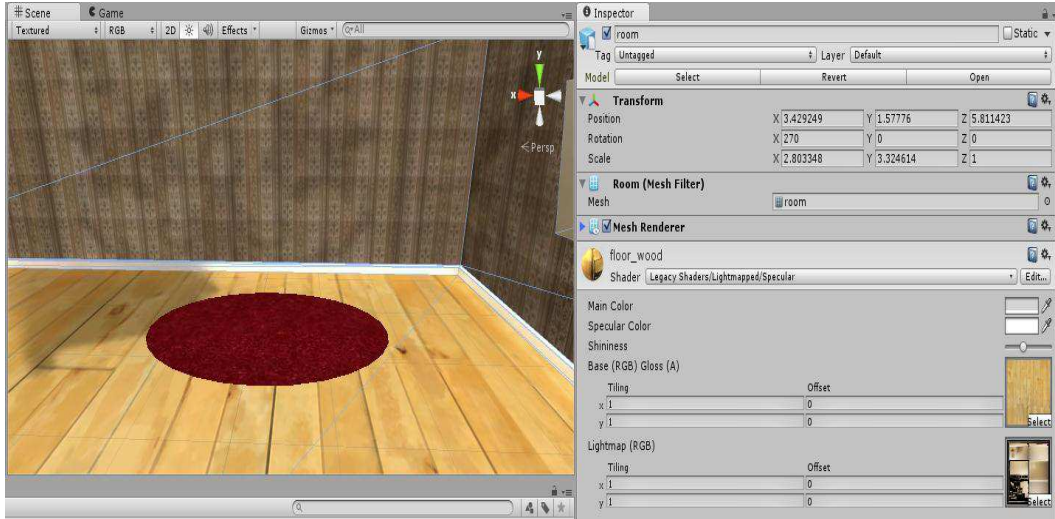
#### **4.3.1. Modelleme tabanlı işlemler**

Unity 3D oyun motoru çeşitli uzantılı dosyaları projelerinize aktarabilmenize olanak sunar. Bu uzantılardan bir tanesi de .fbx uzantılı olan dosyalardır. Bu dosya türü Kaydara tarafından geliştirilen şimdi ise Autodesk firmasının üstlendiği özel bir dosya biçimidir. Dijital içerik oluşturma uygulamaları arasında birlikte çalışabilirlik özelliği sağlar [51].

Blender 3D üzerinde model, giysiler, dokular, shapekey, animasyon vb. tüm özellikler düzenlendikten sonra bu işlemler çeşitli uzantılarda kaydedilebilir. Tez kapsamında modeller ve bileşenlerini içeren dosyalar, Unity motorunun da desteklediği .fbx formatında kaydedilerek Unity içerisine içerisine aktarılmıştır.

##### **4.3.1.1. Sanal kabin bileşenlerinin oluşturulması**

Unity arayüzü ile kullanıcılara 3B sanal bir kabin oluşturulmuştur. Bu kabin arayüzü içerisinde nesnelere yerleştirileceği bir oda, sol tarafta kullanıcıların yüz ifadesi bölmesi, orta kısımda modelin üzerinde duracağı bir halı ve tercih edildiği takdirde görünür olabilen bir ayna düzeneği, sağ tarafta ise kullanıcının giysilerini seçebileceği bir reyon tasarlanmıştır. Sanal değiştirme odası arayüzü tasarım aşamasından örnek bir görünüm Şekil 4.8’de sunulmaktadır.



Şekil 4.8. Sanal kabin tasarım aşamasından bir görünüm

Şekil 4.9 ise uygulama için oluşturulan genel oyun nesnelerinin yer aldığı hiyerarşi sekmesini göstermektedir.



Şekil 4.9. Oyun nesnelerinin (game objects) yer aldığı Hiyerarşi sekmesi

### 4.3.2. Script tabanlı işlemler

Unity, 3 farklı script dili desteklemektedir: C#, Javascript, Boo. Tez kapsamında scriptler C# dili ile gerçekleştirilmiştir. Unity3D obje, parçacık ve parçacığa bağlantılı kod mantığıyla çalışmaktadır. Bir oyun taslağı hazırlamak için obje , objeyi canlandırmak için parçacık , canlandırılan objeye görevler vermek için de parçacığa bağlı script kodu kullanılır.

UnityScript’de iki temel fonksiyon vardır. Bu fonksiyonlar ;



### 1.UPDATE fonksiyonu

```
function Update() {  
    //program çalışırken gerçekleşen olaylar  
}
```

Update fonksiyonu içerisinde gerçekleşen olaylar program kapatılıncaya kadar çalışır durumda olur.

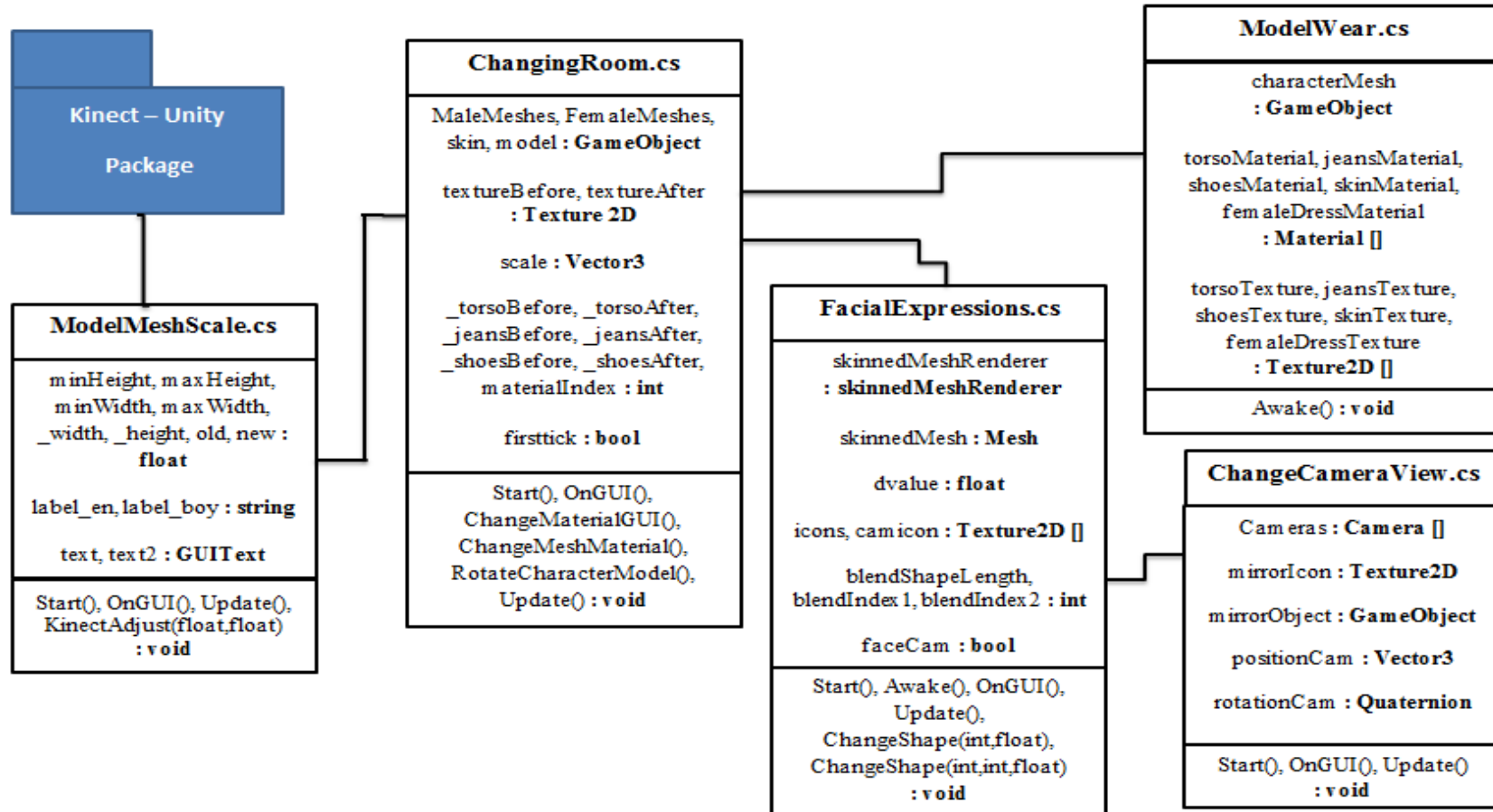
### 2.START fonksiyonu

```
function Start() {  
    //program başlatılırken gerçekleşen olaylar  
}
```

Start fonksiyonu içerisinde gerçekleşen olaylar sadece program açıldığı anda gerçekleşir. Daha sonra yerini Update olaylarına bırakır.

Uygulamaya ait Sınıf Diyagramı Şekil 4.10'da gösterilmiştir.

# UYGULAMA SINIF DİYAGRAMI



Şekil 4.10. Uygulamaya ait Sınıf Diyagramı

#### 4.3.2.1. Kinect cihazı algılanması

Uygulama başlatıldığında ilk olarak sistemde Kinect cihazının bulunması ve kullanıma hazır durumda oluşu kontrol edilmektedir. Eğer ortamda Kinect cihazı tespit edilemezse uygulama manuel olarak devam etmekte, model ve giysi işlemleri mouse ile yapılabilmektedir. Aşağıda, Şekil 4.11’de yer alan kod bloğu Kinect cihazının yüklenmesi kontrolünü gerçekleştirmektedir.

```

int hr = 0;

    try
    {
        hr = KinectWrapper.NuiInitialize(KinectWrapper.NuiInitializeFlags.UsesSkeleton |
        KinectWrapper.NuiInitializeFlags.UsesDepthAndPlayerIndex |
        (ComputeColorMap ? KinectWrapper.NuiInitializeFlags.UsesColor : 0));

            if (hr != 0)
            {
                throw new Exception("NuiInitialize Failed");
            }
    }

```

Şekil 4.11. Kinect kontrolünü sağlayan kod bloğundan bir görünüm

#### 4.3.2.2. Matematiksel formüller ve kod içerisinde kullanımı

Uygulama içerisinde geliştirilen matematiksel formüller ve kullanım yerleri aşağıdaki şekilde özetlenebilir:

- 1- *Kullanıcı eklem noktalarından boyut verilerinin elde edilmesi* : Kinect ile elde edilen kullanıcı eklem noktalarına ait verilerden uygulama içerisinde geliştirilen algoritma ile kullanıcıya ait yeni en ve boy değerleri hesaplanmıştır. Yeni değerler birtakım filtreleme işlemlerinde ve kullanıcının beden ölçülerine sahip insan modeller oluşturulmasında kullanılmaktadır.

$$W(U_1) = J(k_1)_x - J(k_2)_x$$

$$H(U_1) = J(k_1)_y - J(k_2)_y$$

```
Vector3 _width = GetPositionBetweenIndices(ref jointsPos,
NuiSkeletonPositionIndex.ShoulderLeft,NuiSkeletonPositionIndex.ShoulderRight);

Vector3 _height = GetPositionBetweenIndices(ref
jointsPos,NuiSkeletonPositionIndex.ShoulderCenter,NuiSkeletonPositionIndex.HipCenter);
```

Şekil 4.12. Parametrik olarak girilen eksen noktalarının eksenlere göre farkının Vector3 değişkenlerinde tutulması

```
private static Vector3 GetPositionBetweenIndices(ref Vector3[] jointsPos,
NuiSkeletonPositionIndex p1, NuiSkeletonPositionIndex p2)
{
    Vector3 pVec1 = jointsPos[(int)p1];
    Vector3 pVec2 = jointsPos[(int)p2];
    return pVec2 - pVec1;
}
```

Şekil 4.13. Parametrik olarak alınan iki eklem noktasının eksenlere göre farkını döndüren fonksiyon bloğu

2- *Yumuşatma Filtresi* : Kullanıcıya ait Kinect üzerinden elde edilen ölçü değerleri, birtakım filtreleme işlemlerine tutularak bu değerler arasındaki ani geçişlerin engellenmesi amaçlanmıştır.

$$y(t) = a_k * y(t) + (1 - a_k) * y(t - 1), \quad t > 0$$

```

Void KinectAdjust (float w, float h)
{
    Float mult = 0.005f;

    w = (mult * w) + ( (1-mult) * old_w);
    h = (mult * h) + ( (1- mult) * old_h);

    old_w = w;
    old_h = h;

    text.text = h.ToString();      text2.text = w.ToString
}

```

Şekil 4.14. Yumuşatma Filtresinin uygulandığı fonksiyon

3- *Model Döndürme* : İnsan model üzerinde farklı açılardan görüntü elde edilmesi amacıyla insan model ve giysilerin farklı eksenlere göre döndürülebilir olması sağlanmıştır.

y eksenine göre döndürme :

$$y(r)_{x,y,z} = (0, \pm 1, 0) * \Delta_t * 100$$

```

if(GUI.RepeatButton(new Rect(Screen.width/2 -200, Screen.height/2 + 100,70,70),"<"))
    model.transform.Rotate(Vector3.up*Time.deltaTime * 100 );

if(GUI.RepeatButton(new Rect(Screen.width/2 + 200,Screen.height/2+100,70,70),">"))
    model.transform.Rotate(Vector3.down*Time.deltaTime * 100 );

```

Şekil 4.15. İnsan modelin arayüz üzerinde y eksenine göre döndürülmesini sağlayan kod bloğu

4- *Ana Kamera Pozisyon ve Rotasyonu* : Yüz ifadelerinin vb. daha net görülebilir olması amacıyla ana kameranın farklı durumlarda görüntü açılarının ve konumunun değiştirilmesi sağlanmıştır.

$$MC(p)_{x,y,z} = A_{c1,c2,c3} * (\pm 1, \pm 1, \pm 1)$$

$$MC(r)_{x,y,z,w} = (0,180,0,1)$$

```

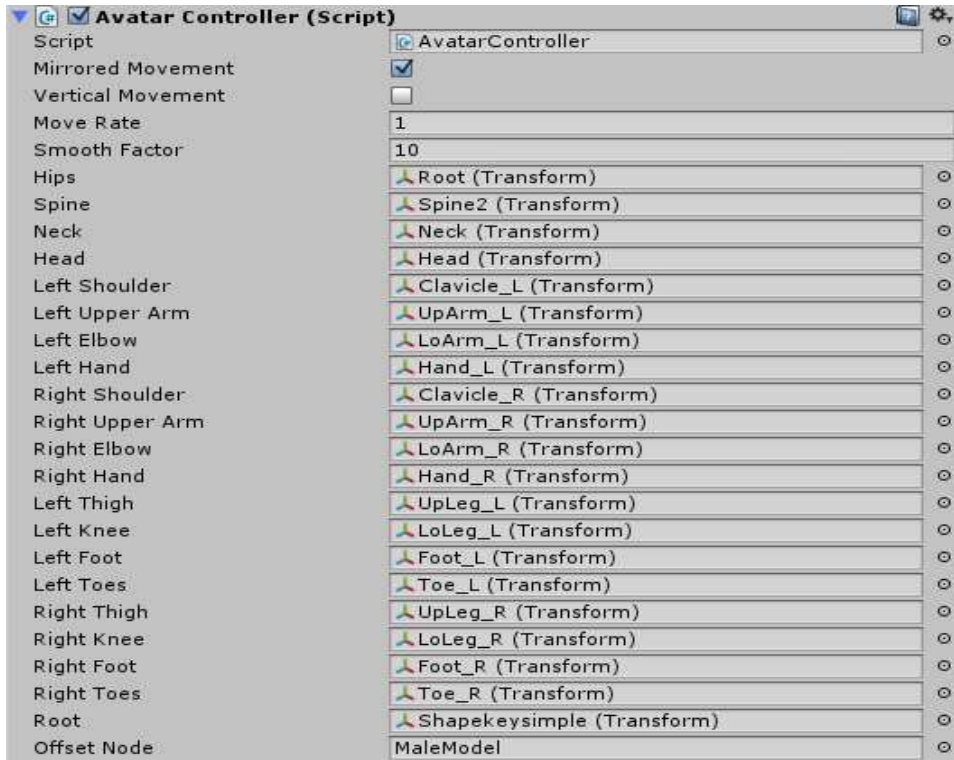
if (FacialExpressions.faceCam == true)
{
    cameras[0].camera.transform.localPosition = Vector3.Scale(new Vector3
(c1,c2,c3), Vector3.one);
    cameras[0].camera.transform.localRotation = new Quaternion(0f,180f,0f,1f);
}
else
{
    cameras[0].camera.transform.localPosition = positionCam;
    cameras[0].camera.transform.localRotation = rotationCam;
}

```

Şekil 4.16. Yüz ifadesinin daha net görüntülenmesini sağlayan kamera konum ve rotasyon değişikliği komutları

#### 4.3.2.3. Kullanıcı ve model entegrasyonu

Cihaz kontrolü gerçekleştirmelerinin ardından Kinect cihazı kullanıma uygun durumda ise model ile kullanıcının eklemleri ilişkilendirilir. Model eklemleri model nesnesinin içerisinden sürükleyip bırak yöntemi ile veya Şekil 4.17’de gösterilen script ekranından seçilerek ilgili kodun içerisinde tanımlı dizinin uygun indisli elemanlarına atanırlar. Şekil 4.17 ve 4.18 bu eklem noktalarının tanımlı dizi elemanlarına atanması ve script içerisinde bu dizilerin tanımlandığı kodları göstermektedir.



Şekil 4.17. Kullanıcı ve model eklemelerinin eşleştirilmesi ve model eklemelerinin dizi elemanlarına atanması

```

void MapBones()
{
    if(Hips != null)
        bones[1] = Hips;

    if(Spine != null)
        bones[2] = Spine;

    if(Neck != null)
        bones[3] = Neck;

    if(Head != null)
        bones[4] = Head;

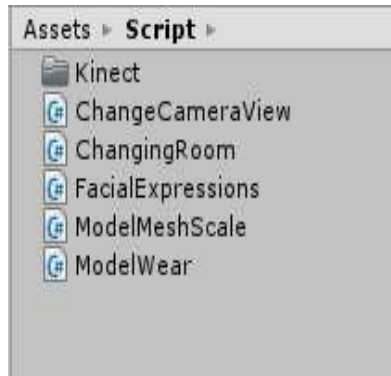
    if(LeftShoulder != null)
        bones[5] = LeftShoulder;
}

```

Şekil 4.18. Arayüz üzerinde ilişkilendirilen model eklemelerini diziye atayan kod bloğu parçası

#### 4.3.2.4. Diğer bazı oluşturulan scriptler ve özellikleri

Uygulama içerisinde birçok nesne ve olaylar bulunduğu için tanımlı birçok script ve uzun kod blokları barındırmaktadır. Scriptler geliştirilirken nesneye dayalı programlamanın özelliklerinden, sınıf ve fonksiyon tanımlamalarından faydalanılmıştır. Şekil 4.19 script sekmesinden bir görünüm sunmaktadır.

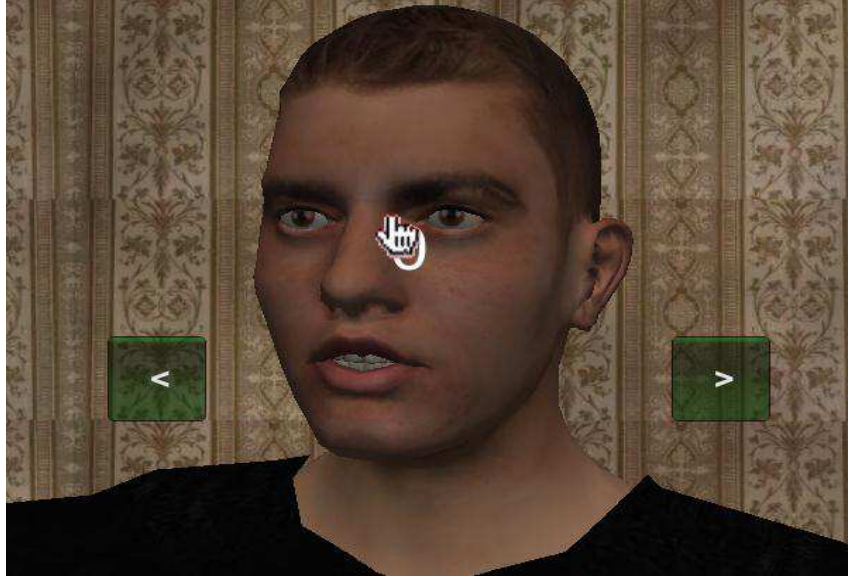


Şekil 4.19. Uygulama scriptlerinin yer aldığı Unity 3D script sekmesi

Bu scriptler içerisinde en temel olanları ve işlevleri aşağıdaki şekilde gösterilebilir:

ChangeCameraView.cs: Herhangi bir yüz ifadesi seçildiği zaman bu script içerisindeki kodlar yardımı ile kamera model yüzüne yaklaşmakta ve yüz ifadeleri ve mimikler daha net şekilde görüntülenebilmektedir. Aşağıda bu işlemi sağlayan kod boğunun gerçekleştirdiği olay örneklendirilmiştir.





Şekil 4.20. Kameranın yüze yaklaşması ve ifadenin daha net görüntülenmesi

ChangingRoom.cs: Uygulma ile ilgili en temel fonksiyonların bulunduğu script dosyasıdır. Tercih edilen erkek veya bayan modelin ekranda yüklenmesi, model döndürme butonlarının tasarımı, giysi kabin buton tasarımı ve seçilen giysilerin model üzerinde görüntülenmesi gibi işlemler bu script içerisinde yazılan kodlar ile gerçekleştirilmektedir.

```

void OnGUI()...

private void ChangeMaterialGUI()...

private void ChangeMeshMaterial()...

private void RotateCharacterModel()...

private void ChangeCharacterMesh()...

void Update()...

private void InstantiateCharacterModel() {

switch(_charModelIndex) {

    case 1: _charModelName = "Female Model";

        break;

    default : _charModelIndex=0;

        _charModelName="Male Model";

        break;

}
}

```

Şekil 4.21. Temel fonksiyonların yer aldığı ChangingRoom.cs adlı script içerisinden bir görünüm

FacialExpressions.cs: Yüz ifade butonlarına tıklandığından model üzerinde tanımlı ifadeleri düzenleyen script kodlarını bulundurur. Aşağıdaki kod bloğu parametrik olarak belirtilen ağırlık oranında modelin yüzü üzerinde işlem yapmaktadır. Şekil 4.22’de gösterilen tanımlı ikinci fonksiyon ile Fonksiyon aşırı yükleme özelliğinden faydalanılarak birden fazla ifade insan model üzerinde uygulanabilmektedir. Örneğin modelin yanakları gülümseme yönünde ilerletilirken kaşları da aynı anda bir miktar yukarıya kaldırılabilir.

```

void ChangeShape(int blendIndex,float degismeOran)
{
    for(int i=0;i<blendShapeLenght;i++)
        skinnedMeshRenderer.SetBlendShapeWeight(i,0);

    skinnedMeshRenderer.SetBlendShapeWeight(blendIndex,degismeOran);
}

void ChangeShape(int blendIndex,int blendIndex2,float degismeOran)
{
    for(int i=0;i<blendShapeLenght;i++)
        skinnedMeshRenderer.SetBlendShapeWeight(i,0);

    skinnedMeshRenderer.SetBlendShapeWeight(blendIndex,degismeOran);
    skinnedMeshRenderer.SetBlendShapeWeight(blendIndex2,degismeOran);
}

```

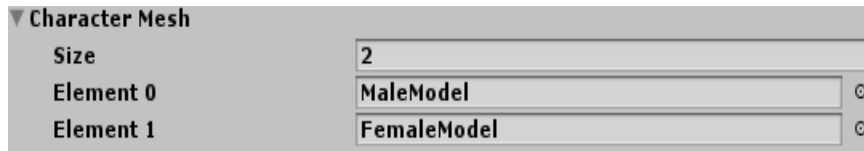
Şekil 4.22. Yüz ifade değişikliğini sağlayan aşırı yüklenmiş fonksiyonlar

ModelMeshScale.cs: Bu script Kinect önünde bulunan kullanıcının belirlenen eklem noktaları arasını metre cinsinden ölçerek bu ölçülen değerlere göre modelin beden ve boy ölçüsünü ayarlamaktadır. Parametrelerin yorumlanması ile insan model beden ölçüsü “Small, Medium, Large veya Xlarge” olarak ayarlanmakta ve boy ölçüsü “Short, Medium, Long” olarak belirlenmektedir. Şekil 4.13 belirli eklemlere göre otomatik ölçüm sağlayan kod parçasından bir görünüm ve Şekil 4.23 hesaplanan bilgilere göre değerlendirilen kullanıcı ölçümlerinin arayüz üzerinde gösterilmesini sunmaktadır.



Şekil 4.23. Kullanıcı beden ölçümlerinin ekran üzerinde yansıtılması

ModelWear.cs: Tüm modellerin, modellere ait ten renkleri, giysi, pantolon, ayakkabı vb. bileşenlerinin dokularının materyaller halinde dizi elemanları içerisinde tutulduğu ve ilgili butonlar tıklandığında indis bilgisi üzerinden çağrıldığı kod tasarımıdır.



Şekil 4.24. Script arayüzü üzerinde istenilen sayıda model uygulamaya eklenebilir

Kinect ile ilgili scriptlerin sağlanmasında Unity 3D Asset Store üzerinde sunulan kinect örnek projelerinden ve Unity 3D için MS SDK ile oluşturulmuş örnek dosyalardan faydalanılmıştır.

KinectWrapper.cs: Wrapper sınıfı genellikle bir modeli Kinect ile birlikte kullanmak için gerekli yapıları ve DLL eklentilerini içermektedir.

KinectManager.cs: Kinect ile ilgili temel fonksiyonların bulunduğu script dosyasıdır. Kinect'in kullanıma hazır olması, derinlik haritası değerleri, renk bilgisi, eklem noktaları, iskelet takibi vb. tüm fonksiyonlar bu script ile tanımlıdır.

AvatarController.cs: model eklem noktaları ile kullanıcının eklem noktalarının eşleştirildiği script dosyasıdır.

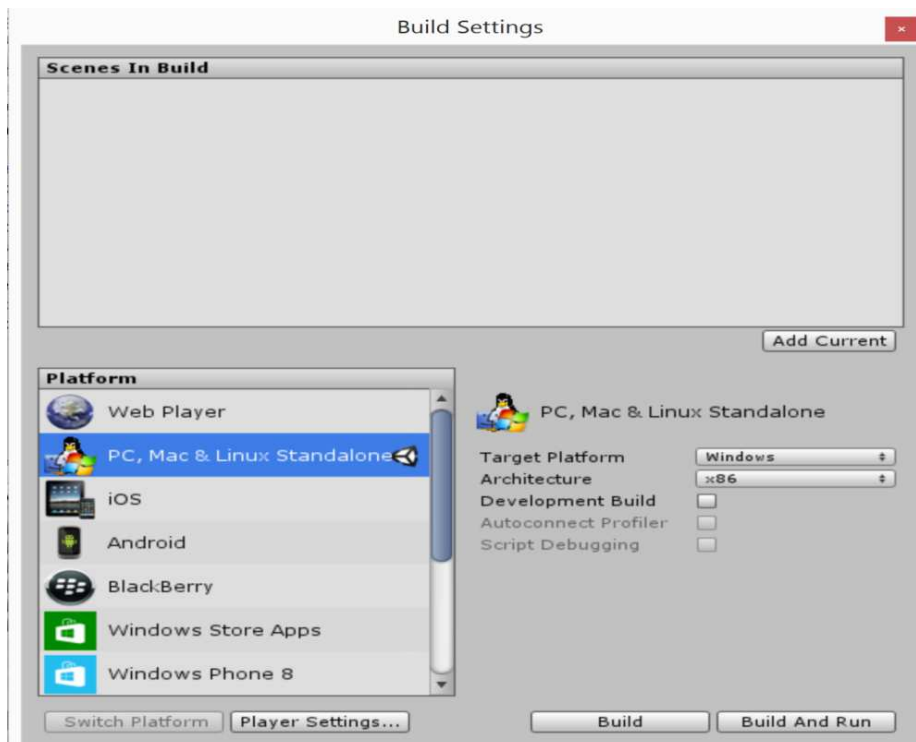
Şekil 4.25'de Kinect'in algıladığı kullanıcının hareketini aynen uygulayan, el hareketleri ile ekrandan seçim yapabilen model, uygulama arayüzü üzerinde gösterilmiştir.



Şekil 4.25. Kullanıcı hareketlerini aynen uygulayan gülümseyen bir avatar

### 4.3.3. Uygulamanın Farklı Platformlara Göre Derlenmesi

Sanal Giyinme kabini uygulaması, Windows, Linux ve Mac. sistemlerde çalışabilecek şekilde Unity içerisinde derlenerek uygulamanın çalıştırılabilir dosyası (.exe) oluşturulmuştur.



Şekil 4.26. Unity ile geliştirilen uygulamaların farklı platformlar için derlenmesi

Uygulamanın Kinect bağı arayüzünden bir görünüm şekil 4.27’de gösterilmektedir.



Şekil 4.27. Kinect bağı arayüzden genel bir görünüm

## **BÖLÜM 5. SONUÇLAR VE ÖNERİLER**

Bu çalışmada alışveriş ve giysi sektöründe kullanılabilecek sanal giysi deneme kabini uygulaması detaylarıyla açıklanmıştır. Mağazalarda giysi deneme ve uzun kuyruklar oluşması, internet ortamında yapılan alışverişlerde giysiyi deneme imkanının bulunmaması gibi olumsuz durumların giderilmesi amaçlanmıştır. Sanal giysi deneme kabini oluşturulmuş, giysi deneme kabininde kullanılmak üzere kullanıcının vücut yapısı ve karakteristiklerine göre parametrik olarak özellikleri değişen bir insan modeli oluşturulmuştur. Sistem içerisinde insan karakteristiklerini ve hareketlerini elde etmek için Microsoft Kinect sensörü kullanılmıştır. İlk olarak gerçekçi bir 3B model açık kaynak kodlu bir yazılım olan MakeHuman ile oluşturulmuştur. Oluşturulan bu model 3B modelleme yazılımları ile düzenlenmiş, kıyafetler ve yüz ifadeleri oluşturulmuştur. Ardından Unity oyun motoru ve Kinect ile bu modelin kontrolü sağlanmıştır. Kullanıcı Kinect cihazı karşısına geçer ve bağımsız olarak hareket eder. Kinect cihazı ile algılanan vücut hareketleri ve karakteristikleri geliştirilen yazılım aracılığıyla model üzerinde gerçek zamanlı olarak gerçekleştirilir. Uygulama yazılımı çevrimi başladığında geliştirilen algoritma ile kullanıcı vücut ölçüleri otomatik olarak hesaplanır, hesaplama sonucu elde edilen beden ve boy değerleri modele otomatik olarak uygulanır ve bu bilgi ekran üzerinden kullanıcıya sunulur. Kullanıcı el hareketleri ile ekranda beliren giysi reyonundan istediği kıyafetleri seçebilir, istenilen yüz ifadesini modele ekleyebilir. Vücut beden ve boy ölçüleri uygulama grafik arayüzündeki düğmeler kullanılarak da değiştirebilir.

Bu çalışma, Unity 3D oyun motoru kullanılarak geliştirilmiştir. Bölüm 3.1.2'de belirtilen özelliklere sahip sistemler üzerinde gerçekleştirilmiş ve test edilmiştir. Sistemde gerçekçi bir izlenim elde edilmiştir. 3B sanal insan modeli ve seçilen 3B kıyafetler, kullanıcıya göre otomatik olarak yeniden boyutlandırılmaktadır. Böylece

farklı beden ölçülerine sahip kullanıcıların sistemi gerçeklik algısı içerisinde kullanabilmesi sağlanmıştır.

Uygulama Kinect destekli olarak çalışmaktadır. Modelin kullanıcıya göre otomatik olarak boyutlandırılması ve mouse vb. bir cihaz kullanılmaksızın el hareketleriyle sistemin kontrolünün gerçekleştirilmesi için Kinect sensörünün sistem ile entegre olması gerekir. Bu durumun sistem için bir dezavantaj olacağı düşünülmüş ve Kinect donanımı bulunmayan cihazlarda sistemin manuel olarak da çalışması sağlanmıştır.

Sistemde zaman zaman senkronizasyon bozuklukları görülebilmektedir. Örneğin kullanıcı hareket ederken sürekli hareket eden kullanıcı ve değişen beden ölçülerine göre kıyafetler de tam oturmama veya zamanla oturma gözlemlenmiştir. Bunun dışında kullanıcının tüm eklemleri ile model ilişkilendirildiğinden model sürekli hareket halinde olmakta ve sisteme dışarıdan başka bir kullanıcı müdahalesi veya cihazın eklem noktası okumasına bir engel bulunduğu zaman model üzerinde karışıklıklar oluşabilmektedir. Kullanıcı ekran üzerinde sağa sola döndüğü zaman bu eklem noktaları çok yakın olarak okunmakta ve hesaplama karışıklıkları ortaya çıkabilmektedir. Buna bağlı olarak mevcut uygulama için bu durum sadece belirli eklemler üzerine odaklanmakla veya yakın eklemlerin okunmasında toleranslı davranan algoritmalar geliştirmekle giderilebilir. Sistemdeki ani değişiklikler için filtre kullanılarak kararlılık artırılmıştır. Şekil 5.1'de oluşabilecek hatalardan bir örnek sunulmaktadır.





Şekil 5.1. Giysi ile model uyumsuzluğu

Gelecek çalışmalarda insan yüzünü görüntü işleme yöntemleriyle algılayan ve elde edilen yüz verilerine göre bu yüz şekillerine sahip bir 3B model geliştirilerek kullanıcının bire bir kendi 3B sanal modelini görebilmesi ve daha gerçekçi görünüm elde edilmesi sağlanabilir. Ayrıca kullanıcının 3B hareketleri üzerinde yoğunlaşan çeşitli uygulamalar geliştirilebilir. Günümüzde 3B model kullanım alanı çok geniş olmakla beraber sürekli artmaktadır. 3B modellerin bulunduğu her alanda bu modellerin donanım bağımsız olarak vücut hareketleriyle kontrol edilebilmesi sistemlerin daha gerçekçi ve kolay gerçekleştirilmesini sağlayacaktır. Bu çalışmada böyle bir kullanımın sadece alışveriş sektöründe sağlayacağı kolaylıklar üzerinde durulmuştur.

## KAYNAKLAR

- [1] CİVAN, M. & BAL, V. (2002). E-Ticaret ve KOBİ'lerin Geleceği, I. Ulusal Bilgi, Ekonomi ve Yönetim Kongresi, 10-11 Mayıs, Hereke-Kocaeli.
- [2] KAYABAŞI, A. "Elektronik (online) Alışverişte Lojistik Faaliyetlere Yönelik Müşteri Şikayetlerinin Analizi ve Bir Alan Araştırması", İşletme Araştırmaları Dergisi, 2:2, 2010, 21- 42.
- [3] CHEN, C.A. (2009). Information-Oriented Online Shopping Behavior in Electronic Commerce Environment, Journal of Software, 4(4):307-314.
- [4] SUKI, N.M. (2011). Cellular Phone Users' Experiences Towards Online Shopping, Information Management and Business Review, 2(1):38-45.
- [5] MESSMER, E., E-comm yet to embrace virtual reality. Network World 05.08.2000.
- [6] HOLLANDER, J., A Glimpse into the Future of E-Commerce. E-Commerce Times 04.02.1999; also available at: <http://www.ecommercetimes.com/news/articles/990204-4.shtml>.
- [7] ALGÜR, S., CENGİZ, F., "The Risk and Benefit Perceptions of Online Shopping According to the Turkish Consumers", Journal of Yasar University 2011 22(6) 3666-3680.
- [8] EUROSTATS (2010), Regional Yearbook 2010, Publication Office of the European Union, Belgium.
- [9] WANG, L., VILLAMIL, R., SAMARASEKERA, S., KUMAR, R, Magic mirror: A virtual handbag shopping system, 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 19-24, 16-21 Haziran 2012.
- [10] ZHOU, Z., SHU, B., ZHOU, S., DENG, X., TAN, P., LİN, D., Image-based clothes animation for virtual fitting, SIGGRAPH Asia 2012 Technical Briefs, 2012.

- [11] IŞIKDOĞAN, F., KARA, G., A Real Time Virtual Dressing Room Application using Kinect, CMPE537 Computer Vision Course Project, January 2012.
- [12] GIOVANNI, S., CHOI, Y.C., HUANG J., KHOO E.T., YIN, K., Virtual try-on using kinect and HD camera, 5th International Conference on Motion in Games, France, 55-65, 15 Kasım 2012.
- [13] CASAS, X., HIERRARA, G., COMA, I., FERNANDEZ, M., A Kinect-based Augmented Reality system for individuals with autism spectrum disorders ,international Conference on Computer Graphics Theory and Applications, Rome, 440-446, 2012.
- [14] BLUM, T., KLEEGER, V., BICHLMEIER, C., NAVAB, N., Miracle: An augmented reality magic mirror system for anatomy education, Virtual Reality Short Papers and Posters (VRW), 115-116, 2012.
- [15] SANNA, A., ZUNINO, C., LAMBERTI, F., HAVS: a human animated VRLM-based virtual shop for e-commerce.
- [16] NARAIN, R., SAMİİ, A., O'BRIEN, J.F., Adaptive Anisotropic Remeshing for Cloth Simulation, ACM Transactions on Graphics, Vol. 31, No. 6, Article 152, Publication date: November 2012.
- [17] KAYAPA, N., Gerçek ve sanal gerçeklik ortamları arasındaki algısal farklılıklarda görselleştirmeye ilişkin özelliklerin araştırılması, Doktora, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü,2010.
- [18] PIMENTAL, K. ve TEIXERIA, K., Virtual Reality Through the New Looking Glass, 2nd Ed., McGraw-Hill,1993.
- [19] BAYRAKTAR, E., KALELİ, F., Sanal Gerçeklik ve Uygulama Alanları, Akademik Bilişim 2007, Dumlupınar Üniversitesi, Kütahya, 31 Ocak-2 Şubat 2007.
- [20] SUİ, Y., GENG, D., ALLEN, C.R., BURN, D., BELL, G.D. ve ROWLAND, R., "Three-Dimensional Motion System ("Data-Gloves"): Application for Parkinson's Disease and Essential Tremor", IEEE International Workshop on Virtual and Intelligent Measurement Systems, Budapest, Hungary, May 19-20,2001.
- [21] HIZAL, S., Histogram tabanlı algoritmalarla sanal giriş birimi tasarımı, Y.Lisans, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Haziran 2009.
- [22] <http://www.vrealities.com>, Erişim Tarihi: 20.05.2014.

- [23] [http://www.1gn.net/Sanal\\_Gerceklikte\\_Yeni\\_Boyut.html](http://www.1gn.net/Sanal_Gerceklikte_Yeni_Boyut.html), Eriřim Tarihi: 20.05.2014.
- [24] <http://www.virtuix.com/>, Eriřim Tarihi: 20.05.2014.
- [25] ÇAVAŞ, B., HUYUGÜZEL ÇAVAŞ, P., TAŞKIN CAN, B., Eğitimde sanal gerçeklik, The Turkish Online Journal of education Technology – TOJET October 2004 ISSN: 1303-6521 volume 3 Issue 4 Article 15.
- [26] <http://www.eonreality.com/press-releases/eon-reality-and-csm-brazil-announce-partnership-for-virtual-reality-interactive-soccer-game/>, Eriřim Tarihi: 20.05.2014.
- [27] <http://webcamsocialshopper.com/virtual-dressing-room-product-tour/virtual-dressing-room>, Eriřim Tarihi: 20.05.2014.
- [28] <http://www.ray-ban.com/international/virtual-mirror>, Eriřim Tarihi: 20.05.2014.
- [29] <http://www.3dmimarimodelleme.com/?Lang=TR&SyfNmb=2>, Eriřim Tarihi: 23.05.2014.
- [30] [http://en.wikipedia.org/wiki/3D\\_modeling](http://en.wikipedia.org/wiki/3D_modeling), Eriřim Tarihi: 23.05.2014.
- [31] ERDEMİR, F., ADİLOĞLU, F., Interactive 3D Animation and Modelling, Bitirme Tezi, İstanbul Kültür Üniversitesi, Mayıs 2008.
- [32] AYANOĞLU, M., Mimarlık Eğitiminde üç boyutlu bilgisayar oyunu motorlarının kullanımı, Y. Lisans, İstanbul teknik Üniversitesi, Ağustos 2006.
- [33] [http://tr.wikipedia.org/wiki/Oyun\\_motoru](http://tr.wikipedia.org/wiki/Oyun_motoru), Eriřim Tarihi: 23.05.2014.
- [34] <http://unity3d.com/pages/what-is-unity>, Eriřim Tarihi: 23.05.2014.
- [35] <http://www.needforspeed.com/media/screenshot/basepoint?game=3467>, Eriřim Tarihi: 23.05.2014.
- [36] <http://grafikod.org/oyun-motoru-nedir/>, Eriřim Tarihi:23.05.2014.
- [37] <http://www.needforspeed.com/media/screenshot/evasion-0?game=3467#node-51557>, Eriřim Tarihi: 23.05.2014.
- [38] HANGÜL, E., Nesneye Dayalı Yaklaşımla Mobil Cihazlar Üzerinde Üç Boyutlu Oyun Motoru Tasarımı ve Gerçekleştirimi, Y.Lisans, Ege Üniversitesi Fen Bilimleri Enstitüsü, Kasım 2007.

- [39] <http://www.xbox.com/en-US/xbox360/accessories/kinect>, Eriřim Tarihi: 25.05.2014.
- [40] SÜZEN, A. A., Kinect teknolojisi kullanılarak engelliler için ev otomasyonu, Y.Lisans, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü,2012.
- [41] BOZGEYİKLİ, E., Introducing rolling axis into motion controlled game play as a new degree of freedom using microsoft kinect, Y.Lisans, Ortadoęu Teknik Üniversitesi, Ağustos 2012.
- [42] <http://msdn.microsoft.com/en-us/library/jj131033.aspx>, Eriřim Tarihi: 25.05.2014.
- [43] YOLCU, G., Kinect destekli gerçek zamanlı sanal ayna tasarımı, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Ocak 2014.
- [44] [http://www.makehuman.org/doc/node/makehuman\\_and\\_its\\_purpose.html](http://www.makehuman.org/doc/node/makehuman_and_its_purpose.html), Eriřim Tarihi:25.05.2014.
- [45] KURNAZ, İ., DURGUT, R., Blender ve Ogre3D kullanarak 3 boyutlu benzetim, Karabük Üniversitesi.
- [46] BRİTO, A., Blender 3D: Architecture, Buildings, and Scenery, 2010.
- [47] <http://www.blender.org/features/>, Eriřim Tarihi: 25.05.2014.
- [48] [http://tr.wikipedia.org/wiki/Unity\\_3D](http://tr.wikipedia.org/wiki/Unity_3D), Eriřim Tarihi: 25.05.2014.
- [49] <http://social.msdn.microsoft.com/Forums/tr-TR/c44067b2-6e9c-41c1-81d5-2ef680aa8919/csharpa-giris-c-nedir?forum=csharptr>, Eriřim Tarihi: 25.05.2014.
- [50] ALGAN, S., Her Yönüyle C#. Pusula Yayıncılık ve İletişim Ltd. Şti, 780s, İstanbul,2009.
- [51] <http://en.wikipedia.org/wiki/FBX>, Eriřim Tarihi:26.05.2014.

## ÖZGEÇMİŞ

Muhammed KOTAN, 02.02.1989 da Muş'ta doğdu. İlköğretim ve lise eğitimini Muş'ta tamamladı. 2007 yılında Muş Anadolu Öğretmen Lisesi'nden mezun oldu ve yine aynı yıl içerisinde başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü 2011 yılında bitirdi. Kasım 2011'den itibaren bir yıl süreyle Afyon Kocatepe Üniversitesi Bilgisayar Mühendisliği Bölümü'nde Araştırma Görevlisi olarak çalıştı. Kasım 2012'de Öğretim Üyesi Yetiştirme Programı kapsamında atandığı Sakarya Üniversitesi Bilişim Sistemleri Bölümü'nde Araştırma Görevlisi olarak göreve başladı ve şu an halen görevine devam etmektedir.