İNSAN HAREKETLERİNİN ARTIRILMIŞ GERÇEKLİK ORTAMINA AKTARILMASI ÜZERİNE BİR ÇALIŞMA

A STUDY ON VISUALISATION OF THE HUMAN MOVEMENT INTO THE AUGMENTED REALITY ENVIRONMENT

DORUKHAN GÜLÖZ

Dr. Öğr. Üyesi SERDAR ARITAN

Tez Danışmanı

Hacettepe Üniversitesi, Bilişim Enstitüsü

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Grafiği Anabilim Dalı için Öngördüğü

Yüksek Lisans Tezi olarak hazırlanmıştır.

Nisan 2024

Dorukhan Gülöz'ün hazırladığı "**İNSAN HAREKETLERİNİN ARTIRILMIŞ GERÇEKLİK ORTAMINA AKTARILMASI ÜZERİNE BİR ÇALIŞMA"** adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR GRAFİĞİ ANABİLİM DALI**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Haşmet GÜRÇAY

Başkan ..……

Dr. Öğr. Üyesi Serdar ARITAN

Danışman ..……

Prof. Dr. Sadettin KİRAZCI

Üye ...……..

……..……

Bu tez Hacettepe Üniversitesi Bilişim Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak …. / ….. / ….. tarihinde onaylanmıştır.

Prof. Dr. Şahap Armağan TARIM

Bilişim Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Bilişim Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

….. / ….. / …..

DORUKHAN GÜLÖZ

YAYINLANMA FİKRİ MÜLKİYET HAKKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan "***Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge***" kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H. Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

□ Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.

□ Enstitü / Fakülte yönetim kurulu gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren .... ay ertelenmiştir.

□ Tezim ile ilgili gizlilik kararı verilmiştir.

..... / ..... /.....

DORUKHAN GÜLÖZ

ÖZET

İNSAN HAREKETLERİNİN ARTIRILMIŞ GERÇEKLİK ORTAMINA AKTARILMASI ÜZERİNE BİR ÇALIŞMA

Dorukhan GÜLÖZ

Yüksek Lisans, Bilgisayar Animasyonu Ve Oyun Teknolojileri Bölümü

Tez Danışmanı: Dr. Öğr. Üyesi Serdar ARITAN

Nisan 2024, 88 Sayfa

İnsan hareketi analizinin artırılmış gerçeklik ile birleştirilmesi, biyomekanik, tıp, robotik başta olmak üzere birçok bilim alanının ilgi konusu olmuştur. Literatürdeki çalışmalar genellikle artırılmış gerçeklik ortamında hareketsiz nesneler ile ilgilenir. Bu tez çalışmasında, bir insan modelinin artırılmış gerçeklik ortamında iskeleti yaratılarak hareket etmesi ve gerçek insanında bu hareketi taklit etmesi amaçlanmıştır. Önce bvh modeli ile bir insan iskeleti oluşturulmuş ve bu iskelete bir hareket kazandırılarak hareket eden bir model elde edilmiştir. Oluşturulan model ile tekrar eden bir animasyon artırılmış gerçeklik ortamına aktarılmıştır. Bu hareketli model üzerinden gerçek insanın oluşturduğu model üst üste oturtularak insanın hareketi öğrenmesi ve ne kadar doğru yapıp yapmadığı ile ilgili bir çıkarımda bulunup, hareketi iyileştirmeleri için bir temel ortam oluşturulması amaçlanmıştır.

Anahtar Kelimeler: Artırılmış gerçeklik, insan hareket analizi, unity oyun motoru, poz tahminleme

ABSTRACT

A STUDY ON VISUALISATION OF THE HUMAN MOVEMENT INTO THE AUGMENTED REALITY ENVIRONMENT

Dorukhan Gülöz

Master of Science, Department of Computer Graphics

Supervisor: Assistant Prof. Serdar ARITAN

Nisan 2024, 88 Pages

The integration of human motion analysis with augmented reality has gained significant attention in various fields of science, including biomechanics, medicine, and robotics. Most studies in the literature typically focus on stationary objects within an augmented reality environment. In this thesis, the aim is to create a skeletal structure of a human model in an augmented reality environment, allowing it to move, and then to mimic this movement in a real human. Initially, a human skeleton was constructed using a BVH mocap files. An animation generated from this file was then transferred into an augmented reality environment. The aim is to create a foundational environment for improving movement by overlapping the model generated by real human movements on top of this dynamic model to learn human movement and make inferences about how accurate it is

Keywords: Augmented reality, human, motion analysis, unity game engine, pose estimation

TEŞEKKÜR

Tez çalışmam boyunca bana yol gösteren, bilimsel anlamda ufkumu genişleten değerli danışmanım Dr. Öğretim Üyesi Serdar ARITAN'a teşekkür ederim.

Tez çalışmam boyunca ihtiyacım olan her anda yardımlarını esirgemeyen Ar. Gör. Nihat Şükrü ÖZGÖREN'e teşekkür ederim.

Her zaman yanımda olan, bilgi ve ışığıyla beni aydınlatan ve beden eğitimi öğretmeni annem Leyla GÜLÖZ'e, bana hayatta yol gösteren babam Erdal GÜLÖZ'e ve hayatımın her noktasında bana destek olan teyzem Lale ŞAHİN'e teşekkür ederim.

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

SİMGELER VE KISALTMALAR

Kısaltmalar

AR	Artırılmış Gerçeklik
BVH	BioVision Hierarchy
XR	Extended reality

1. GİRİŞ

1.1. Bilgisayar Grafiği

Günümüzde bilgisayarlar, iletişimden sanat ve eğlenceye kadar birçok alanda vazgeçilmez bir rol oynamaktadır. Bu alanlarda görsel iletişim, bilgisayar grafikleri sayesinde güçlenmektedir.

Bilgisayar grafikleri, bilgisayarların yardımıyla görsel içeriklerin oluşturulduğu ve işlendiği bir alanı ifade eder. Temel olarak, gerçek dünyadaki nesnelerin ve olayların dijital ortamda görsel temsiliyle ilgilenir. Bilgisayar grafiklerinin temel bileşenleri üç ana kategoride incelenebilir: modelleme, işleme ve görselleştirme. [1].

Modelleme, bilgisayar grafiklerinin ilk adımı, gerçek dünyadaki nesneleri ve sahneleri dijital ortamda temsil eden modellerin oluşturulmasıdır. Üç boyutlu (3B) modelleme, nesnelerin üç boyutlu geometrik şekillerini oluşturma işlemini ifade eder. İşleme, modelleme aşamasında oluşturulan veriler, işleme adımında düzenlenir ve düzeltilir. 3B modellerin yüzeyleri renklendirilir, doku eklenir ve aydınlatma efektleri uygulanır. İşleme aşaması, gerçekçi ve estetik açıdan çekici sonuçlar elde etmek için kritik bir aşamadır. Görselleştirme, modelleme ve işleme aşamalarından geçen veriler, son olarak görselleştirilir. Bu aşamada, nesnelerin ve sahnelerin görünüşü oluşturulur. Perspektif, kamera açıları, aydınlatma ve gölgeleme gibi faktörler, nihai görüntünün kalitesini etkiler. [1, 2].

Bilgisayar grafikleri, geniş bir yelpazede uygulama alanlarına sahiptir; film ve animasyon, sinema endüstrisi, bilgisayar grafiklerini geniş ölçekte kullanmaktadır. Özel efektler, dijital dublörler ve fantastik dünyaların yaratılması gibi alanlarda bilgisayar grafikleri büyük bir rol oynar. Oyun Geliştirme, video oyunları, etkileyici görsellerin olmazsa olmaz olduğu bir alandır. Oyun geliştiricileri, gerçekçi dünyalar oluşturmak ve oyuncuları içine çekmek için bilgisayar grafiklerini kullanır. Tıp ve eğitim, bilgisayar grafikleri, tıp alanında hastalıkların simülasyonu ve eğitim amaçlı kullanılabilecek interaktif 3B modeller oluşturmak için kullanılır.

Mimarlık ve tasarım, mimarlar ve tasarımcılar, binaların ve ürünlerin dijital prototiplerini oluşturarak tasarımlarını daha iyi görselleştirebilirler. Bilimsel görselleştirme, bilimsel verilerin görsel olarak temsili, karmaşık verilerin daha anlaşılır hale gelmesini sağlar. [2]

1.2. 3 Boyutlu Bilgisayar Grafiği

Üç boyutlu bilgisayar grafikleri, ilk basit çizimlerden günümüzün karmaşık ve gerçekçi görsellerine kadar uzanan etkileyici bir evrime sahiptir. Teknolojik ilerlemeler, bu alandaki gelişmeyi hızlandırmış ve farklı sektörlerde büyük etkiler yaratmıştır.

Üç boyutlu bilgisayar grafiklerinin temelleri, 1960'lı ve 1970'li yıllarda atıldı. Bu dönemde bilgisayarlar genellikle üniversiteler ve araştırma laboratuvarlarında bulunuyordu ve sınırlı hesaplama gücüne sahipti. İlk 3D görüntüler genellikle kabaca çizilmiş nesnelerdi ve temel olarak askeri ve bilimsel alanlarda kullanılıyordu. Sonra kablolu modelleme ve ilk render motorları 1980'ler de ortaya çıktı. 1980'lerde bilgisayarlar güçlenmeye başladıkça, 3 boyutlu grafiklerin gelişimi hız kazandı. Kablolu modelleme adı verilen yöntem, nesneleri temel geometrik şekilleriyle temsil etmeyi ve bunları birleştirerek sahneler oluşturmayı içeriyordu. Bu dönemde ilk render motorları geliştirildi. [3].

Render motorları, 3D model verilerini alır ve onları gerçekçi görüntülere dönüştürür. Dönüm noktası 1990’lar ve hızlandırıcılar oldu. 1990'lar, 3 boyutlu bilgisayar grafikleri için büyük bir dönüm noktasıydı. Bu dönemde grafik hızlandırıcıları ve GPU'lar (Graphics Processing Unit) geliştirilmeye başlandı. Bu donanımlar, render işlemlerini daha hızlı ve daha gerçekçi bir şekilde gerçekleştirmek için özelleştirilmiş işlem gücü sağlıyordu. Sinema Film endüstrisinin katılımıyla animasyon 2000'lerin başıyla birlikte, 3 boyutlu bilgisayar grafikleri özellikle film endüstrisinde büyük bir etkiye sahip oldu. Özel efektler ve dijital karakterler, büyük bütçeli filmlerin vazgeçilmez bir parçası haline geldi. Aynı dönemde video oyunları da 3 boyutlu grafiklerin kullanımını genişletti.

Günümüzde ileri teknoloji özellikle 2010'lar ve sonrasında 3 boyutlu bilgisayar grafikleri, inanılmaz bir seviyeye ulaştı. Yüksek çözünürlükler, fotogerçekçi aydınlatma ve detaylar, gerçek dünya ile dijital dünya arasındaki sınırları giderek daha belirsiz hale getiriyor. Sanal gerçeklik (VR) ve artırılmış gerçeklik (AR) gibi teknolojiler, 3 boyutlu grafikleri daha da ileri taşıyor ve kullanıcıları deneyimin içine çekiyor. [3, 4].

1.3. Artırılmış Gerçeklik

Günümüzün hızla gelişen teknolojisi, insanların gerçek dünya ile dijital dünyayı bir araya getirme yeteneğini artırıyor. Bu bağlamda artırılmış gerçeklik (AR), son yıllarda büyük bir ilgi ve gelişme görmüş bir teknolojidir.

AR, gerçek dünyanın görüntüsünü ve sesini dijital verilerle birleştirerek, kullanıcılara daha zengin ve etkileşimli deneyimler sunar. Artırılmış gerçeklik, dijital içeriklerin gerçek dünya ile birleştirildiği bir teknolojidir. Bu teknoloji sayesinde fiziksel dünya üzerindeki nesneler ve ortamlar, bilgisayar tarafından oluşturulan grafikler, metinler, sesler ve videolar gibi dijital verilerle zenginleştirilir. Bu veriler, genellikle bir akıllı telefon, tablet, gözlük veya başka bir AR cihazı aracılığıyla görüntülenir. Şekil 1-1 [5].

Şekil 1-1 AR Uygulaması Örnek Gösterimi ([40] numaralı kaynaktan alınmıştır)

AR teknolojisinin temel işleyişi, gerçek dünyanın algılanması, anlaşılması ve üzerine dijital verilerin eklenmesi şeklinde gerçekleşir. Bu işlem, genellikle şu algılama adımı ile başlar, bir AR cihazı, çevresini kamerası veya sensörleri aracılığıyla sürekli olarak tarar ve gerçek dünyayı algılar. Sonra işleme ile algılanan veriler, cihazın işlemcisi tarafından analiz edilir. Nesnelerin konumu, boyutu ve özellikleri belirlenir. Daha sonra eşleştirme adımında cihaz, algıladığı verileri önceden belirlenmiş dijital içeriklerle eşleştirir. Bu içerikler genellikle bir uygulama veya AR platformu aracılığıyla sağlanır. En son görüntüleme ile eşleştirilen dijital veriler, gerçek dünyanın görüntüsüne eklenir. Kullanıcı, AR cihazını kullanarak hem gerçek dünyayı hem de dijital içerikleri görebilir. [5].

Artırılmış gerçeklik, birçok farklı sektörde kullanım bulmuştur: Eğitim ve öğretim alanında öğrencilere interaktif öğrenme deneyimleri sunarak karmaşık konuları daha anlaşılır hale getirebilir.

Sağlık ve tıp alanında tıp öğrencileri ve cerrahlar, artırılmış gerçeklik kullanarak vücut içi organları daha iyi anlayabilir ve cerrahi prosedürleri canlandırabilir edebilir. [6].

Perakende ve alışveriş sektöründe mağaza içi deneyimi geliştirerek müşterilere ürünler hakkında daha fazla bilgi sunabilir ve ürünleri sanal olarak deneyebilirler. Mimari ve inşaat alanında mimarlar, artırılmış gerçeklik ile tasarladıkları binaları gerçek dünya ortamında görselleştirerek daha iyi bir anlayış elde edebilirler. Sanat ve eğlence de artırılmış gerçeklik, sanat eserlerini canlandırmak, etkileşimli sergiler oluşturmak ve eğlenceli deneyimler sunmak için kullanılabilir. Artırılmış gerçeklik teknolojisi, sürekli olarak gelişmeye devam ediyor. Daha hızlı işlemciler, daha gelişmiş algılama teknolojileri ve daha etkileyici grafikler, AR deneyimlerini daha da zenginleştirecek. Gelecekte, artırılmış gerçeklik, günlük yaşamın bir parçası haline gelebilir ve daha fazla sektörde kullanım bulabilir. [5,6]

1.4. İşaretleyici Kullanarak Hareket Yakalama

Günümüzde teknolojinin gelişmesiyle birlikte, insan hareketinin dijital olarak kaydedilip üç boyutlu bilgisayar grafiğine aktarılması sağlanmıştır.

Bu alanlarda hareket yakalama (motion capture) teknolojisi, karakterlerin ve nesnelerin hareketlerini dijital olarak yakalayarak gerçekçi animasyonların oluşturulmasına olanak tanır.

Hareket yakalama, gerçek dünyada insanların veya nesnelerin hareketlerini dijital formatta yakalayan bir teknolojidir. Bu teknoloji sayesinde canlı performanslar, bilgisayar animasyonlarına dönüştürülebilir. Hareket yakalama, gerçekçilik ve doğallık arayışında olan endüstrilerde oldukça yaygın bir şekilde kullanılmaktadır. [7,8].

Hareket yakalama, temel olarak şu adımları içerir: Veri toplama, bir veya daha fazla kameralı sistem, hareketi yakalamak için hareket eden nesnenin veya kişinin üzerindeki işaretleyicileri (marker) izler. Bu işaretleyiciler, hareketin üç boyutlu koordinatlarını belirlemek için kullanılır. Şekil 1-2

Şekil 1-2 Model üzerinde işaretleyiciler

Veri İzleme, kameralar, işaretleyicilerin konumunu sürekli olarak takip eder ve bu veriler bilgisayara iletilir. Veri İşleme, bilgisayar, izlenen işaretleyici verilerini analiz eder, işler ve hareketin doğru ve kusursuz bir şekilde yakalanmasını sağlar. Animasyon Oluşturma, işlenmiş veriler, dijital karakterlerin veya nesnelerin hareketini taklit edecek şekilde animasyonlara dönüştürülür. [7].

Hareket yakalama teknolojisi, birçok alanda kullanım bulur; Film ve televizyon alanında hareket yakalama, canlı aktörlerin hareketlerini dijital karakterlere aktarmak için sıkça kullanılır. Özel efektler, fantastik yaratıklar ve gerçekçi karakter hareketleri oluşturulmasında etkilidir. Video oyunları sektöründe ki video oyunları, oyuncuların daha gerçekçi ve etkileyici deneyimler yaşaması için hareket yakalama teknolojisini kullanır. Oyuncuların gerçek dünyadaki hareketleri, dijital karakterlere yansıtılır. Eğitim ve simülasyon alanında hareket yakalama, tıp, askeri eğitim, havacılık ve daha fazla alanda gerçekçi simülasyonlar oluşturmak için kullanılır. Sanat ve performans sanatlarında dansçılar, aktörler ve sanatçılar, sahnede veya ekranlarda daha yaratıcı ve etkileyici performanslar sunmak için hareket yakalama teknolojisini kullanabilirler. Rehabilitasyon alanında hareket yakalama, fizyoterapi ve rehabilitasyon süreçlerinde hastaların hareketlerini izlemek ve değerlendirmek için kullanılabilir. Hareket yakalama teknolojisi, hızla gelişen bir alandır.

Gelişen sensörler, daha hızlı işlemciler ve daha akıllı algoritmalar, hareket yakalama sistemlerini daha hassas, etkili ve kullanıcı dostu hale getirecektir. Ayrıca artırılmış gerçeklik (AR) ve sanal gerçeklik (VR) gibi teknolojilerle birleştirildiğinde, daha etkileyici ve etkileşimli deneyimler sunmak için kullanılabilecektir. [7,8,9]

1.5. İşaretleyicili Hareket Yakalama Teknolojileri ve Markaları

Her bir hareket yakalama teknolojisi farklı avantajlar ve dezavantajlar sunar ve kullanılan uygulamaya, ortama ve bütçeye bağlı olarak tercih edilir. Optik sistemler genellikle geniş alanlar ve yüksek doğruluk gerektiren uygulamalarda tercih edilir.[10] Genellikle uygun maliyetlidir. Kablosuz sistemleri mevcuttur. Optik engeller örneğin,

gölgeler veya parazitler performansı etkileyebilir. Dış mekan kullanımlarında ışık koşulları veya hava durumu problemlerine duyarlı olabilirler.[10]

İvmeölçer tabanlı sistemler taşınabilirlik ve düşük maliyet nedeniyle tercih edilebilir. Hızlı tepki süreleri vardır. Hassas hareketlerde sınırlı doğruluk sağlar. Hareketlerin sınırlı bir aralığını algılayabilir.

Manyetik sistemler, optik engellerin olduğu ortamlarda daha güvenilir olabilir. Optik sistemlere kıyasla daha az duyarlıdır. Manyetik alanın etkileşimiyle metal objelerin performansı etkileyebilir. Daha yüksek maliyetlidir.[10]

Hareket yakalama teknolojisinin temelini oluşturan markalar, kullanıcılarına yüksek hassasiyet ve doğruluk sunarak farklı endüstrilerde benzersiz çözümler sağlarlar. Vicon, uzun yıllardır hareket yakalama endüstrisinde lider konumda olan bir markadır. Optik kameralar ve özel yazılım kullanarak, sanatçıların, araştırmacıların ve mühendislerin karmaşık hareketleri yakalamasına ve analiz etmesine olanak tanır. [11]

OptiTrack, geniş kapsamlı kameralar ve yüksek çözünürlüklü lensler kullanarak profesyonel düzeyde hareket yakalama çözümleri sunar. Oyun geliştirme, film yapımı ve araştırma gibi birçok alanda tercih edilir. [11]

Xsens, taşınabilir hareket yakalama sistemleri sunan öncü markalardan biridir. Giyilebilir sensörler aracılığıyla gerçek zamanlı hareket verileri toplar ve analiz eder. Spor performansı izleme, tıp araştırmaları ve sanal gerçeklik uygulamaları gibi birçok alanda kullanılır. [11]

Leap Motion, el ve parmak hareketlerini hassas bir şekilde takip eden bir hareket yakalama teknolojisidir. Sanal gerçeklik deneyimleri, cerrahi eğitim simülasyonları ve interaktif sanat projeleri gibi alanlarda kullanılmaktadır.

1.6. Poz Tahminleme Yöntemi ile Hareket Yakalama

Pose estimation (poz tahmini) teknolojisi, insanların veya nesnelerin konumlarını ve duruşlarını algılamak için kullanılan bir bilgisayar görüşü yöntemidir. Bu teknoloji, kameralardan alınan görsel verileri analiz ederek insan vücudu, yüz, el veya başka nesnelerin 2D veya 3D koordinatlarını saptar. Pose estimation, sağlık, spor, oyun, güvenlik ve otomasyon gibi pek çok alanda uygulama bulmaktadır. [12]

Teknolojinin temel prensipleri şu şekilde özetlenebilir, Görüntü Yakalama ve İşleme ile başlanır. İlk adım, kameralar aracılığıyla görüntü veya video yakalamaktır. Bu görüntüler, duruş tahmini için gerekli verileri sağlar. Sonra özellik tespiti yapılır. Algoritma, insan vücudu gibi özgün noktaları (örneğin, eklemler) tespit eder. Bu noktaların konumları, duruşun anlaşılması için kritik öneme sahiptir. Daha sonra İskelet Modelleme başlar. Tespit edilen özellikler, insan iskeletini temsil eden bir model oluşturmak için birleştirilir. Bu model, vücudun farklı bölümlerinin nasıl hareket ettiğini gösterir. Bazı sistemler sadece iki boyutlu (2D) koordinatları tespit ederken Şekil 1-3, daha gelişmiş sistemler üç boyutlu (3D) duruş tahmini yapabilir. Pose estimation teknolojileri genellikle makine öğrenimi ve yapay zeka algoritmalarından yararlanır. Bu algoritmalar, büyük veri setleri üzerinde eğitilerek insan duruşunu daha doğru bir şekilde tahmin edebilir. [12]

Şekil 1-3 Pose Estimation ile İskelet Yapısının Gösterimi ([43])

Uygulama alanları olarak, Sağlık ve Fizyoterapi alanında Hasta rehabilitasyonunda ve fiziksel terapilerde duruş analizi yapmak için kullanılır. Spor Analizinde sporcuların performansını analiz etmek ve antrenman tekniklerini iyileştirmek için kullanılır. Eğlence ve Oyunlar alanında hareket tabanlı video oyunları ve sanal gerçeklik deneyimlerinde kullanıcıların hareketlerini algılamak için kullanılır. Güvenlik ve Gözetim için kamu alanlarında güvenliği sağlamak ve şüpheli davranışları tespit etmek için kullanılabilir. Robotik ve Otomasyon alanında robotların insanlarla etkileşimini ve çevresel adaptasyonunu geliştirmek için kullanılır. [13]

Pose estimation teknolojisi sürekli gelişmektedir. Yapay zeka ve makine öğrenimi algoritmalarının ilerlemesiyle birlikte, bu teknolojinin doğruluğu ve etkinliği artmaktadır. Gelecekte, daha karmaşık insan hareketlerinin daha doğru bir şekilde tespit edilmesi, geniş çaplı uygulamaların yolunu açabilir. Özellikle artırılmış gerçeklik deneyimleri, bu teknolojinin gelişimiyle daha etkileyici hale gelecektir. [12, 13]

1.7. Poz Tahminleme Kütüphaneleri

Uygulamalarda kullanılan birden çok kütüphane mevcuttur. En çok kullanılan 3 kütüphanenin avantaj ve dezavantajları incelenmiştir.

OpenCV (Open Source Computer Vision Library), bilgisayar görüşü projeleri için yaygın olarak kullanılan bir açık kaynaklı kütüphanedir. Pose estimation işlevselliği için birçok araç ve algoritma içerir. Avantajlarından, OpenCV açık kaynaklı bir projedir ve ücretsiz olarak kullanılabilir. Bu, birçok geliştirici ve araştırmacının erişim sağlayabilmesini kolaylaştırır. Windows, Linux, MacOS ve Android gibi birçok farklı işletim sistemi ve platformunda çalışabilir. Dezavantaj olarak, başlangıçta karmaşık gelebilecek bir uygulama programlama arayüzüne sahiptir ve bu nedenle yeni başlayanlar için öğrenme eğrisi olabilir. OpenCV, bellek yönetimi ile ilgili dikkat edilmesi gereken detaylara sahiptir. Bellek sızıntıları veya veri sıkıştırmaları gibi sorunlarla karşılaşabilirsiniz. OpenCV, 3D görüntü işlemeye tam olarak uygun değildir. 3D veri işlemek istiyorsanız, diğer özel kütüphanelere veya araçlara ihtiyacınız olabilir. [14, 15]

Başka bir kütüphane MediaPipe, Google tarafından geliştirilen MediaPipe, el, yüz, vücut ve nesne tespiti gibi birçok görsel işleme görevi için kullanılan bir kütüphanedir. Özellikle vücut pose estimation için MediaPipe Pose modeli kullanılabilir. Avantajlarını sayacak olursak, hız ve performansdan bahsedilebilir. MediaPipe, GPU hızlandırmayı kullanarak hızlı ve gerçek zamanlı görüntü işleme yetenekleri sunar. Bu, canlı video akışlarında veya gerçek zamanlı uygulamalarda kullanım için uygundur. MediaPipe, Android, iOS ve Linux gibi çeşitli platformlarda kullanılabilir. Bu, farklı mobil ve masaüstü uygulama geliştirme projeleri için uygundur.

Dezavantaj olarak donanım gereksinimlerini sayabiliriz. Gerçek zamanlı işleme ve yüksek performanslı görsel işleme görevleri için uygun bir donanım gereksinimi vardır. Bazı eski veya düşük güçlü cihazlarda performans sorunları yaşanabilir. [16]

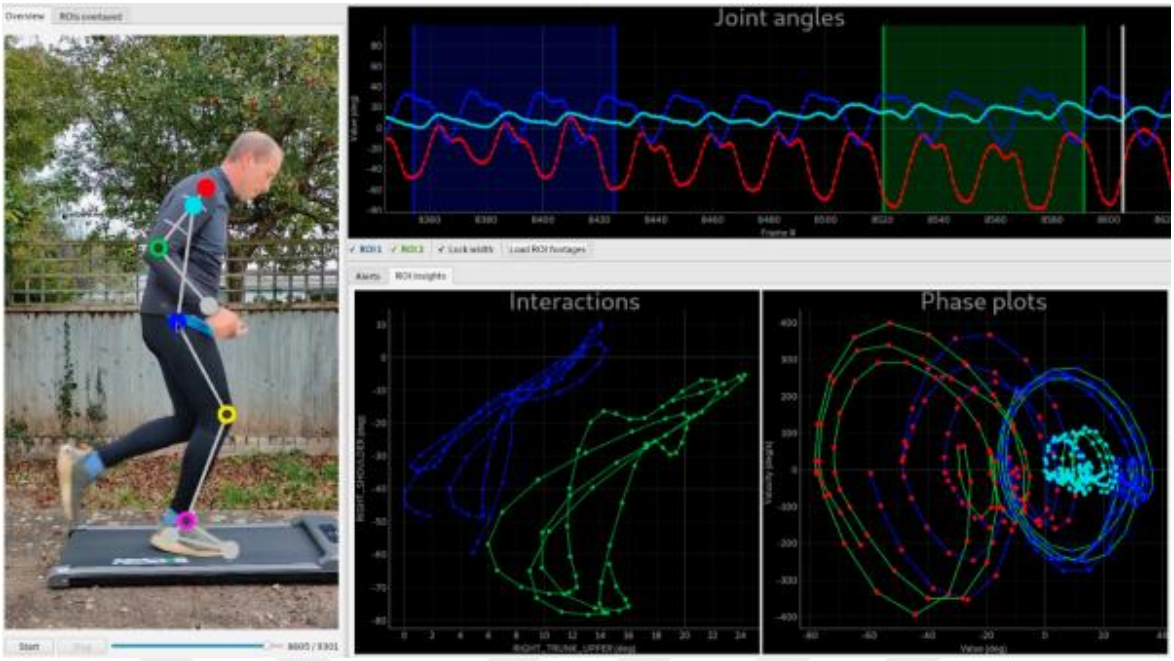
Bir diğer kütüphane Apple'ın Vision Framework'üdür. iOS ve macOS geliştiricileri için görüntü işleme ve görüntü analizi görevlerini kolaylaştıran bir araçtır. Bu framework, uygulamalarınızda görsel verileri işlemenize, analiz etmenize ve kullanmanıza olanak tanır. İçerisinde vücut pozlarının tahmin edilmesi gibi çeşitli görsel işleme yeteneklerini barındırır. Avantajlarından bahsedecek olursak, iOS ve macOS platformlarına doğrudan entegre edilmiştir. Bu, geliştiricilerin işletim sistemi düzeyinde bir görüntü işleme çözümüne kolayca erişmesini sağlar. Ayrıca Apple’ın geliştirmiş olduğu AR kütüphanesi ARKIT ile uyum içerisinde çalışmaktadır. Diğer avantajından biri Framework, görüntü işleme görevlerini hızlı ve verimli bir şekilde gerçekleştirir. Özellikle Apple'ın özel tasarım işlemcileri (örneğin, Neural Engine) ile kullanıldığında yüksek performans sağlar. Dezavantaj olarak yalnızca iOS ve macOS gibi Apple platformlarında kullanılabilir. Bir diğer dezavantajı ise yüksek performanslı işlemciler gerektirir, düşük güçlü cihazlarda kullanımı sınırlıdır. Pose estimation yapabilmesi için en az A12 bionic işlemci gerektirir. Bu da sadece 2018 yılı(Iphone XR) ve sonrası çıkan Apple cihazlarında çalışmaktadır. [17, 18]

1.8. Hareket Analizi

Hareket, insan yaşamının vazgeçilmez bir parçasıdır ve günlük aktivitelerden spor etkinliklerine kadar her alanda yer alır. Hareket analizi, insanların fiziksel aktivitelerini, hareketlerini ve duruşlarını anlamlandırmak için kullanılan bir teknoloji ve yöntemler bütünüdür. Hareket analizi, insanların hareketlerini bilimsel ve analitik bir yaklaşımla inceleyen bir disiplindir. Bu analiz, fiziksel aktivitelerin detaylı bir şekilde gözlemlenmesini, kaydedilmesini ve anlamlandırılmasını içerir.

Sağlık, spor, eğitim ve eğlence gibi birçok alanda kullanılan bu analiz, daha sağlıklı yaşam tarzları oluşturulmasından, performansın izlenmesine kadar birçok fayda sağlar. Hareket analizi, insan biyomekaniği (vücut mekaniği) ve motor kontrol (hareket kontrolü) prensiplerine dayanır ve genellikle bilgisayar destekli teknolojilerle gerçekleştirilir. Gelişen teknoloji ile birlikte hareket analizinin etkileri daha da genişleyecek ve yeni uygulama alanlarına yayılacaktır. [19]

Hareket analizi, genellikle aşağıdaki adımları içerir: veri toplama, hareket analizi için veri toplamak için sensörler, kameralar, manyetik alanlar gibi çeşitli teknolojiler kullanılır. Bu cihazlar, kişinin vücut pozisyonunu, eklemlerinin hareketini ve hareket yörüngelerini kaydeder. Veri İşleme, toplanan veriler, bilgisayar programları veya yazılımlar aracılığıyla işlenir. İşlem aşaması, verilerin düzenlenmesi, filtrelenmesi ve analiz için hazırlanmasını içerir. Analiz ve Yorumlama, işlenen veriler, uzmanlar tarafından analiz edilir. Bu aşamada, hareketin biyomekanik özellikleri, asimetri, hız, açılar gibi faktörler incelenir ve yorumlanır. Sonuçlar ve Uygulama, analiz sonuçları, ilgili alanda kullanılmak üzere raporlar, grafikler veya görseller halinde sunulur Şekil 1-4. Bu sonuçlar, hastalıkların teşhisinde, spor performansının izlenmesinde, rehabilitasyon planlarının oluşturulmasında ve daha birçok alanda değerli bilgiler sağlar.



Şekil 1-4 Hareket Analizi Raporlama Ekranı ([44])

Hareket analizi, birçok farklı alanda kullanım potansiyeline sahiptir. Tıp ve rehabilitasyon alanında, fizyoterapistler, hastaların hareketlerini analiz ederek doğru rehabilitasyon programları oluşturabilirler. Aynı zamanda, hastalıkların teşhisi ve tedavi süreçlerinin takibi için de kullanılır. Spor ve performans izlemede, sporcuların performanslarını değerlendirmek ve geliştirmek için hareket analizi kullanılır. Koçlar, sporcuların tekniğini ve hareket kalitesini inceleyerek daha etkili eğitim planları oluşturabilirler. Eğitim ve araştırma alanında öğrencilerin ve araştırmacıların hareketleri daha iyi anlamalarına yardımcı olmak için hareket analizi kullanılır. [20]

Oyun ve eğlence sektöründe video oyunları ve sanal gerçeklik uygulamaları, kullanıcıların fiziksel hareketlerini takip ederek etkileşimi artırmak ve daha gerçekçi deneyimler sunmak için hareket analizini kullanır. Hareket analizi teknolojisi, gelişen sensörler, artırılmış gerçeklik ve veri analizi yöntemleri ile birlikte daha da gelişmeye devam ediyor.

Bu teknoloji, insanların hareketlerini daha ayrıntılı ve hassas bir şekilde analiz etme yeteneğini artırarak sağlık, eğitim, spor ve daha birçok alanda faydalı olmaya devam edecek. [19, 20]

1.9. BVH

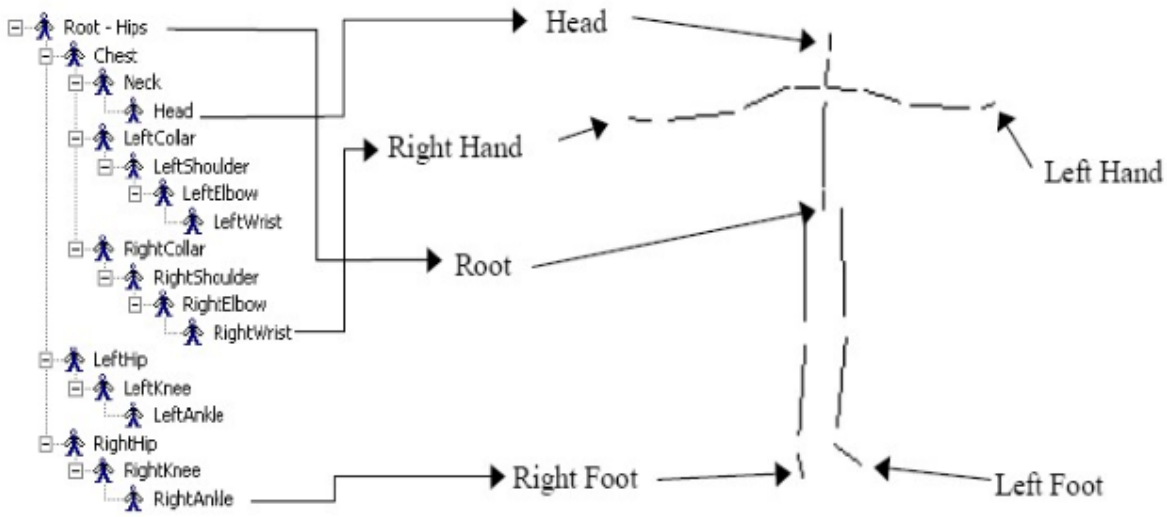
BioVision Hierarchy (BVH), hareket yakalama verilerini depolamak için kullanılan bir dosya biçimidir. Bu biçim, daha sonra Motion Analysis Corporation tarafından devralınan BioVision adlı bir şirket tarafından geliştirilmiştir. Biçim, her bir eklemin dönme ve çeviri verilerini içeren hiyerarşik bir yapıdan oluşur.

İnsan, bilgisayar grafikleri, oyun geliştirme, sanal gerçeklik ve hareket analizi gibi birçok alanda önemli bir konsept haline gelmiştir. Bu alandaki gelişmeler hem bilim dünyasında hem de endüstride büyük ilgi uyandırmıştır. İnsan modellemesi için kullanılan bir önemli araç olan BVH (Biovision Hierarchy) formatı bulunmaktadır. BVH, insan iskeletini ve hareketini temsil etmek için kullanılan bir dosya formatıdır. Bu format, bilgisayar grafikleri uygulamalarında ve hareket analizi alanında oldukça yaygın bir şekilde kullanılmaktadır.

BVH, insan hareketini temsil etmek için kullanılan bir ASCII (metin tabanlı) dosya formatıdır. Bu format, bir hiyerarşi ağacı kullanarak insan iskeletini ve hareketini tanımlar. BVH dosyaları genellikle bir dizi eklemlerin (kemiklerin) dönme açıları ve pozisyonlarını içerir. Bu bilgiler, bir bilgisayar programı veya uygulama tarafından kullanılarak insanın hareketinin canlandırılmasını sağlar. [21].

Bu formatın diğer bazı yaygın formatlarla karşılaştırılmasında dikkate alınması gereken bazı anahtar avantajlar ve dezavantajlar şunlardır: Hareket verilerinin ayrıntılı gösterimi, BVH, Şekil 1-5’ de gösterilen iskelet yapılarını ve hareket verilerini ayrıntılı bir şekilde ifade edebilir. Bu, animasyonlarda daha gerçekçi ve doğru hareketlerin oluşturulmasını sağlar. Geniş yazılım desteği, BVH, birçok profesyonel 3D modelleme ve animasyon yazılımı tarafından desteklenmektedir. Bu, kullanıcıların mevcut araçları kullanarak kolayca BVH dosyalarını içe aktarmasını ve işlemesini sağlar. Basit yapı, BVH dosyalarının yapısı görece basittir, bu da geliştiricilerin bu formatı kendi projelerine entegre etmesini kolaylaştırır. [22].

Bazı dezavantajları da şunlardır: Dosya boyutu, BVH formatı, detaylı hareket verilerini sakladığı için dosya boyutları büyük olabilir. Bu, depolama ve transfer için ek kaynak gereksinimine neden olur. Sınırlı kullanım alanları, BVH, özellikle hareket yakalama verileri için tasarlandığından, bu özgün amacın dışındaki kullanımlarda sınırlı olabilir. Uyumluluk sorunları, bazı yazılımlar, BVH formatındaki verileri doğru bir şekilde okuyamayabilir veya içeri aktarırken hatalara neden olabilir.



Şekil 1-5 Bvh İnsan İskelet Modeli

BVH dosyaları, genellikle çok daha karmaşık ve uzun olabilir, ancak Şekil 1-6 ve Şekil 1-7 de gösterilen temel bileşenleri açıklamak için bu örnek kullanışlı olacaktır.

```
HIERARCHY
ROOT Hips
        {
                OFFSET 0.00 0.00 0.00
                CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
                JOINT Chest
                {
                        OFFSET 0.000000 6.275751 0.000000
                        CHANNELS 3 Zrotation Xrotation Yrotation
                        JOINT Neck
                        {
                                OFFSET 0.000000 14.296947 0.000000
                                CHANNELS 3 Zrotation Xrotation Yrotation
                                JOINT Head
                                {
                                        OFFSET 0.000000 2.637461 0.000000
                                        CHANNELS 3 Zrotation Xrotation Yrotation
                                        End Site
                                        {
                                                OFFSET 0.000000 4.499004 0.000000
                                        }
                                }
                        }
```

Şekil 1-6 Bvh Dosya yapısı Parent ve Chil Nodelarının Pozisyon ve Rotasyon bilgilerinin yer aldığı Kısım

```
MOTION
Frames: 2
Frame Time: 0.04166667
-9.533684       4.447926        -0.566564       -7.757381       -1.735414       89.207932       9.763572
                6.289016        -1.825344       -6.106647       3.973667        -3.706973       -6.474916
                -14.391472      -3.461282       -16.504230      3.973544        -3.805107       22.204674
                2.533497        -28.283911      -6.862538       6.191492        4.448771        -16.292816
                2.951538        -3.418231       7.634442        11.325822       5.149696        -23.069189
                -18.352753      15.051558       -7.514462       8.397663        2.953842        -7.213992
                2.494318        -1.543435       2.970936        -25.086460      -4.195537       -1.752307
                7.093068        -1.507532       -2.633332       3.858087        0.256802        7.892136
                12.803010       -28.692566      2.151862        -9.164188       8.086427        -5.641034
                -12.596124      4.366460
-8.489557       4.285263        -0.621559       -8.244940       -1.784412       90.041962       8.849357
                5.557910        -1.926571       -5.487280       4.119726        -4.714622       -5.790586
                -15.218462      -3.167648       -15.823254      3.871795        -4.378940       22.399654
                2.244878        -29.421873      -6.918557       6.131992        4.521327        -18.013180
                3.059388        -3.768287       8.079588        10.124812       5.808083        -22.417845
                -15.736264      18.827469       -8.070700       9.689109        2.417364        -7.600582
                2.595005        -1.625679       2.430162        -27.579708      -3.852241       -1.830524
                12.520144       -1.653632       -2.688550       4.545600        0.296320        8.031574
                13.837914       -28.922058      2.077955        -9.176716       7.166249        -5.170825
```

Şekil 1-7 Bvh Dosyası Hareket Verilerinin Bulunduğu Kısım

Bvh dosyasındaki bölümlerin ne anlama geldiklerini inceleyelim.

Hierarchy: Bu bölüm, hiyerarşik yapıyı (düğümler arasındaki ilişkileri) tanımlar. Her düğümün (joint) adı, pozisyonu, rotasyon kanalları (channels), alt düğümler (eğer varsa) ve bunların başlıklarını içerir. "ROOT" düğümü genellikle karakterin veya nesnenin ana düğümünü temsil eder ve altındaki düğümlerle bağlantılıdır. [23,24].

Motion: Bu bölüm, animasyonun zaman içinde nasıl değiştiğini ve düğümlerin her karedeki durumunu içerir. "Frames" alanı, animasyonun toplam kare sayısını belirtir. "Frame Time" alanı, her bir karenin ne kadar süreyle (saniye cinsinden) temsil edildiğini belirtir. Ardından, her kare için düğümlerin pozisyon ve rotasyon bilgileri sırayla verilir. [23,24].

Offset (Düzeltme): "Offset," bir düğümün (joint) ebeveynine (parent) göre lokal konumunu ve rotasyonunu belirlemek için kullanılır. Her düğümün "offset" değeri, üst düğüme göre yerel bir koordinat sistemini temsil eder. Bu değer, bir düğümün ebeveynine göre ne kadar uzakta olduğunu ve bu düğümün hareketlerinin nasıl hesaplanması gerektiğini gösterir. "Offset" değerleri, üç bileşen içerebilir: x, y ve z ekseni boyunca uzaklık ve rotasyon için açısal değerler. Bu değerler genellikle BVH dosyasının başında düğümlere atanır ve her düğüm için ayrıca tanımlanır. [23,24].

Channels (Kanallar): "Channels," bir düğümün (joint) hareketini tanımlayan dönme ve/veya pozisyon değişikliklerini belirtir. Her düğümün "channels" tanımı, düğümün belirli eksenler etrafındaki rotasyonunu veya pozisyon değişikliğini tanımlar. Bu tanımlama, karakterin veya nesnenin belirli bir düğümünün nasıl hareket edeceğini belirler. "Channels" tanımı, genellikle "Xrotation," "Yrotation," "Zrotation," "Xposition," "Yposition," ve "Zposition" gibi kanalları içerir. Bu kanallar, düğümünün belirli bir eksende dönme veya pozisyon değiştirme yeteneğini temsil eder. [23,24].

BVH dosyasının içindeki her düğümün "offset" ve "channels" tanımları, karakterin veya nesnenin her bir parçasının nasıl hareket edeceğini ve birbirine nasıl bağlandığını tanımlar. Bu tanımlamalar, BVH dosyasının yorumlanmasını ve animasyonun oluşturulmasını mümkün kılar. [23,24].

1.10. Dosya Formatları

Bvh’dan farklı olarak birçok farklı dosya formatı bulunmaktadır. Bunların kendine göre avantajı ve dezavantajları vardır.

Bunları incelersek, FBX (Filmbox), FBX, BVH'dan daha kapsamlı bir 3D veri formatıdır. Model, animasyon ve diğer 3D bilgileri içerebilir. Ancak, daha karmaşık yapısı nedeniyle, bazı durumlarda daha fazla kaynak gerektirebilir. [25].

COLLADA, Açık standart bir format olan COLLADA, 3D nesneler, sahneler ve animasyonlar için geniş destek sunar. Bu format, farklı yazılımlar arasında veri aktarımı için ideal olabilir, ancak BVH kadar hareket yakalama verilerinde uzmanlaşmamıştır. [26]

Motion Capture Formats (C3D, ASF/AMC), Bu formatlar, özellikle hareket yakalama için tasarlanmıştır ve BVH gibi detaylı hareket verileri sağlarlar. Ancak, bunlar genellikle daha spesifik yazılımlar veya donanımlarla sınırlıdır. [27]

Bu formatlar çeşitli yazılım ve donanımlar ile birlikte kullanma zorunlulukları bulunmaktadır. Makine kodu dosya formatında oldukları için bvh dosya formatı gibi metin editörleri ile kolayca açılıp, görüntülenip, üzerinde değişiklik yapılamazlar. [25, 26, 27]

1.11. Oyun Motoru

Oyun motorları, günümüzün video oyun endüstrisindeki başlıca teknolojik unsurlardan birini temsil eder. Oyun geliştiricilerin oyunlarını daha hızlı, daha verimli ve daha etkileyici bir şekilde yaratmalarına olanak tanır. Bu motorlar, grafiklerden fizik simülasyonlarına, yapay zeka sistemlerinden oyun mekaniğine kadar birçok kritik bileşeni içerir. Oyun motorları, farklı platformlarda (bilgisayarlar, konsollar, mobil cihazlar) çalışabilme yeteneği sayesinde, oyun geliştiricilerine geniş bir kullanıcı kitlesine ulaşma olanağı sunar. Ayrıca, oyun motorları oyun geliştiricilerinin yoğun rekabetin olduğu bu sektörde başarılı ve kaliteli oyunlar üretmelerine yardımcı olur. Bu nedenle, oyun motorları oyun endüstrisinin temel taşlarından biri olarak kabul edilmektedir. [28]

Oyun motorları, genellikle birçok bileşeni içerir. Bu bileşenler, grafik motorları, ses motorları, fizik motorları, yapay zeka sistemleri, dünya oluşturma araçları, animasyon, görsel efektler, ağ özellikleri ve daha birçok özelliği içerir Şekil 1-8.

Şekil 1-8 Oyun Motoru Bileşenleri([45] referanstan alınmıştır)

Oyun motorları, geliştiricilerin bu bileşenleri kolayca bir araya getirmelerini ve oyunlarını oluşturmalarını sağlar. Ayrıca, oyun motorları, farklı platformlarda (PC, konsol, mobil cihazlar) çalışabilme yeteneğine sahiptir, bu da oyunların daha geniş bir kitleye ulaşmasını sağlar. [28,29].

Oyun motorları, birçok avantaj sunar. Hızlı geliştirme, oyun motorları, oyun geliştirme sürecini hızlandırır, çünkü birçok temel bileşeni içerir ve geliştiricilere tekrar tekrar aynı şeyleri oluşturma zorunluluğunu ortadan kaldırır. Verimlilik, oyun motorları, kaynakları daha verimli bir şekilde kullanmaya yardımcı olur. Bu, geliştiricilerin daha fazla içeriğe odaklanmalarına olanak tanır. Çapraz platform desteği, oyun motorları, farklı platformlarda oyunların çalışmasını kolaylaştırır, bu da oyunların daha fazla kullanıcıya ulaşmasını sağlar. [30].

Bazı örnek oyun motorlarından örnek vermek gerekir ise;

Unity, dünya genelinde en popüler oyun motorlarından biridir ve geniş bir kullanıcı tabanına sahiptir. Avantajları şunlar olabilir: çapraz platform desteği, Unity, farklı platformlarda (PC, konsol, mobil, VR/AR) çalışabilme yeteneği sunar, böylece geliştiriciler oyunlarını daha geniş bir kitleye ulaştırabilir. Geniş topluluk, Unity, büyük bir kullanıcı topluluğuna sahiptir ve bu, geliştiricilerin sorunlarını çözmelerine yardımcı olabilir. Asset store, Unity, Asset store adı verilen bir mağaza sağlar ve bu mağaza, geliştiricilere hazır varlıkları (modeller, sesler, efektler) satın alma veya paylaşma

olanağı sunar. Dezavantajlar: Lisans ücreti, Unity'nin ücretsiz sürümü bulunsa da, büyük ölçekli projeler veya ticari kullanımlar için ücretli lisans gerekebilir. [30]

Diğer bir oyun motoru Unreal Engine, özellikle görsel kaliteye odaklanan büyük bütçeli oyunlar için tercih edilen bir oyun motorudur. Görsel kalite, Unreal Engine, foto-gerçekçi grafikler ve görsel efektler oluşturmak için güçlü bir araç seti sunar. Ücretsiz kullanım, Unreal Engine, projenizin ticari başarısızlığı durumunda ücretsiz kullanım sunar ve sadece projeniz başarılı olduğunda ücretlendirme yapar.

Dezavantaj olarak: Öğrenme eğrisi, Unreal Engine'in karmaşıklığı, yeni başlayanlar için öğrenme eğrisini zorlaştırabilir. Bellek kullanımı, Unreal Engine, büyük ve ayrıntılı dünyalar oluşturulduğunda yüksek bellek kullanımına neden olabilir. Mobil uygulama yapmak için de performansı düşüktür. [31].

Diğer bir oyun motoru da Godot, bağımsız geliştiriciler ve stüdyolar için popüler bir açık kaynak oyun motorudur. Avantaj olarak açık kaynak bir proje olarak geliştirilmektedir ve ücretsiz olarak sunulmaktadır. Bu, geliştiricilerin Godot'u kullanarak oyunlarını geliştirirken lisans maliyetlerinden kaçınmalarını sağlar. Godot, farklı platformlarda çalışabilme yeteneği sunar. Bu, oyun geliştiricilerinin oyunlarını PC, konsol, mobil cihazlar ve web gibi farklı platformlara kolayca taşıyabilmelerini sağlar. Fakat dezavantaj olarak; Godot, Unity ve Unreal Engine gibi daha büyük ve önde gelen oyun motorlarına kıyasla daha az kullanıma ve topluluğa sahiptir. Bu, Godot'ta geliştirilen projelerin daha az kapsamlı ve özellikli olmasına neden olur. Karmaşık AR VR gibi projeler için bazı önde gelen rakiplere kıyasla daha zayıftır. Bu nedenle AR, VR desteği isteyen projeler için diğer motorlar daha uygun olabilir. [31]

1.12. Unity Oyun Motorunda Animasyon

Unity oyun motorunda animasyon oluşturmak için kendi içinde mevcut bir yapısı mevcuttur. Hareket oluşturmak için Unity içerisinde update mekanizmalarından faydalanılır. Şekil 1-9 de belirtildiği üzere iki tane update metodu vardır. Oyun motoru bu metotları sürekli çağırır.

Şekil 1-9 Unity Update Mekanizmaları

Update fonksiyonu, her karede bir kez çağrılır ve oyunun ana güncelleme döngüsüdür. Genellikle karakter hareketleri, kullanıcı girdilerinin alınması ve oyun içi mantığın çoğu burada işlenir.

FixedUpdate fonksiyonu, fizik güncellemeleri için kullanılır ve sabit bir zaman aralığında çağrılır. Fizik hesaplamaları genellikle burada yapılır. FixedUpdate, Update'ten farklı olarak belirli bir zaman aralığında sabit bir şekilde çağrılır, bu da onu fizik hesaplamaları için daha uygun hale getirir. [32]

1.13. Unity AR Foundation

Unity AR Foundation, mobil cihazlar için zengin ve etkileşimli AR deneyimleri yaratmak isteyen geliştiriciler için güçlü, esnek ve erişilebilir bir araçtır. Bu aracın temel özellikleri ve kullanılmasında bazı avantajlar vardır.

Çapraz platform desteği sunar. AR Foundation, iOS ve Android gibi çeşitli platformlarda çalışabilen uygulamalar geliştirmenize olanak tanır. Bu, geliştiricilerin tek bir kod tabanı üzerinden birden fazla platforma uygulama dağıtabilmeleri anlamına gelir. [33]

Entegre AR özellikleri bulunur. Unity'nin AR Foundation'ı, yüz algılama, düzlem algılama, görüntü ve nesne takibi gibi bir dizi AR özelliğini destekler. Bu özellikler, gerçek dünya ortamlarına sanal nesneler yerleştirmek için kullanılabilir. [34]

Genişletilebilir bir yapıdadır. AR Foundation, çeşitli ARKit ve ARCore özelliklerini entegre ederken, aynı zamanda özelleştirilebilir ve genişletilebilir. Geliştiriciler, özel efektler ve işlevler ekleyerek uygulamalarını özelleştirebilirler. [35]

Gelişmiş takip ve stabilizasyon ortamı sunar. AR uygulamalarının temel bir gerekliliği olan hassas ve güvenilir takip, AR Foundation ile sağlanır. Bu, sanal nesnelerin gerçek dünyadaki nesnelerle doğru bir şekilde hizalanmasını ve etkileşimini sağlar. [35]

Kullanım kolaylığı vardır. Unity'nin kullanıcı dostu arayüzü ve geniş kaynak kitaplığı, geliştiricilere AR uygulamalarını hızlı ve kolay bir şekilde geliştirme imkanı sunar. [35]

Sürekli güncellemeler ve yenilikleri içerir. Unity, AR Foundation'ı sürekli olarak günceller ve yeni AR teknolojilerini ve özelliklerini entegre eder, böylece geliştiriciler en son trendlerden yararlanabilirler. [33, 35]

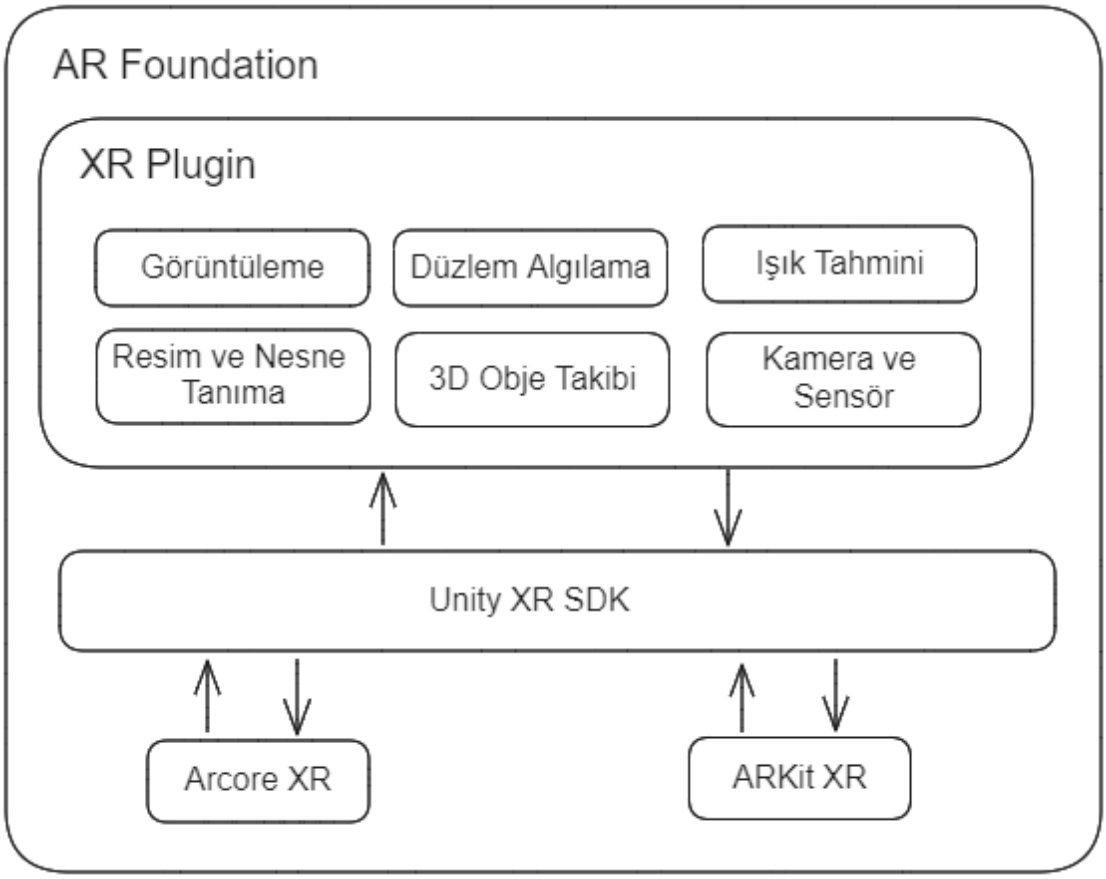
1.14. Unity AR Foundation'ın Desteklediği Pluginler

Ar uygulamaları geliştirmek için platforma özgü pluginler kullanmak zorunludur. Unity AR foundation kendi bünyesinde iki önemli plugin sunar. Bunlardan biri Android cihazlarda AR uygulamaları geliştirmek için Google ARCore XR plug-in kullanılır. Diğeri ise Apple cihazlarda AR uygulamaları geliştirmek için kullanılır. İsmi Apple ARKit XR plug-in'dir. Apple ARKit XR plugin ve Google ARore XR pluginleri Unity için bir eklentidir ve Apple'ın ve Google'ın ARKit teknolojisini Unity oyun motoru ile entegre etmeyi sağlar. Bu entegrasyon, Ios ve Andorid cihazları üzerinde çalışan artırılmış gerçeklik (AR) uygulamalarının geliştirilmesine olanak tanır. Bu eklentiler, Unity'nin görsel editörü ve sağladığı teknoloji birleştiğinde, geliştiriciler etkileyici ve etkileşimli AR deneyimleri hızlı ve etkili bir şekilde oluşturabilirler. Unity AR Foundation ile pluginlerin ilişkisini anlatan mimari çizelgeye Şekil 1-10 de görebilirsiniz. Apple ARKit XR pluginin kullanımın sağladığı avantajlara bakacak olursak: [35, 36]

Apple'ın ARKit'i, iOS cihazlarda güçlü ve gerçekçi AR deneyimleri sunmak için tasarlanmıştır. Kamera ve sensör verilerini kullanarak, gerçek dünyaya sanal nesneler yerleştirebilir.[36]

Unity ile entegrasyonu vardır. ARKit XR plugin, ARKit özelliklerini Unity'nin kullanıcı dostu arayüzü ve araçları ile birleştirir. Bu, geliştiricilerin Unity'nin sahip olduğu geniş özellik setinden yararlanarak iOS için AR uygulamaları geliştirmelerini sağlar. Özellik ve işlevsellikleri olarak sayabileceğimiz birkaç madde vardır. Düzlem algılama, yatay ve dikey düzlemleri algılayabilir, bu sayede sanal nesneleri doğru bir şekilde gerçek dünya ortamlarına yerleştirebilirsiniz. Işık tahmini yapılabilir. Gerçek dünyadaki ışık koşullarını taklit ederek, sanal nesnelerin daha gerçekçi görünmesini sağlar. Resim ve nesne tanıma yapılabilir. Önceden tanımlanmış resimleri ve nesneleri tanıyabilir, bu sayede özel içerik ve bilgileri gösterebilir. Üç boyutlu obje takibi yapabilir. Gerçek dünyadaki üç boyutlu nesneleri tanıyabilir ve takip edebilir. [37]

Performans ve optimizasyon ayarlamalarında bulunur: Apple'ın donanımı ve ARKit'in verimliliği sayesinde, ARKit XR plugin yüksek performanslı ve optimize edilmiş AR deneyimleri sunar. Geliştirici desteği ve topluluk sunar. Unity'nin geniş geliştirici topluluğu ve çeşitli eğitim materyalleri, ARKit XR plugin ile çalışırken yardımcı olabilir. [36, 37]



Şekil 1-10 Unity AR Foundation Mimari Yapısı

1.15. Tezin Amacı

Arttırılmış gerçeklik ortamında referans insan modeli hareketlendirilmiştir. Gerçek insanın bu hareketi taklit etmesi amaçlanmıştır . Önce çıkarılan bvh dosyasından bir insan iskeleti oluşturulmuş ve bu iskelete bir hareket kazandırılarak hareket eden bir referans model elde edilmiştir. Daha sonra poz tahminleme yöntemi ile gerçek insanın kamera yardımı ile hareketleri takip edilmiştir. Bu tezde, referans model ile gerçek insanın oluşturduğu modelin analizleri sonucunda hareket öğrenimi ve geribildirim ile ilgili çıkarımlarda bulunulması ve hareket iyileştirmeleri için bir temel ortam oluşturulması amaçlanmıştır. Arttırılmış gerçeklik, hareket yakalama ve poz tahminleme teknolojileri kullanarak bu çıkarımların yapılabileceğini gösteren bir uygulamanın zemini hazırladı.

2. METHOD VE YÖNTEM

2.1. BVH Dosyalarının Hazırlanması

Uygulamanın ihtiyacı olan animasyonları gösterebilmek adına animasyonun bir formata kaydetme ihtiyacı doğdu. Böylece kaydedilmiş bir animasyonu kullanıcıya gösterip oynatma şansı elde edildi. Bunun için Vicon Blade programı tarafından daha önce kaydedilmiş insan hareketleri kullanıldı. Bu program, animasyonları kaydettiği gibi daha sonra tekrar oynatabilmek adına bvh formatına aktarım yapabildiği için tercih edildi.

Uygulamada 3 tane insan hareketi kullanıldı. Skuat hareketi, yürüme hareketi ve merdiven çıkma hareketinden oluşan 3 tane animasyonu bvh dosyalarına kaydedilerek kullanıldı. Bu dosyalar bvh uzantısı olarak kaydedildi. Bvh dosyaları metin tabanlı olduğu için kolayca uygulamaya girdi olarak kullanılabilmektedir. Bvh dosyalarını girdi olarak tercih edilmesindeki sebep kolay okunabilir, yazılabilir ve en çok tercih edilen animasyon formatı olduğu için tercih edildi. Bvh dosyası metin formatında olduğundan dolayı çok az yer kaplamakta ve normal bir metin editörü ile açılıp kolayca düzenlenebilmektedir. Birden fazla animasyon seçilmesinin sebebi animasyon çeşitliliğini arttırmak, yapılacak uygulama için seçilebilecek birden fazla animasyonun olduğunu gösterebilmektir. Animasyonların türlerinin birbirinde farklı olmasına dikkat edildi. Böylece uygulamanın sadece bir tip animasyonda değil birden fazla ve farklı türlerdeki animasyonlarda da çalışabildiğinin test edilmesi amaçlandı.

2.2. BVH Dosyalarının Girdi Olarak Kullanılması

Kaydedilen bvh dosyalarının animasyon olarak tekrardan uygulamada gösterebilmek adına işlenmesi gerekti. Bu işleme süreci bvh dosyasını alıp, gerekli parçalara ayırıp, işleyip animasyona çevirme görevini yerine getirmesi amaçlandı. Bunun için önce C# programlama dilinde bvh parser yazıldı. C# dilinin tercih edilmesindeki sebep Unity oyun motorunun desteklediği tek dil olduğu içindir. Bvh dosyasında alanların tek tek parse edilip animasyon için anlamlı ve bellekte az yer kaplayacak bir biçimde yazılması amaçlandı.

Tüm bvh dosyaları için genel bir parser yazılmaya dikkat edildi. Bunun sebebi bir bvh uzantılı dosya girdi olarak geldiğinde ek bir çalışma yapmadan uygulamanın bunu animasyona zahmetsiz bir şekilde animasyona çevirmesi amaçlandı.

2.3. Unity Oyun Motorunun Seçilmesi

Tez çalışması Unity oyun motoru kullanılarak geliştirilmiştir. Unity oyun motorunun seçilmesinin nedeni AR uygulamaları için gelişmiş bir ortam sunmasıdır. Unity oyun motoru üzerinde daha tecrübeli olunduğu için seçilmiştir. Kolayca mobil uygulama olarak telefonlara çıktı alınabilmektedir.

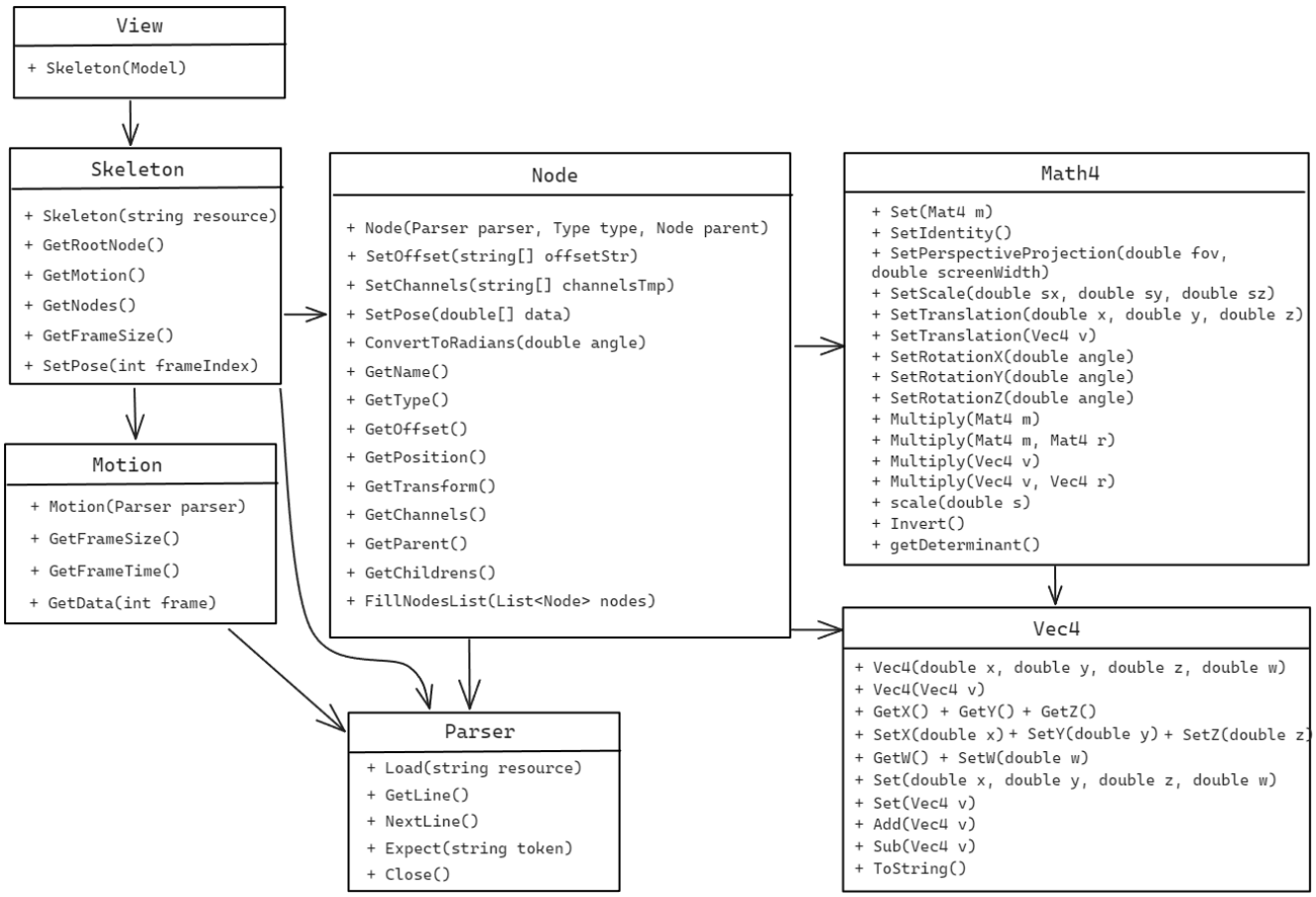
AR uygulamaları için geniş bir topluluğu mevcuttur ve sorunların çözümü için geniş bir bilgi tabanına sahiptir. Tez süresi içerisinde AR ortamına özgü teknik problemlerle ile uğraşmak yerine tez özelinde yapılan çalışmanın gerektirdiği özel zorluklara odaklanmak için kendini kanıtlamış bir oyun motoru olması tercih sebebinde önemli bir etmendir.

İlk çalışmada tez çalışmasında oyun motoru yerine swift dili kullanılmıştır. Swift dilinin tercih edilmesinde sebep Iphone telefona kolayca uygulama geliştirebilmek için seçilmiştir. Iphone telefonun tercih edilmesindeki sebep taşınabilir bir kamerası olduğu için ve AR uygulamaları geliştirilebildiği içindir. Iphone telefonuna kolayca uygulama yapabilmek için Mac book bilgisayar üzerinden swift ile Xcode programı kullanarak geliştirme yapıldı. Fakat swift ile AR uygulaması geliştirme tecrübe eksikliğinden dolayı bvh parser oluşturmada güçlükler çekildi. Aynı zamanda hareket animasyonu oluşturma da teknik güçlükler çekildi. Swift kendi içerisinde kapalı bir animasyon döngüsü barındırdığı için dışarıdan müdahale edilip kendi animasyon döngümüz yaratılamadı.

Bunun üzerine Unity oyun motoru tercih edilmiştir. Unity oyun motoru birçok AR geliştirme aracına sahip olması teknik zorlukların üstesinden gelmek için kolaylaştırıcı bir etkili olmuştur. Aynı zamanda kolayca tüm telefon modellerine çıktı verebilmektedir. Unity de oluşturulan çalışma, çıktı olarak swift koduna oradan da yerel mobil makine koduna çevrim yapmaktadır. Böylece hiçbir performans kaybı yaşanmamaktadır.

2.4. BVH Dosyası Derleyici Yazılımı

Tüm bvh dosyalarında çalışacak bir parser yazmak için bvh dosyasının bölümleri çeşitli metotlara ve sınıflara ayrıldı. Sınıflara ayrılmasındaki sebep modüler olmasıdır. Modülerlik sayesinde kolay geliştirme yapılabilir hale gelmiştir. Ayrıca kodun okunabilirliği artmıştır. Metotlara ayrılmasında sebep, kod tekrarlarının önüne geçilerek her bir metoda bir işlevin yerine getirilmesi amaçlanmıştır. Yazılım mühendisliğindeki tek sorumluluk prensibi uygulanmıştır. Sınıfı diyagram Şekil 2-1 deki gibidir.



Şekil 2-1 Bvh Derleyici Sınıf Diyagramı

Bu sınıfların işleyişleri ve neler yaptığı ile ilgili detaylı bilgiler şunlardır:

`View` sınıfı, animasyon modelinin seçildiği ana sınıftır. Hangi animasyonun oynatılacağı burada karar verilir. `View` sınıfı, `Skeleton` sınıfını referans olarak kullanır. Bu referans Bvh derleyici yazılımını çalıştırır ve `View` sınıfına gerekli bilgileri döner. `View` sınıfında ayrıca Unity oyun motoru ile ilgili konfigürasyon ayarları yapılır. Bu ayarlar detaylı olarak bir sonraki bölümlerde incelenecektir. `View` sınıfı ayrıca Unity oyun motoru ile konuşarak animasyonun gerçekleştirilmesi için gerekli kodları içerir.

`Skeleton` sınıfı bvh derleyici yazılımın ilk giriş noktasını oluşturur. `Motion`, `Parser` ve `Node` sınıflarının referanslarını içerir. Bu sınıflarla konuşarak bilgileri toplar ve onları konfigürasyonunu gerçekleştirir. Bu sınıfın ana amacı iskeleti oluşturmaktır. Tüm iskelet bilgisini elinde bulundurur. İskelet ağacını çıkarılması bu sınıf aracılığı ile meydana gelir. Böylece tek bir yerden tüm karelerdeki iskelet duruşunun bilgileri öğrenilmiş olur. Metotları şunlardır:

`GetRootNode`: Bvh dan root node bilgisini geri döner.

`GetMotion`: `Motion` sınıf bilgisini döner.

`GetNodes`: Bvh' dah yüklenen tüm node bilgilerini liste halinde döner.

`GetFrameSize`: `Motion` sınıfının FrameSize metodunu çağırır.

`SetPose`: Girdi olarak frame numarasını alır ve girilen frame numarasına göre model pozisyon alır.

`Motion` sınıfı animasyonun oluşturulması için gereken bilgileri toplayan sınıftır. `Parser` sınıfı ile ilişki içerisindedir. `Parser` sınıfı sayesinde motion datasını kendi içinde toplar ve hafızaya kaydeder. Bvh da parse edilen motion datası bu sınıf içeresinde anlam bulmaktadır. İlgili metotlarına bakarsak:

`GetFrameSize`: Frame Size bilgisini döndürür. Bu bilgi `Skeloton` sınıfı tarafından çağırılmaktadır.

`GetFrameTime`: Frame Time bilgisini döndürür.

`GetData`: Frame numarası alır. Girilen frame numarasına ait verileri geri döner. Bu veriler koordinat verilerinden oluşur.

`Parser` sınıfı bvh dosyasının hafızaya alınıp her satırının parse edilip hafızada ilgili değerlere atandığı yerdir. Sınıfların metotları kendini açıklamaktadır. Yazılımdaki başka sınıfılar referansını içermez. `Skeleton`, `Motion` ve `Node` sınıfları bu sınıfın referansını kullanarak hafızaya yerleştirdiği bilgilere erişebilir. Metotları şunlardır:

`Load`: Bvh dosyasının dosya sistemindeki konumu verilir. Bu konumu bulan metot bvh dosyasının tamamını hafızaya yükler.

`GetLine`: Bvh dosyasının hafızaya alınmış kısmından bir satır döndürmek için kullanılır.

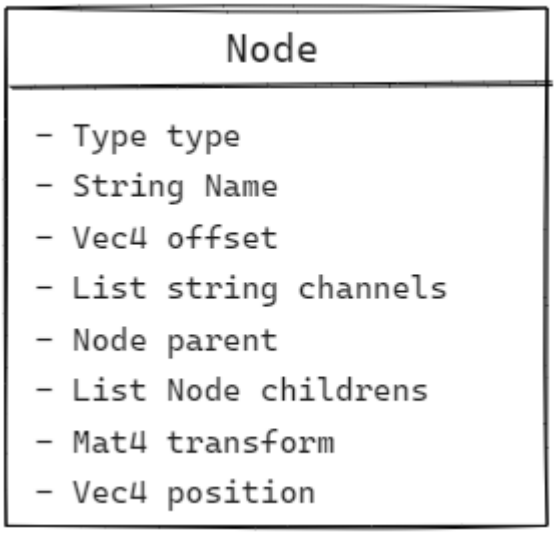
`NextLine`: Bu metot hafızaya alınan dosyanın her bir satırı hafızaya alınır ve her bir satır numaralandırılır. Daha sonra kolay erişebilmek için bu numaralar verilir. Bu metot çağırılan her bir anda bir devam eden bir satır hafızaya alınır ve numaralandırılır. Böylece satırlar ve numaralardan oluşmuş bir liste elde edilir.

`Expect`: Bu metot satırların içindeki boşluklar ile ayrılmış tüm kelime, numara, ilgili başlıkları ayırır ve bunları liste halinde döner. Bu ilgili satırdaki bilgilere ulaşmak için önemlidir.

`Close`: Her dosya okunduktan sonra kapanmak zorundadır. Bu metot ilgili dosyayı kapatmak için gerekli yordamları içerir. Böylece program herhangi bir hataya düşmez.

`Node` sınıfı önemli bir sınıftır. Bvh dosyası içerisinde tanımlanan her bir iskelet noktasını anlamlandırarak ilgili kategorilere ayrılmasını ve bunların bir anlam içerisinde tutulmasını sağlar. Her bir `Node` sınıfı hafızaya alınır. `Skeleton` sınıfı bu `Node` sınıflarına ulaşarak iskelet ağacını oluşturacaktır. Her bir nodun kendine özgü anlamları

vardır. Bu anlamları yaratırken `Vec4` ve `Mat4` sınıflarının referanslarını içererek oluşturulan matematiksel kütüphaneden yardım alır. Ayrıca `Parser` sınıfını kullanarak ilgili bilgiler bu sınıftan temin eder. Bu sınıfın içerdiği metotlar bu sınıfın anlaşılmasını kolaylaştırmaktadır. `Node` veri yapısı Şekil 2.2. deki gibidir. Her nodun tipi, isim, ofset bilgisi, kanal bilgisi listesi, ebeveyn bilgileri, bağlı bulunduğu çocuk nodelarının listesi, transformasyon matrisi ve pozisyon vektörü bulunur.



Şekil 2-2 `Node` veri yapısı

`Node yapıcı metodu`: `Parser`, `Type`, ve `nodeParent` bilgisini girdi olarak alır. Bu metot node veri yapısını oluşturur. Node tiplerini Root, Joint ve End olarak üç kategoriye ayırarak segment haline getirir. Her bir nodun tipini kaydeder. Ayrıca `SetOffcet` metodunu kullanarak ofset bilgilerini kaydeder. Eğer varsa `setChanneels` metodunu kullanarak kanal bilgilerini kaydeder. Noda bağlı childrен yani çocuk node var ise bunları da bağlı bulunduğu parent yani ebeveyn noda bağlayarak bir tree oluşturmasına izin verir. Böylece yukarıdan aşağıya parent-childrен ağacı oluşturulmuş olur. Bu metot recursive(yinelemeli) olarak çalıştırılır. Böylece hem performans olarak ve yazım kolaylığı olarak avantaj kazandırılmıştır. Bu metot aracılığı ile hem ağaç hem de veri yapısı oluşturularak node lar anlamlandırılarak hafızaya kaydedilir.

`SetOffset`: Ofset bilgisi x,y,z,w bilgisi olarak kaydeder.

`SetChannel`: Bvh da yer alan kanal bilgilerini kaydeder.

`GetType`: Node bilgisini getirir.

`GetName`: Her bir nodun bvh da yer alan bir ismi vardır. Bu isim geri döndürür.

`GetOffset`: Ofset bilgisini getirir.

`GetPosition`: Pozisyon bilgisini getirir.

`GetTransform`: Transformasyon bilgisini getirir.

`GetChannels`: Eğer çocukları var ise tüm çocuk nodeların kanal bilgilerini getirir.

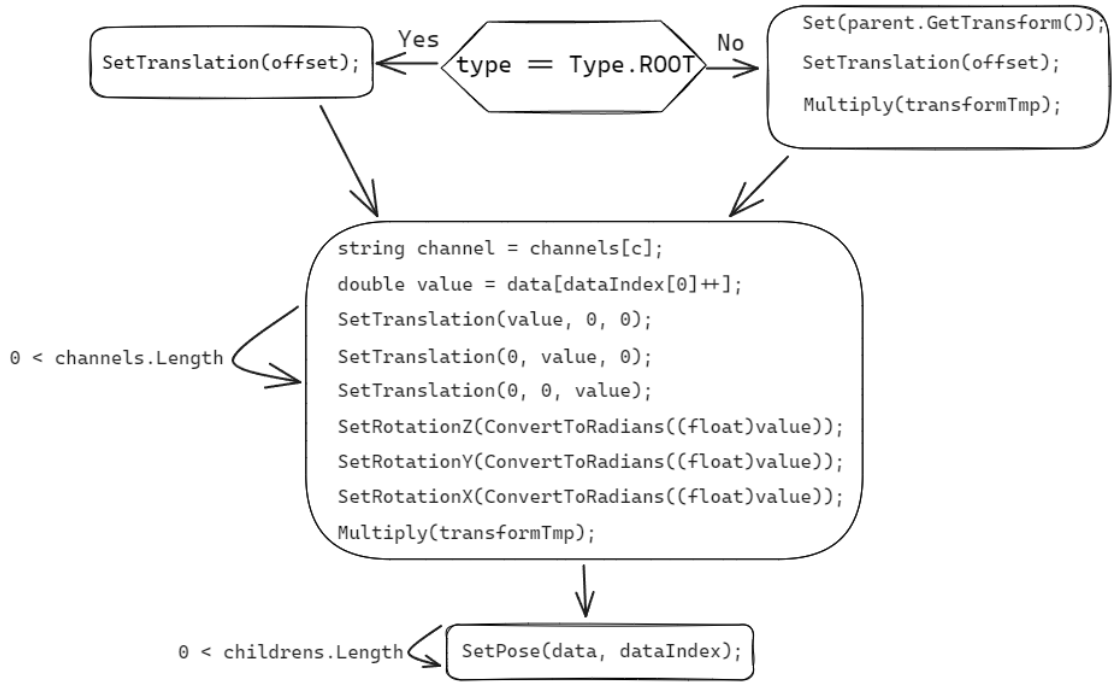
`GetParent`: Nodun bağlı bulunduğu ebeveyn nodeları getirir.

`GetChildrens`: Nodun bağlı olduğu tüm çocuk nodeları liste halinde döner.

`FillNodeList`: Node listesini girdi olarak alır. Bu node listelerini ilgili nodun çocukları olacak şekilde ağaca ekler.

`ConvertToRadians`: Derece olarak belirtilen açıyı radian birimine dönüştürür.

`SetPose`: Poz verisini girdi olarak alır. Bu metoda transformasyon matrisi hesaplanır. Önce nodun tipine karar verilir. Eğer tipi Root ise offset bilgisini girdi olarak alarak `Math4` sınıfının `SetTranslation` metodu çağrılır. Eğer Root değilse Parent nodun Transformasyon bilgileri ile birlikte `Set` metodu çağırılır. Offset değeri kaydedilir ve `Multiply` methodu çağrılır. Bvh da yer alan motion datasından gelen xposition, yposition, zposition, zrotation, yrotation, xrotation verisini alarak transformasyon matrisine dönüştürür. Bunu poz verisi olarak kaydeder. Noda bağlı tüm çocuk nodeların poz bilgilerini de aynı şekilde aynı methodu recursive olarak çalıştırarak hesaplar. Poz aldırma mantığı Şekil 2.3 de gösterilmiştir.



Şekil 2-3 `SetPose` methodu sözde kod gösterim diyagramı

`Mat4h`: Bu sınıf matris hesaplamaları için yaratılmış bir kütüphanedir. Matematiksel işlemler yapılır. Her bir matematiksel işlem metotlara ayrılmıştır. Şekil 2.4 de görülen dörde dört transformasyon matrisini hafızasında tutar. Metotları aşağıdaki gibidir.

$$\begin{bmatrix} m00 & m01 & m02 & m03 \\ m10 & m11 & m12 & m13 \\ m20 & m21 & m22 & m23 \\ m30 & m31 & m32 & m33 \end{bmatrix}$$

Şekil 2-4 `Math4` sınıfında kullanılan Transformasyon matrisi

`Set`: Girdi olarak alınan matrisi belleğe kaydetmek için kullanılır.

`SetIdentity`: Hafızaya aldığı matrisi identitiy matrisi olarak kaydeder.

`SetPerspectiveProjection`: Hafızadaki transformasyon matrisinin perspektif projeksiyona ayarlanması için kullanılır. Böylece uzaktaki objeler daha küçük yakındaki objeler daha büyük olur.

`SetScale`: Ölçek değerini girdi olarak alır. Girilen ölçek değerine göre scale matrisi hesaplar ölçeklendirme yapar. Sonuç girilen x,y,z değerleri için Şekil 2.5 deki gibidir.

$$\begin{bmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 2-5 Scale transformasyon matrisi

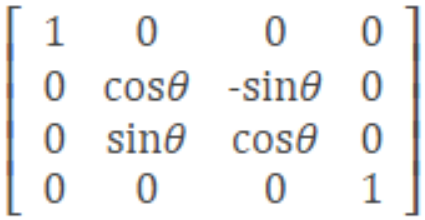
`SetTranslation(x,y,z)`: x,y,z kartezyen koordinatlarını girdi olarak alır. Bu koordinatlar boyunca yer değiştirme yapmak için kullanılır. Sonuç girilen x,y,z değerleri için Şekil 2.6 deki gibi olur.

`SetTranslation(Vec4)`: Vektör olarak girdi kabul eder. Yer değiştirme yapmak için vektör koordinatlarını kullanır. Sonuç şekil girilen vektördeki x,y,z değerleri için Şekil.2.6 deki gibi olur.

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

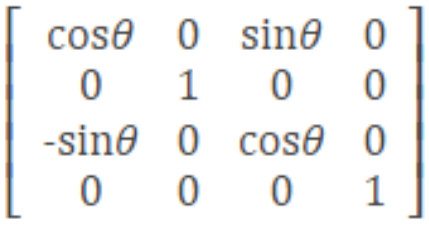
Şekil 2-6 Yer değiştirme transformasyon matrisi

`SetRotationX`: Açı bilgisi alır. Ve X koordinatı boyunca döndürme yapmak için rotasyon matrisini hesaplar. Sonuç Şekil 2.7 de ki gibidir.



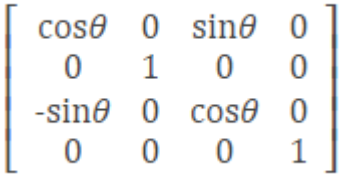
Şekil 2-7 X koordinatı boyunca rotasyon transformasyon matrisi

`SetRotationY`: Açı bilgisi alır. Ve Y koordinatı boyunca döndürme yapmak için rotasyon matrisini hesaplar. Sonuç Şekil 2.8 de ki gibidir.



Şekil 2-8 Y koordinatı boyunca rotasyon transformasyon matrisi

`SetRotationZ`: Açı bilgisi alır. Ve Z koordinatı boyunca döndürme yapmak için rotasyon matrisini hesaplar. Sonuç Şekil 2.9 de ki gibidir.



Şekil 2-9 Z koordinatı boyunca rotasyon transformasyon matrisi

`Multiply(Mat4 m)`: Girdi olarak 4x4 matris alır. Ve bu matrisi hafızadaki asıl matris ile matris çarpımı yapar ve hafızadaki matrise kaydeder.

`Multiply(Mat4 m, Mat4 r)`: Girdi olarak 4x4 matris alır. Ve bu matrisi hafızadaki asıl matris ile matris çarpımı yapar ve girdi olarak aldığı ikinci matrise kaydeder.

`Multiply(Vec4 v)`: Girdi olarak vektör koordinatları alır. Ve bu vektörü asıl matris ile vektör çarpımı yapar ve vektöre kaydeder.

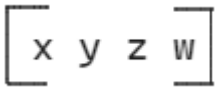
`Multiply(Vec4 v, Vec4 r)`: Girdi olarak vektör koordinatları alır. Ve bu vektörü asıl matris ile vektör çarpımı yapar ve girdi olarak girilen ikinci vektöre kaydeder.

`scale`: Boyutlandırmak için kullanılır. Girdi olarak nümerik bir değer alır. Bu değeri hafızadaki matris ile çarpar ve kaydeder.

`Invert`: Hafızada tutulan matrisin tersini alır.

`GetDeterminant`: Hafızada tutulan determinantını alır.

Şekil 2.10 deki gösterildiği gibi `Vektör` sınıfı vektörel işlem yapmak için hafızada x,y,z kartezyen koordinatlarını ve w değerini tutar. Bu koordinat sistemine homojen koordinatlar denir. Ve bu koordinatlarla her türlü işlemi barındıran matematiksel fonksiyon kütüphanesini içerir. Bu sınıfın referansını `Node` ve `Math4` sınıfları kullanır. Vektörel verileri tutan temel bir veri yapısı oluşturmak için kullanılır.



Şekil 2-10 `Vektör` sınıfında kullanılan homojen koordinat vektörü.

Üç boyutlu bilgisayar grafiklerinde homojen koordinatlar, genellikle (x,y,z,w) şeklinde ifade edilir ve çeşitli geometrik dönüşümleri daha basit ve etkili bir şekilde gerçekleştirmek için kullanılır. Bu koordinat sisteminde, w değeri özellikle önemlidir ve bazı önemli görevleri yerine getirir.

Perspektif projeksiyon, homojen koordinatlar, üç boyutlu nesnelerin iki boyutlu ekranlara perspektif projeksiyonunu kolaylaştırır. Bu projeksiyon sırasında, w değeri perspektif etkisini sağlamak için kullanılır. w değeri nesnenin görüntüleme kamerasına olan mesafesine bağlı olarak değişir.

Dönüşümlerin kolaylaştırılması, homojen koordinatlar, dönüşüm matrislerini kullanarak çeşitli geometrik dönüşümleri (yer değiştirme, ölçeklendirme, dönme) basitleştirir. w değeri, bu dönüşümlerin daha etkili bir şekilde uygulanmasını sağlar.

Koordinat dönüşümleri, normalde, bir nesnenin koordinatları (x, y, z) şeklinde ifade edilir. Ancak, w değeri, bu koordinatların daha geniş bir aralıkta dönüştürülmesine olanak tanır. Örneğin, w değeri 1 ise, elde edilen koordinatlar doğrudan üç boyutlu uzaydaki koordinatlara karşılık gelir. Eğer w farklı bir değere sahipse, örneğin 2, o zaman gerçek koordinatlar (x/w, y/w, z/w) şeklinde hesaplanır.

Ölçeklendirme ve işleme kolaylığı, homojen koordinatlar, grafik işlemciler ve grafik yazılımları tarafından daha kolay işlenebilir hale getirilir. w değeri, bu işlemlerin daha verimli bir şekilde yapılmasına yardımcı olur. Özetle, w değeri, üç boyutlu grafiklerin işlenmesi ve dönüştürülmesi sırasında esneklik ve etkinlik sağlar

Aşağıdaki metotları bulunur:

`Vec4(double x, double y, double z, double w) yapıcı methodu:` x,y,z,w değerlerini alır ve hafızaya kaydeder.

`Vec4(Vec4 v) yapıcı methodu:` x,y,z,w değerlerini `Vektör` sınıfı tipinde alır ve hafızaya kaydeder.

`GetX`: x koordinatını döner.

`SetX`: x koordinatını hafızaya kaydeder.

`GetY`: y koordinatını döner.

`SetY` y koordinatını hafızaya kaydeder.

`GetZ`: z koordinatını döner.

`SetZ`: z koordinatını hafızaya kaydeder.

`GetW`: w değerini döner.

`SetW`: w değerini hafızaya kaydeder.

`Set(double x, double y, double z, double w)`**:** x,y,z,w değerlini alır ve hafızadaki değerleri değiştirir.

`Set(Vec4 v)`**:** x,y,z,w değerlini `Vec4` sınıfı formatında alır ve hafızadaki değerleri değiştirir.

`Add(Vec4 v)`**:** Hafızadaki vektör ile girdi olan vektörü toplar hafızaya yeni vektör olarak kaydeder.

`Sub(Vec4 v)`**:** Hafızadaki vektör ile girdi olan vektörünü çıkarır hafızaya yeni vektör olarak kaydeder.

`ToString()`**:** Bilgi amacıyla hafızadaki vektörü metin olarak kullanıcıya gösterir.

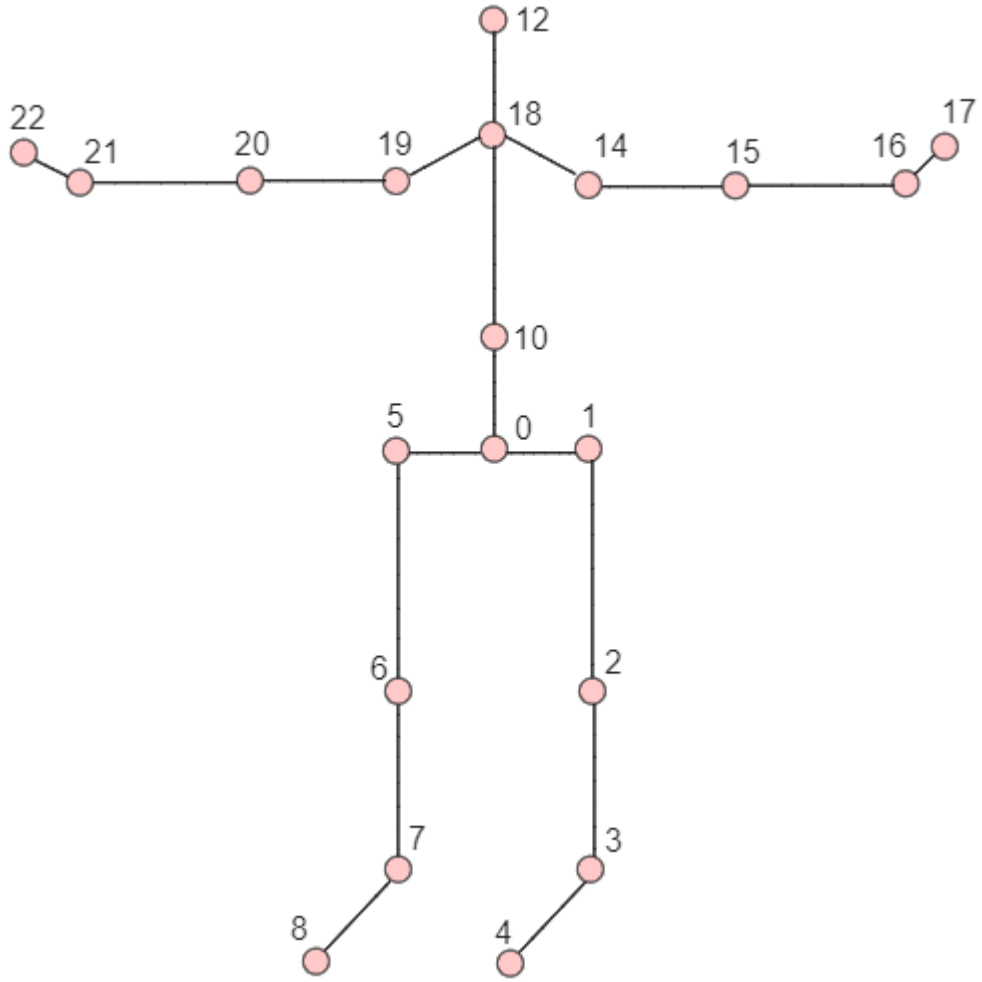
2.5. Unity 3D İskeleti Oluşturma

İnsan iskeletine benzer bir yapıyı oluşturmak için Unity oyun motorunun 3 boyutlu objelerinden yararlanıldı. İnsan iskeletindeki kemikleri gösterebilmek için Unity 'de ki silindir objesinden yararlanıldı. Kemiklerin birbirleriyle bağlantı noktasını gösterebilmek için Unity' nin küre objesi kullanıldı. Parse edilmiş bvh dosyasında hafızaya alınan, transformasyon ve rotasyon bilgilerine göre insan iskeleti oluşturuldu. Unity silindir objesi yaratmak için 3 vektöre ihtiyaç duymaktadır. Transformasyon, scale ve rotation vektörleri. Bu vektörleri doğru hesaplayabilmek için `View` sınıfı `CreateCylinderBetweenPoints` ve `CreateSpherePoint` isminde iki metot yazıldı. Bu metotlar detayları ile aşağıda verilmiştir.

`CreateCylinderBetweenPoints(Vector3 start, Vector3 end, float width, int parent_number, int childer_number, string name)`**:** Bvh Parser ile elde etmiş olduğumuz vektörler node ların başlangıç ve bitiş koordinatlarını içerdiğinden dolayı Unity objesi için dönüştürücü bir metoda ihtiyaç doğmuştur. Girdi olarak bu koordinatlar ve kalınlık bilgisi alır. Ayrıca şekil 2.11 de gösterildiği gibi iskelet numaralandırmalarını girdi olarak almaktadır. Parent_number ve çocuklarına ait numaralandırmalar yer almaktadır. Bu metot girilen kartezyen koordinatlarından, transformasyon, rotasyon ve scale vektörlerini hesaplayıp ilgili Unity

objesine kaydeder. Ayrıca Unity her objenin bir ismi olduğundan dolayı bu isim de objeye girilmiş olur.

`CreateSpherePoints(Vector3 position, int parent_number ):` Küre yaratmak için de birleşim noktalarını vektör şeklinde girdi olarak alır. Pozisyon olarak kürenin orta noktası referans alınır. Parent_number birleşim noktası olarak kullanılacak parent node'nun numarası girdi olarak girilir. Şekil 2.11

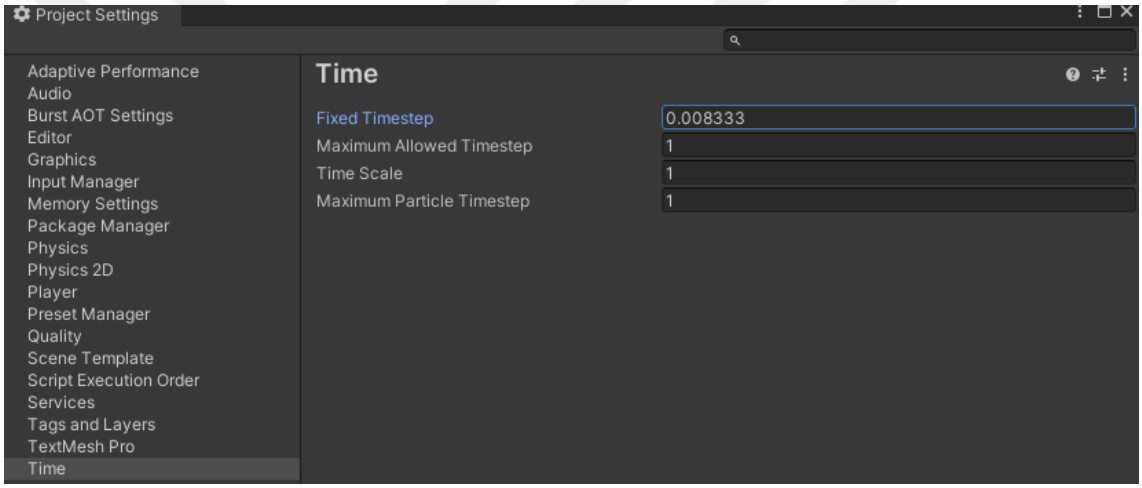


Şekil 2-11 İskelet yapısı parent node numaraları ile birlikte gösterimi

2.6. Unity 3D İskelet Animasyonunu Oluşturma

Animasyon oluşturma işini `Viev` sınıfı üstlenir. `Wiev` sınıfı animasyon oluşturmak için daha önce belirttiğimiz tüm metot ve sınıflar kullanılır. Ayrıca Unity oyun motorunun sunduğu `FixedUpdate` metodundan yararlanılır. Bir karenin

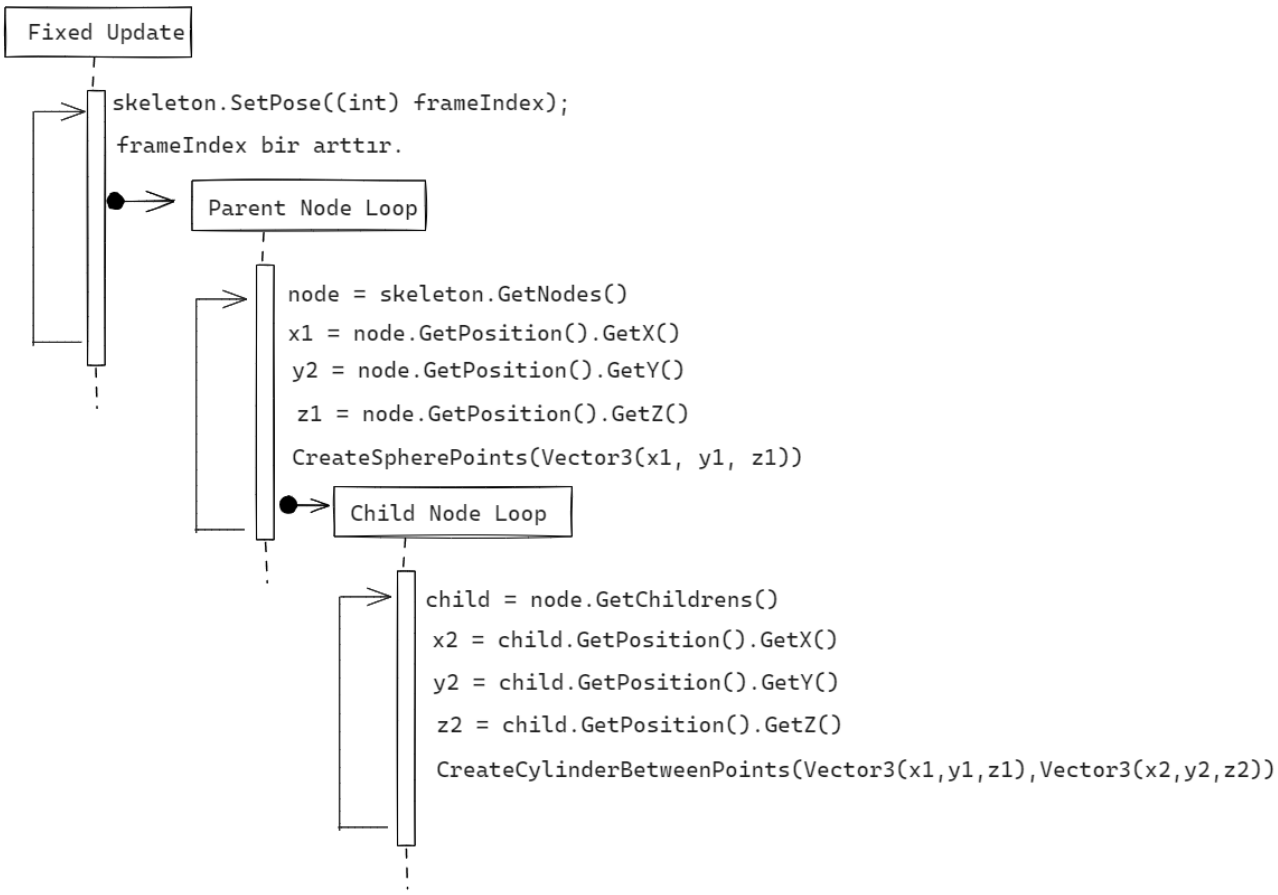
görüntülenme süresi gerçek zamanlı olarak cpu daki işlem gücünden etkilenir. Bu nedenle `Update` yerine `FixedUpdate` methodu kullanıldı. Bunun sebebi animasyonun akıcı oynatılması içindir. Her bir karenin değişken sürede değil, sabit bir sürede oynatılması gerektiğidir. Animasyonun kaç kareden oluştuğu ve her bir karenin kaç saniye sürdüğü bvh dosyasındaki "Frames" ve "Frames Time" alanında yazılıdır. Frames Time alanı bize her bir karenin sabit bir sürede gerçekleştirilmesi gerektiği bilgisini verdiğinden dolayı. Unity oyun motoruna bu değer girilerek animasyonun akıcılığı sağlanmıştır. `FixedUpdate` değeri bvh dosyalarındaki Frame Time kısmında yazan değere göre belirlendi. Bu değer Şekil 2.12 de gösterilen Project Settings kısmındaki Time bölgesinde ayarlandı. Fixed Time Stamp Alanına 0.008333 değeri girildi. Böylece her bir karenin 0.008333 sn de oynatılması sağlandı.



Şekil 2-12 Proje Time Ayarlarının Girildiği Yer

Şekil 2.13 de gösterildiği gibi `Fixed Update` içerisinde frame numarasını her defasında bir arttırarak `Skeleton` sınıfının `SetPose(frame numarası)` metoduna girdi olarak verildi. Böylece o frame ait bir iskelet yapısı oluşturuldu. Bu metot içerisinde bir döngü daha oluşturularak o frame ait iskelet bilgisini `Sceleton` sınıfının `GetNodes()` methodu kullanılarak alındı. O noda ait kartezyen dünya koordinat bilgileri x,y,z olarak `GetPosition()` methodu kullanılarak hafızaya alındı.

Daha sonra bu bilgiler `CreateSpherePoints()` metodunun argümanlarına girdi olarak kullanıldı. Böylece birleşim noktaları çizilmiş oldu. Kemikleri oluşturmak için bu döngünün içerisinde ikinci bir döngü oluşturularak parent noda bağlı tüm çocuk nodelar üzerinde gezinilmesi sağlandı. Çocuk nodelar `GetChildrens()` methodu kullanarak hafızaya alındı. Hafızaya alınan nodelar `GetPosition()` yardımı ile dünya koordinatlarına ulaşıldı ve hafızaya x,y,z olarak kaydedildi. `CreateCylinderBetweenPoints` methodu yardımı kullanılarak ilk döngüden alınan parent koordinatları başlangıç noktası ve ikinci döngü içerisindeki çocuk nodelardan alınan xyz bilgileri bitiş noktası olarak belirlendi. Böylece silindir gösterimine sahip insan iskeleti oluşturulmuş oldu.



Şekil 2-13 `Fixed Update` animasyon oluşturma akış diyagramı

2.7. Unity UI da Animasyon Seçme İşlemi

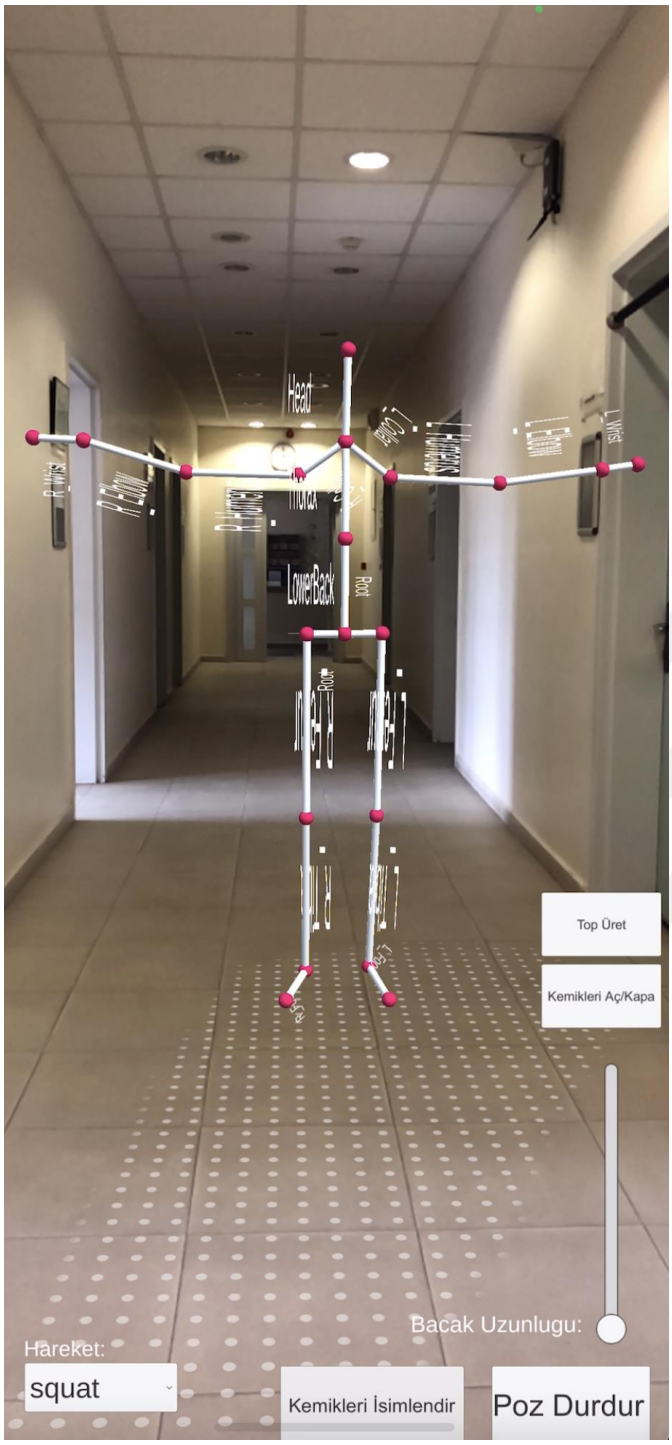
Tez çalışmasında test etmek için 4 tane animasyon kullanılmıştır. Kullanıcıya bu animasyonlar arasında seçim yapmak ve bu animasyonu oynatmak için seçim şansı verilmiştir. Bunun için Unity UI özelliği kullanılmıştır. Şekil 2-14

Unity UI içerisinde dropdown eklenmiştir. Böylece dropdown da açılan menüden animasyon türünü seçtikten sonra kullanıcı tarafından hangi noktaya ekleneceği dokunarak seçilir ve referans model o noktaya hazır bir şekilde eklenir ve oynatmak için sesli komut beklenir.

Şekil 2-14 Animasyon seçim çoklu seçim bölmesi

2.8. Modelde Etiketlerin Görüntülenmesi

İskelet modelinde görüntülenebilecek çeşitli etiketler eklenmiştir. Bu etiket türleri, İskeletteki kemiğin ismi, Bvh parser yazılımının vermiş olduğu kemik numarası, kemiklerin kesişim noktalarında bulunan kürelere etiket eklenmesi bunlardan bir kaçıdır. Şekil 2-15 Bu etiket kameranın açısına her zaman kameraya bakacak şekilde dönecek şekilde ayarlanmıştır. Ayrıca kullanıcı bu etiketlerin görüntülenip görüntülenmemesi bir buton aracılığı ile kontrol edebilmektedir.



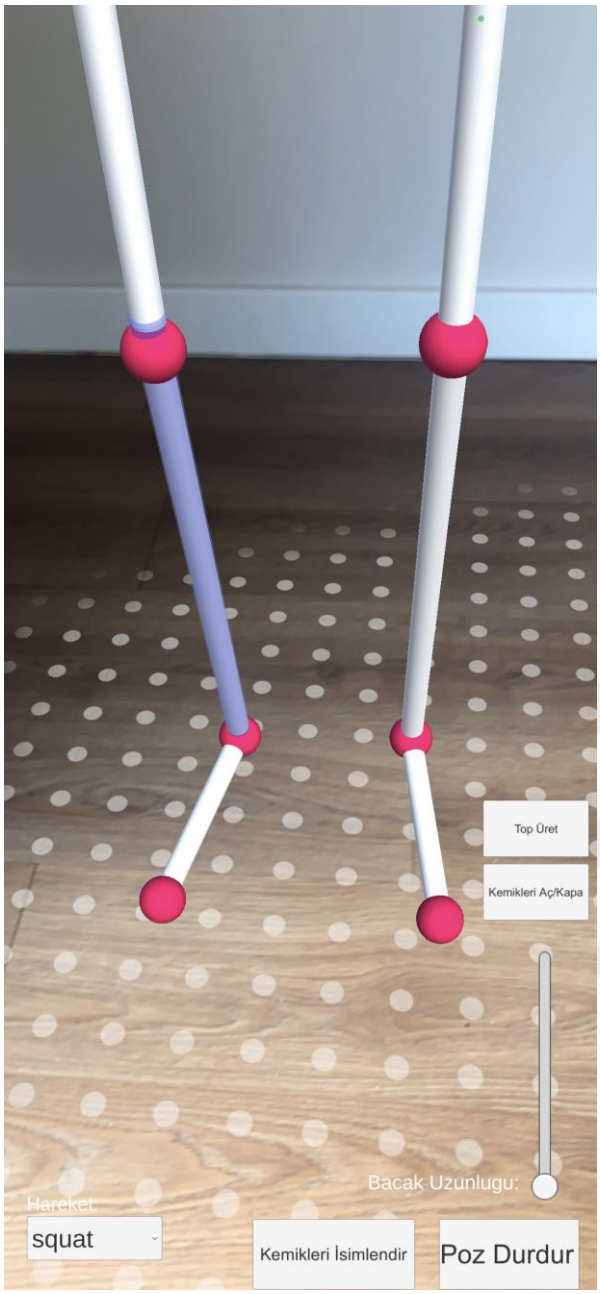
Şekil 2-15 Kemik etiketlerinin görüntülenmesi

2.9. Ortama Küre Şeklinde Bir Cismin Bırakılması

İskelete katı obje özelliği kazandırılmıştır. İskelet katı objesinin diğer katı objeler ile etkileşimini gözlemlemek adına kameranın olduğu yerden bir buton yardımı ile yaklaşık 1 metre öteye bir top bırakılabilmektedir. İskelet yürüdükçe küre şeklinde obje ile yürüyen iskelet modelimiz Unity fizik motoru ile etkileşime girerek küreyi hareket ettirebilmektedir.

2.10. İskelet Modelinde İskeletin Seçilebilmesi ve Etkileşim

İskelet modeli ile kullanıcı etkileşime girebilmektedir. Kullanıcı seçmek istediği kemiğe veya birleşim noktaları olan kürelere dokunarak seçme işlemi uygulayabilmektedir. Seçilen kemik mavi renkte gösterilir. Şekil 2-16 Seçtiği kemiğin boyunu uzatıp kısaltabilmektedir. Bunun için Unity UI kullanılmıştır. Unity de Slider objesi üzerinden kullanıcı seçtiği kemiğin boyunu değiştirebilmektedir. Bu değişim animasyona anlık olarak yansıyacaktır. Böylece gerçek zamanlı olarak değişimi gözlemleyebilecektir. Seçtiği kemik veya kesişim noktalarında bulunan küreler görüntüden kaldırıp tekrar görüntülenebilmesi sağlanabilmektedir. Böylece istenmeyen uzuvların görüntüden kaldırıp istenilen uzuvların kalması sağlanabilmektedir.



Şekil 2-16 Kemik Seçiminin gösterimi

2.11. İskelet Modelinin Boyutunun Ayarlanması

Bvh da parse edilen model dünya, koordinatlarına aktarıldığı zaman orijinal değerlerin çok büyük veya çok küçük olduğu gözlemlenebilmektedir. Bunun için modelin uygun ölçeğe getirilmesi gerekmektedir. Bunun çözümü için 3 farklı yöntemden yararlanılmıştır. En basitten en karmaşığa giden çözümler gerçekleştirilmiştir.

1. Statik bir ölçek değerinin belirlenmesi.

2. Gerçek bir insan boyunun ölçümünün yapılması ve bu boy ile modelin senkronizasyonun sağlanması.

3. Pose estimation kullanarak gerçek insanın boyu hesaplanarak bu boy ile modelin senkronizasyonun sağlanması.

Bu üç yöntemin nasıl sağlandığını aşağıda incelenmiştir.

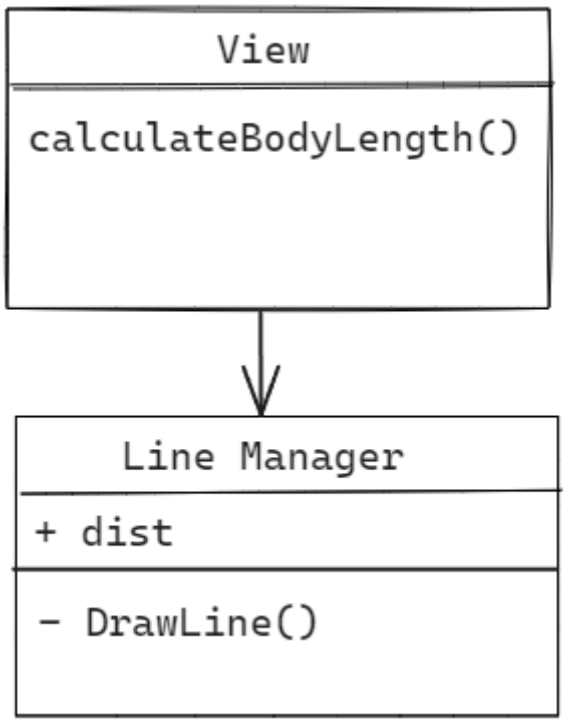
Statik bir ölçek belirlenmesi, deneme yanılma yöntemi modelin yaklaşık 1.68 cm yüksekliğinde olması için gerekli ölçek çarpımı belirlenmiştir. Bu ölçeğin değeri 0.025 olarak bulunmuştur. Şekil 2.17 de gösterildiği gibi bu değerle Nodun bulunduğu x,y,z koordinatları çarpılarak dünya koordinatlarında yer alan bir iskelet modeli oluşturulmuştur.

```
scaleFactor = 0.025
x1 = scaleFactor * node.GetPosition().GetX()
y1 = scaleFactor * node.GetPosition().GetY()
z1 = scaleFactor * node.GetPosition().GetZ()
```

Şekil 2-17 Node koordinatlarının ölçeklendirilmesi

Bu yöntemin dezavantajı statik bir değer olduğu için farklı bvh dosyalarından gelen modellerin koordinatları üzerinde uygulandığında aynı boy oranı çıkmayacaktır. Her model için ayrı bir statik değer bulunması gerektiğidir. Bunun yanında bu statik değer deneme yanılma ile bulunduğu için istenen uzunluğa ölçeklendirmek istendiğinde tekrar bir ölçek değeri bulunması gerektiğidir. Bundan dolayı 2. Yöntemin gerekliliği doğmuştur.

İkinci yöntemde kişi kamera karşısına geçer. Kişiyi referans alacağımız boy hesabı uygulama yardımı ile belirlenir. Bunun için düz zeminde duvara yaslanarak boy ölçüsü alınır. Bu boy ölçüsü, analizi yapılacak referans modelin boy ölçüsüne eşitlenir. Böylece analizin yapılacağı gerçek zamanlı insan modeli ile bvh dosyalarından aktarılan referans modelin ölçekleri senkron edilmiş olur. Bunun için Unity de Şekil 2.18 de gösterilen LineManger isminde bir sınıf tanımlanmıştır.



Şekil 2-18 Line Manager Sınıf Diagramı

Bu sınıf hafızasında dist isminde ölçülen mesafeyi tutar. Sadece bir metotlara sahiptir. Bu sınıfı View sınıfı kullanılır. Böylece hesaplanan değere erişebilir. Hesaplamada kullanılan metotları detaylı inceleyelim:

DrawLine: Kullanıcı kamera karşısına geçen modelin boyunu ölçmek için önce ayak tabanına dokunarak ilk noktayı seçer. Daha sonra başının en üstte noktasına dokunarak ikinci bir noktayı seçer. Uygulama bu iki nokta arasındaki mesafeyi hesaplar. Bu değer dist değerine kaydedilir.

calculateBodyLength: Bu metotta bulunmuş olan yükseklik değeri girdi olarak alınır. Önce bvh dan aktarılmış referans modelin ilk karesindeki boyu hesaplanır. Boy hesaplamada ilk kare kullanılmıştır. İlk kare modelin boy uzunluğunun açık bir şekilde gösterildiği kare olduğu için seçilmiştir. Daha sonra ölçek değerini hesaplamak için şekil 2.19 hesaplama kullanılmıştır:

(dist / bodyLength)

Şekil 2-19 Ölçek değeri hesaplama formülü

Bulunan yükseklik değeri referans modeline bölünerek ölçek değeri hesaplanmıştır. Bu ölçek değeri daha sonra birinci yöntemdeki gibi Nodun bulunduğu x,y,z koordinatları çarpılarak dünya koordinatlarında yer alan bir iskelet modeli oluşturulmuştur.

Bu yöntemin dezavantajlarında biri her seferinde gerçek kişinin boyunun ölçüsünün alınması için yapılacak işlemlerin olması gerektiğidir. İkinci bir dezavantaj ise bu işlemler insan eliyle gerçekleştiği için her zaman hesaplamada hata ile karşılaşmanın yüksek olduğudur. Bundan dolayı işlemleri otomatikleştirmek adına pose estimation yardımı ile kişinin gerçek zamanlı iskelet modeli üzerinden alınan boy ölçüsü ile referans modeli senkron etme yoluna gidildi.

Üçüncü yöntemde kişi kamera karşısına geçtikten sonra Unity UI ile hazırlanmış Sync butonuna basılır. Böylece gerçek kişinin boyu ile referans modelin boyları eşitlenmiş olur. Pose estimation ile otomatik hesaplanan kişinin boyu yine ikinci yöntemde kullanılan şekil 2.19 deki formül ile ölçeklendirme değerine ulaşılır. Dist değeri elle değil bilgisayar yardımı ile hesaplanmıştır. Referans modelin uzunluğunu almak için yine birinci kare seçilmiştir.

Bu yöntemin avantajı işlerin otomatikleştirilmesidir. Daha doğru bir boy hesabı yapılmış olur. Kişi zaten kamera karşına geçeceğinden önceden referans model ile kendi boyu arasında senkronlama işlemini bir sadece buton yardımı ile önceden gerçekleştirmiş olur. Böylece elle boy hesaplama veya statik bir ölçekleme değeri girilmesine gerek kalmaz.

2.12. Pose Estimation Kütüphanesinin Seçimi

Gerçek zamanlı insan hareketinin analizi için işaretleyici kullanılmadan pose estimation tekniğinden faydalanılmıştır. Bunun tercih edilmesinin sebebi herhangi bir rgb kamerası ile insan hareketinin takibi mümkündür. İnsan analizi için işaretleyiciler takmaya veya stüdyolara gerek duyulmamaktadır. Telefon kamerası ile poz tahminleme yapılabilmektedir. Poz tahminleme için Google şirketinin MediaPipe isimli projesi kullanılmıştır. Bu proje Unity oyun motoruna entegre edilmiştir. MediaPipe kütüphanesinin seçilmesindeki temel sebep tüm işletim sistemlerinden çalışabilmesi ve Unity oyun motoruna entegre edilebildiği için seçilmiştir. Tez uygulaması Iphone X model telefonda gerçekleştirildi. Apple Vision pose estimation kütüphanesi Iphone X modelini desteklemediğinden dolayı kullanılamamıştır. OpenCV kütüphanesi AR ve 3D uygulamaları için uygun olmadığından dolayı kullanılmamıştır.

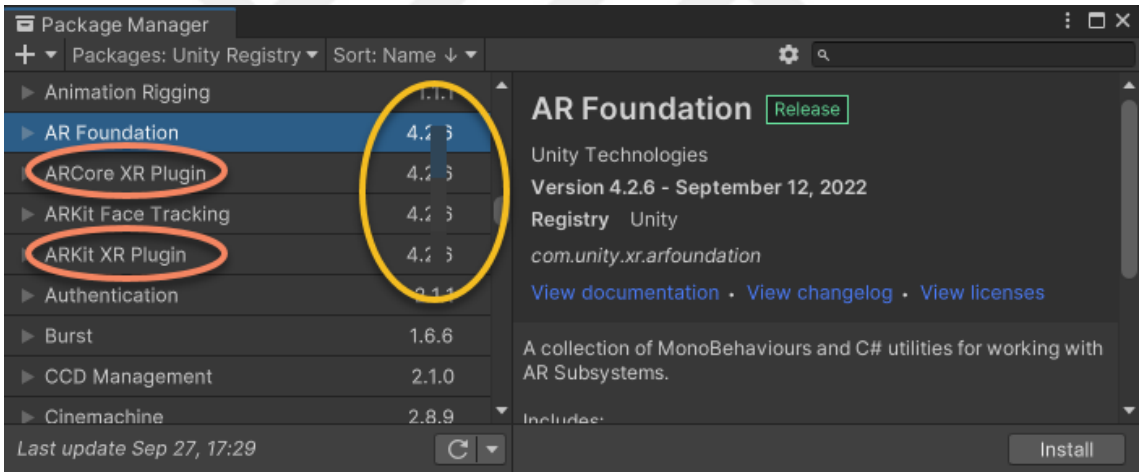
2.13. Pose Estimation İle Kareket Analizi Uygulaması

Bu tezde amaçlanan uygulamanın temel mantığı Pose estimation ile gerçek zamanlı insan hareketi analizi kontrollü ortamda birleştirmek ve bu iki hareketten gerçek zamanlı bir hareket analiz sonucu çıkararak kullanıcıyı bilgilendirmek. Pose estimation için mediapipe kütüphanesinden yararlanarak kullanıcının gerçek zamanlı iskeleti iskeletinin gerçek dünyada oluşturulması sağlandı. Aynı zamanda bvh dosyasından aktarılan referans animasyonda oynatılmaya başlandı. Bu iki insan iskeleti önce sync butonu ile boyları eşitlendi. Daha sonra sürekli oynayan referans animasyonun üstüne gerçek zamanlı insan hareketi bindirildi. Bu iki hareketin oluşturduğu farklar kullanıcının cep telefonundan görüntülendi. Böylece Bu iki insan hareketi modelinin birbirleriyle karşılaştırma şansı elde edilmiş oldu.

2.14. Unity AR Kurulumu ve Konfigürasyonu

Unity de AR uygulamaları çalıştırmak için Unity nin bize sunmuş olduğu AR Foundation altında toplanan AR plugininden yararlanıldı. Bu plugin ios a uygulama yapıldığı için Apple ARKit XR Plug-inidir. Bu pluginin bizzat sağlamış olduğu bazı avantajlar vardır. Bu avantajlardan biri ios projesine AR uygulaması yaptığımızda unity nin kolay bir şekilde ios işletim sistem ile haberleşerek bizim bir çok iletişim protokolünü yerine getirmesidir. Böylece AR dünyası için gerekli hesaplamaların detayına girmeden sadece UI üzerinden konfigüre edebildiğimiz bir yapıya kavuştuk.

Unity de AR uygulaması geliştirebilmek için ilgili Unity paketlerinin önceden yüklenmesi gereklidir. Unity bu paketlere ihtiyaç duyar. Şekil 2.20 de yer alan Unity Package Manager yerinden AR Foundation ve ARCore XR Plugin paketleri Unity'e yüklendi.

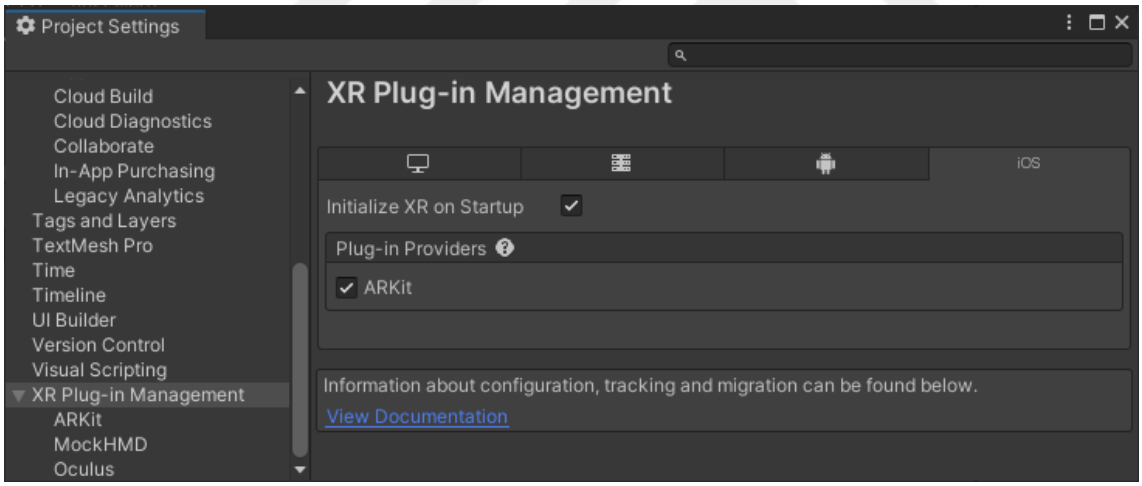


Şekil 2-20 Unity Package Manager

Böylece Unity de AR uygulamaları geliştirebilmek için gerekli araçlar yüklenmiş oldu. Daha Sonra Şekil 2.21 deki Unity proje ayarlarından XR Plugin Management kısmından de XR Plugin Management ortamı kuruldu. Bu ortamın kurulması ile birlikte AR uygulamaları geliştirmek için Pluginlerin yer aldığı seçenekler belirmiş oldu.

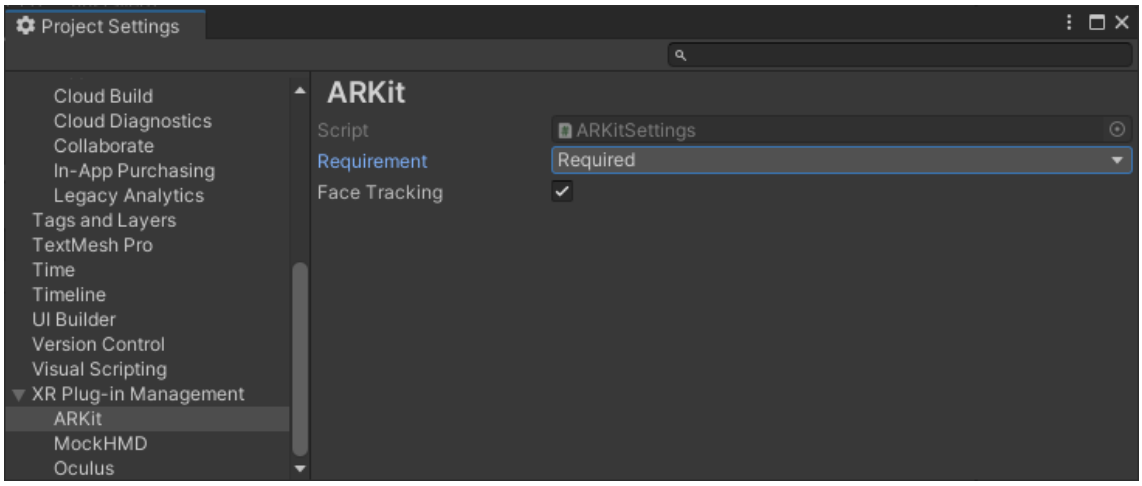
Şekil 2-21 Unity de XR Plugin Manegement ortamının kurulması.

Kurulduktan sonra Şekil 2.22 de yer alan Plug-in Provider kısmında artık AR Pluginlerinin seçilebildiği bir alan çıktı. Uygulama Mac Pro M2 işlemcili diz üstü bilgisayarda gerçekleştirildi. Bundan dolayı XR Plugin Manegement kısmında IOS butonu aktif hale geldi. Bu alan üzerinden ARKit seçildi. ARKit'i seçebilmek için Apple marka bir bilgisayarın olması gereklidir.



Şekil 2-22 ARKit Seçim Kısmı

ARKit seçildikten sonra Şekil 2.23 da gösterilen kısımdan Requirement Required olarak seçildi. Bunun sebebi ARKit kütüphanelerinin uygulama çalışırken gerekli olduğunu belirtmek içindir. Böylece uygulama telefona yüklenirken gerekli kütüphaneleride uygulamanın içerisine gömer ve kullandırır.

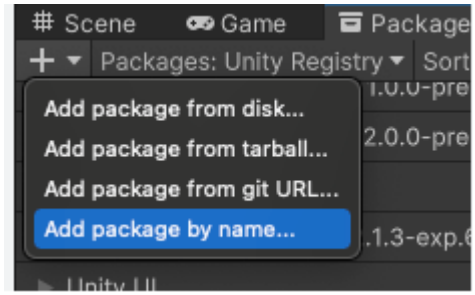


Şekil 2-23 ARKit Ayarları Bölgesi

2.15. Unity Mediapipe Kurulumu ve Konfigürasyonu

Unity de mediapipe kütüphanesini kullanmak için MediaPipeUnityPlugin paketi kullanıldı. Bu paketi yüklemek için Şekil 2.24 de gösterilen Package Manager menüsünden seçim yapıldı ve "MediaPipeUnityPlugin.*.unitypackage" isimli dosya girilerek yüklendi. Böylece Mediapipe kütüphanesi unity ile Entegre hale gelmiş oldu.

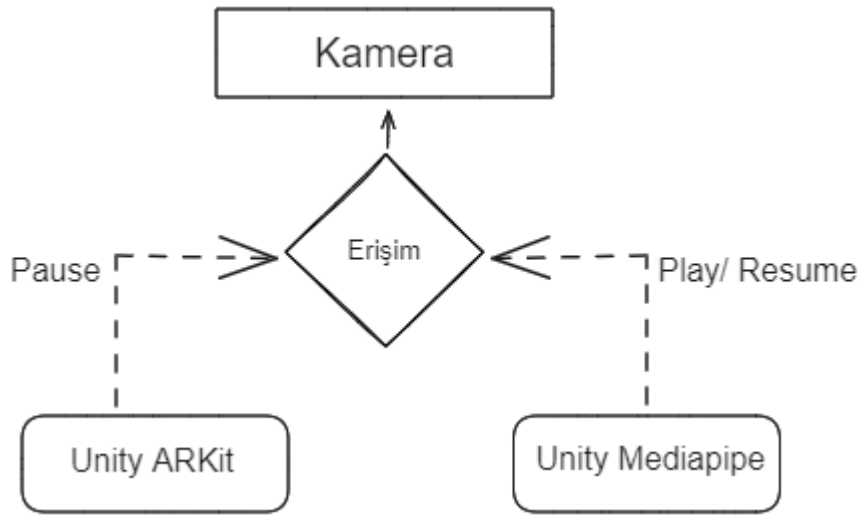
Mobil uygulamalarda bu paketin çalışması için Mediapipe ayarlarından iki tane konfitürasyon yapıldı. Preferable Inference Mode GPU olarak belirlendi. Çünkü bu kütüphane mobil uygulamalarda sadece GPU modunda çalışmaktadır. Ayrıca Asset Loader Type StreamingAssets olarak belirlendi. Bu kütüphane mobil uygulamalarda sadece gerekli dosyaları stream ederek çalışmaktadır.



Şekil 2-24 Unity Paketi yükleme menüsü

2.16. Unity ARKit ile Unity Mediapipe Kamera Erişimleri

Unity AR kütüphanesi ile Mediapipe kütüphanesi aynı kameraya erişim isteğinde bulundu. Cep telefonu kamerası aynı anda sadece bir kütüphanenin kullanımına açılabildi. Unity ARKit ile Mediapipe kütüphaneleri aynı kamerayı kullanma istekleri olduğundan dolayı birim zamanda sadece bir kameradan görüntü alınabildi. Analiz yapabilmek için her iki kameradan gelen görüntüye ihtiyaç olduğundan dolayı bir çözüm geliştirildi. Şekil 2.25 daki gibi bir mantık eklendi. Önce AR kütüphanesinin kullandığı kamera aktif edildi. Poz hesaplamanın kameraya erişimi kapatıldı. Bu kamera ile AR için gereken gerçek ortam bilgileri alınabildi. Böylece bvh dan alınan referans model gerçek dünyaya yerleştirilebildi. Daha sonra gerçek zamanlı pose estimation yapabilmek için mediapipe kütüphanesinin kullanıldığı kamera Play/Resume butonundan aktif edildi. Bu sefer AR kütüphanesin kameraya erişimi kapatıldı. Bu aşamadan sonra Mediapipe kütüphanesi çalıştırılmaya başladı. Bu kameradan poz bilgilerinin alınması ve gerçek dünyaya aktarılması sağlandı. Bu aktarımı durdurup AR kütüphanesinin kullanmış olduğu kameraya geçmek için ise Pause butonu kullanıldı. Bu buton Pose estimation hesaplamalarını durdurup tekrar AR kamerasına geçmek için kullanıldı. Tekrar mediapipe kütüphanesinin kullandığı kameraya geçmek için Play/Resume butonuna basılarak AR kamerası kapatıldı ve Pose Estimation kaldığı yerden devam ettirildi. Bu işlemler sırasında referans animasyonun oynatılmasında veya poz hesaplamada herhangi bir sorun ile karşılaşılmadı. Referans animasyon devam ettiği için her iki kameradan da farklı zamanlarda bilgiler alınabilmiş oldu.

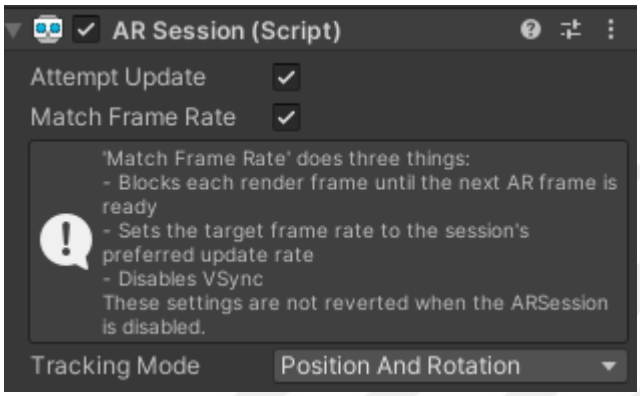


Şekil 2-25 Kamera Erişim Durum Diyagramı

2.17. Unity AR Uygulamada Kullanılan Komponentler

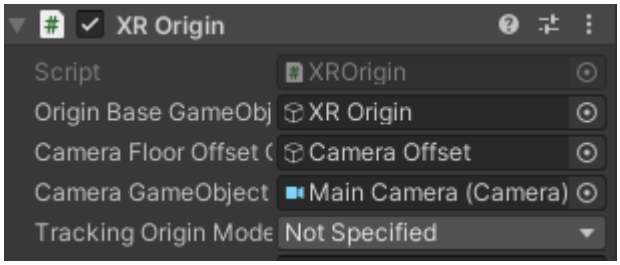
Uygulama Unity versiyon 2021.3.23f1 versiyonunda gerçekleştirildi. Uygulama gerçekleştirildiği anda kullanımda olan en son versiyon seçildi. Uygulamayı gerçekleştirebilmek için çeşitli Unity AR bileşenleri kullanıldı. Bu bileşenler şu şekildedir.

Şekil 2.26 da yer alan AR Session: AR uygulamasının yaşam döngüsünü ve yapılandırma seçeneklerini denetler. Yalnızca bir etkin Session vardır.



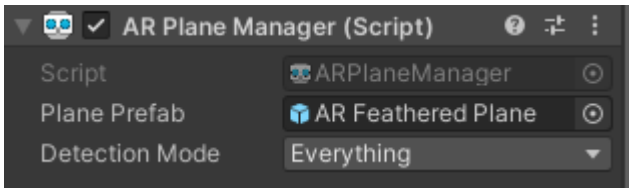
Şekil 2-26 AR Session Komponenti.

XR Origin: AR Kamerasını aktif eden ve kameranın pozisyonunu dünya koordinatları içinde belirlenmesi için kullanıldı. Şekil 2.27



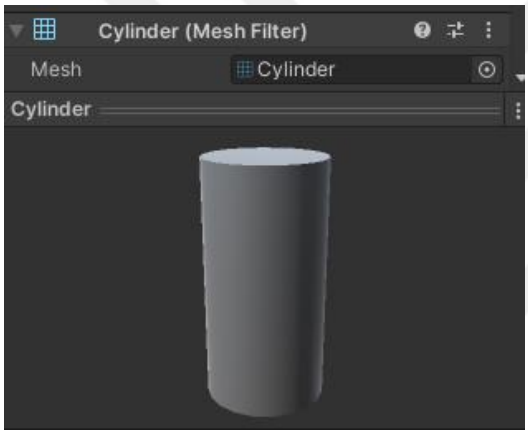
Şekil 2-27 XR Origin Komponenti

Ar Plane Manager: Kamera ile gerçek dünyayı algılamak ve bu dünyanın düşey ve yatay düzlemlerine bir sanal düzlem yerleştirmek için kullanıldı. Tüm objeler bu sanal düzlemin üzerine yerleştirilir. Detection Mode olarak Everything seçildi. Bu seçim tüm eksenlerdeki yüzeyleri taramak için kullanıldı. Şekil 2.28

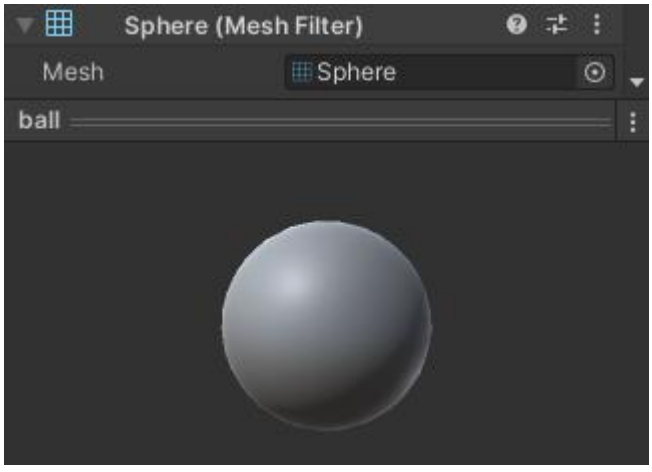


Şekil 2-28 AR Plane Management Komponenti

İskeletleri oluşturmak iki tane komponent kullanıldı. Kemiklerin gösterimi için Unity Şekil 2.29 daki Cylinder objesi kullanıldı. Kesişme noktası olarak Şekil 2.30 deki Sphere komponenti kullanıldı.

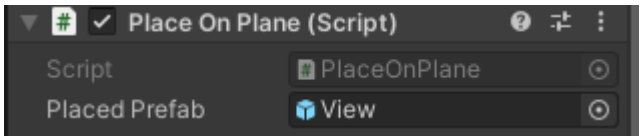


Şekil 2-29 Cylinder Komponenti



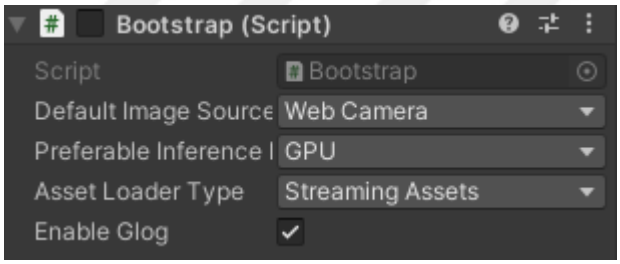
Şekil 2-30 Sphere Komponenti

Place on Plane Komponenti: Bu komponent AR Plane Management komponenti sayesinde çizilen düzlemler üzerine istenilen Unity Objelerini koymak için kullanıldı. Böylece Kullanıcıya dünya üzerinde istediği bir noktaya dokunarak obje ekleme özelliği kazandırıldı. Şekil 2-31 de View objesini ekleme noktası olarak bu komponent kullanıldı.



Şekil 2-31 Place On Plane Komponenti

Mediapipe Bootstrap Komponenti: Bu komponent sayesinde mediapipe pose estimation çalışırken kullanılacak konfigürasyonlar seçildi. Görüntü kaynağı olarak telefonun web kamerası seçildi. Pose Estimation algoritmalarının CPU/GPU da koşması için seçeneklerden GPU seçildi. Böylece GPU da daha hızlı çözümleme yapılabildi. Asset Loader Type olarak Streaming Assets seçildi. Mobil telefonlarda çalışması için bu seçeneğin seçilmesi gerekiyor. Şekil 2-32



Şekil 2-32 Mediapipe Bootstrap Komponenti

Mediapipe Modelleri, bir kişinin 3 boyutlu vücut pozunu tahmin etmek için akıllı telefon kameralarından çekilen gerçek zamanlı görüntüler için optimize edilmiştir. Bu optimazasyon 3 ayrılır. Lite, Heavy ve Full. [36]

Lite: Gpu modunda Yaklasık 49 FPS görüntü sağlayabilen fakat yaklaşık yüzde 87 oranında doğru poz tahminleme yapabilir.

Heavy: Gpu modunda Yaklasık 40 FPS görüntü sağlayabilen fakat yaklaşık yüzde 91.8 oranında doğru poz tahminleme yapabilir.

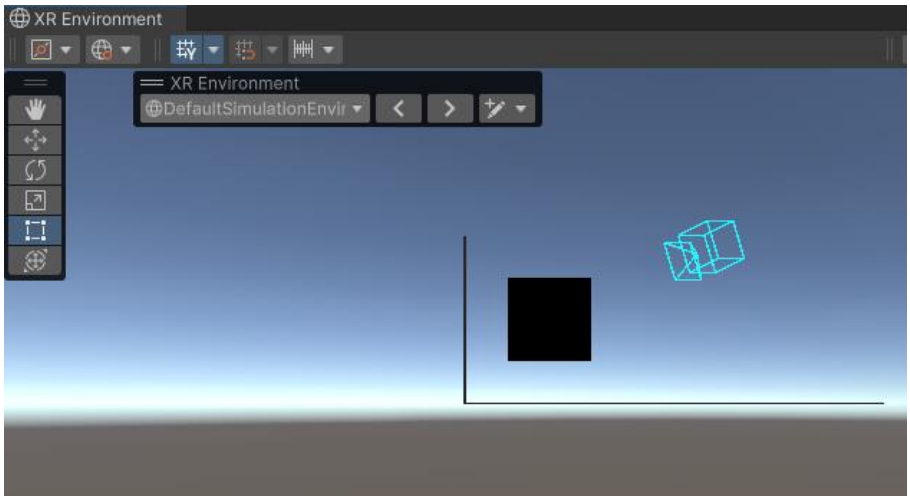
Full: Gpu modunda Yaklasık 19 FPS görüntü sağlayabilen fakat yaklaşık yüzde 94.2 oranında doğru poz tahminleme yapabilir. [36]

Tez çalışmasında hem yüksek FPS de çalışabilmesi için hem de doğruluk oranının yüzde 90'nın üzerinde olmasının yeterli olacağını düşünerek Heavy Seçilmiştir. Bu seçim Pose Tracking Graph komponentinin Şekil 2-33 de gösterilen Model Complexity alanından seçilmiştir.

Şekil 2-33 Model Complexity Seçim Alanı

2.18. Simülasyon Ortamı

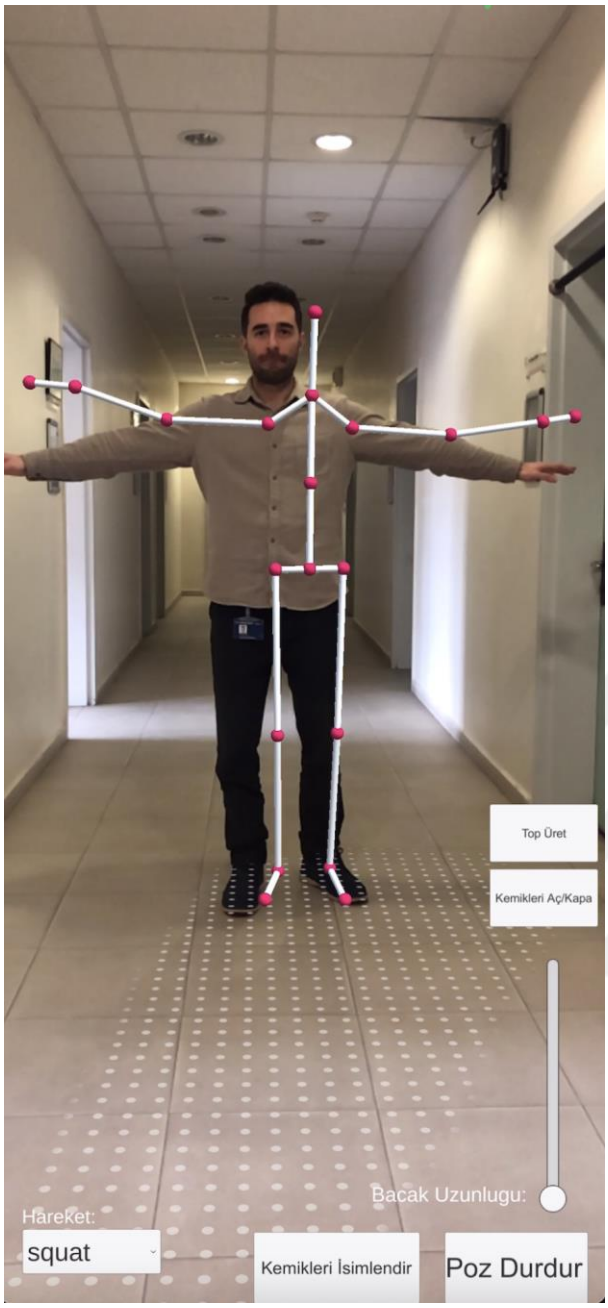
Unity içerisinde yapılan değişikliklerin etkisini görmek için sanal bir simülasyon ortamı kullanıldı. Böylece sürekli telefona aktarıp test etmeye gerek kalmadan bu ortam üzerinde testler gerçekleştirilebildi. Unity'nin sağlamış olduğu bu özellik sayesinde geliştirme süresi düştü ve hatalar azaldı. Anında değişiklikler yansıtılarak Unity oyun motoru ekranından çıkmadan gözlemleme fırsatı oluşturuldu. Fakat bu ortamda yapılan testlerin tek eksikliği kameranın aktif olamamasıdır. Kamera gerekmeyen yerlerde bu test ortamı kullanıldı. Mesela bvh dosyasından parse edilen referans animasyonun doğruluğu ve testi için bu ortam kullanıldı. Pose estimation için kameraya ihtiyaç duyulduğundan dolayı testleri gerçek telefon üzerinden yapılması gerekti. Şekil 2-34 de görüldüğü üzere XR Enivorment olarak DefaultSimulation Environment isminde AR objeleri yerleştirebileceğimiz bir ortam sağlanmış oldu.



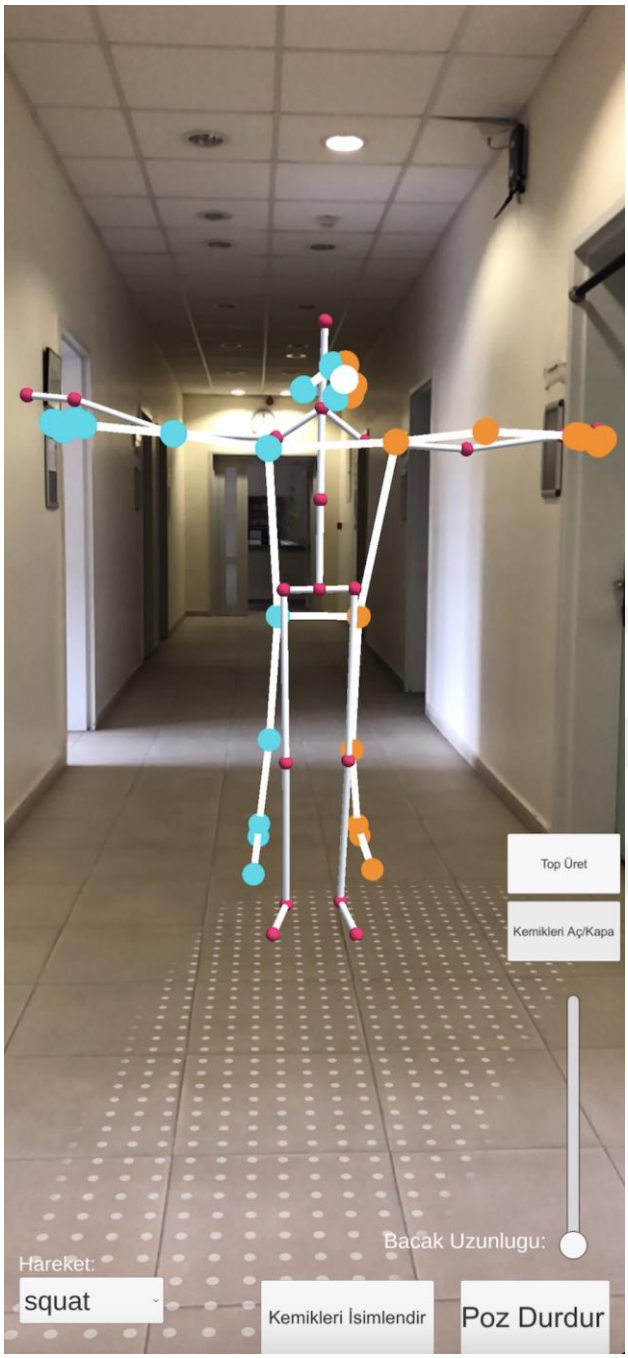
Şekil 2-34 Unity Simülasyon Ortamı

2.19. Tez Uygulaması Örnek Senaryo Uygulaması

Uygulama senaryosu Hacettepe Üniversitesi Spor Bilimleri Fakültesi binasında gerçekleştirildi. Tripoda uygulamanın bulunduğu Apple Iphone X telefonu sabitlendi. Uygulama çalıştırıldı. Uygulama çalıştıktan sonra arka kamera aktif hale geldi. Arka kamera geniş bir boşluğa bakacak şekilde ayarlandı. Bu boşluğun algılanması için önce tripod ile birlikte telefon ortamda gezdirildi. Kamera ortamı algılayarak zemin tanıtıldı. Daha sonra kamera tanıtılan yerin biraz gerisine konumlandırıldı. Böylece kamera daha geniş bir açıyla ortamı görüntülemeye başladı. Sağdaki açılır menüden squat hareketi animasyonu seçildi. Daha sonra ekran üzerinden istenilen noktaya dokunularak dokunulan yere referans animasyon modelinin eklenmesi sağlandı. Sesli komut dinlendi ve hareketi yapacak kişi kamera karşısına geçip kollarının iki yana açıp beklenmesi istendi. Şekil 2-35 Hazır olunca start komutu ile animasyonun başlanması sağlandı. Animasyon sonra erince tekrar animasyonun başına dönerek sürekli hareket etmesi sağlandı. Start komutu ile ile pose estimation hesaplamaları çalışmaya başladı. Program insan bedenini algılayarak pose estimation yardımı ile insan iskeletini referans modelin olduğu yerde oluşturdu. Şekil 2-36 İnsan bedeni hareket ettikçe bu iskelet modelde hareketleri takip etmeye başladı. Otomatik olarak referans animasyon modelinin boyu gerçek insan modelinin boyu ile eşitlendi. Kamera karşısına geçen insan telefonun ekranında gördüğü referans modelin yaptığı hareketin aynısı yapmaya çalıştı ve 6 saniye sonra hareket verileri ftp sunucusuna aktarıldı.



Şekil 2-35 Referans pozisyonunun alınması



Şekil 2-36 Poz Takibi ile referans animasyonun takibi

2.20. Sesli Komutlar ile Yönlendirme

Uygulama içerisinde kullanıcıyı yönlendirmek adına sesli komutlar eklendi. Bu sesli komutlar kullanıcının uygulamayı daha doğru kullanması için oluşturulmuştur. Kullanıcıyı doğru yönlendirebilmek ve uygulamada yaptığı hareketlerin geri bildirimlerini sesli komutlar ile alabilmek için oluşturulmuştur. Bu komutlar için metinden sese dönüştürme teknolojisinden yararlanılmıştır. Erkek sesi ve Türkçe seçilmiştir. Bu sesler; önce telefonunuzda yeri tarayınız, eklemek istediğiniz hareketi seçiniz, kişinin hangi hareketi seçtiğini kaç tekrar yapması gerektiğini ve kameranın karşısına geçip başlangıç pozisyonu alması gerektiğini söyleyen ve başla başla diyerek harekete başlamasını söyleyen sesli komutlardan oluşmaktadır.

Aynı zamanda uygulamayı kullanan kişi tarafından sesli komutlar ile uygulamayı kontrol edebilme özelliği eklenmiştir. Örnek olarak Başla diyerek hem animasyon hareketi hem de poz tahminleme algoritması çalışmaya başlamıştır. Sesli komut olarak Ios işletim sisteminin konuşmadan metne çeviri özelliği Unity oyun motoruna entegre edilmiştir. Daha sonra uygulamanın içerisine eklenmiştir.

2.21. Hareketleri Kayıt altına alma ve Dışarı Aktarma

Harekete başla komutu verildikten sonra uygulama kayıt moduna geçer ve tüm hareketler output.csv isminde bir dosyaya kaydedilir. Kayıt altına alınan veriler arasında hem modelin hareketi hem de poz tahminleme yöntemi ile elde edilen hareket kayıt altına alınır ve dosyaya yazılır. Dosya formatı her bir satır bir frame olacak şekilde kaydedilir. Her bir satırda aralarında virgülle ayrılmış bir formatta Frame Sayısı, kemik numarası, modelin x,y,z koordinatları ve poz tahminlemeden gelen x,y,z koordinatları kaydedilir. Daha sonra bu dosya ilerde kullanılmak üzere filemanager.ai ismindeki ücretsiz bir dosya sunucusuna 6 saniye sonra gönderilir ve sunucuda kaydedilir.

Frame Sayısı	Kemik Numarası	Model X koordinatı	Model Y koordinatı	Model Z koordinatı	Poz Tahminleme X Koordinatı	Poz Tahminleme Y Koordinatı	Poz Tahminleme Z Koordinatı
1	1	0.1	0.4	0.4	0.3	0.1	0.1

Tablo 2-1 output.csv dosyası başlık yapısı ve örnek değerleri

3. TARTIŞMA VE SONUÇLAR

3.1. Unity Seçimi

Tez Çalışmasında Unity oyun motoru tercih edilmiştir. Bunun birkaç sebebi vardır. Bunlardan birincisi kolay bir şekilde ios işletim sistemi platformu için geliştirme yapılmasına izin vermesidir. İkinci kapsamlı AR platformu için sayısız kütüphane ve araç barındırmasıdır. Mediapipe kütüphanesi ile uyumlu çalışabilmektedir. Bu sayede geliştirme süreci hızlanmıştır. Kolay öğrenme eğrisi mevcuttur. Sıfırdan başlanılan bu proje için önceden aşina olunan Unity oyun motoru ile hızlı geliştirme süreci gerçekleşmiştir.

3.2. Unity Oyun Motorunun Sınırlılıkları

Bu tez çalışmasına konu olmayan çeşitli sınırlılıklar mevcuttur. Bunlardan birkaçı şunlardır.

Unity, performansın iyi olması için optimize edilmiştir, ancak büyük ve karmaşık sahnelerde veya çok yüksek kaliteli grafikler ve fizik simülasyonları gerektiren oyunlarda performans sorunları yaşanabilir. Bu durum, optimize etmek ve verimli kod yazmak gerektiği anlamına gelir.

Unity'nin bazı özellikleri ve hizmetleri için ücretli lisans gerekebilir. Büyük ve ticari projeler için bu maliyetler artabilir.

Unity tabanlı oyunlar genellikle büyük uygulama boyutlarına sahiptir, bu da kullanıcılar için uzun indirme süreleri ve daha fazla depolama alanı gerektirebilir. [39]

3.3. ARKit ile AR Uygulamalarının Sınırlılıkları

Kullanılan bu teknolojinin bazı sınırlılıklarından bahsedebiliriz.

AR uygulamaları oluşturmak için donanımın örneğin, AR gözlükleri veya akıllı telefon kameraları kullanılmasına dayanır. Donanımın teknik özellikleri ve yetenekleri, AR uygulamalarının performansını ve işlevselliğini etkileyebilir.

Hareket Takibi Hassasiyeti kullanıcıların hareketlerini ve pozlarını takip etmek için kameralar ve sensörler kullanır. Ancak bu sistemler, kullanıcının hareketlerini tam olarak ve yüksek hassasiyetle takip etmekte sınırlı olabilir, özellikle hızlı veya karmaşık hareketlerde.

AR uygulamaları bazen optik illüzyonlar veya takip hataları gibi görüntü sorunlarına neden olabilir. Örneğin, nesnelerin beklenmedik şekilde kaybolması veya belirsizlik yaratan yansımalar olabilir.

Güç ve Batarya Sorunları: AR uygulamaları, donanımın daha fazla güç tüketmesine neden olabilir ve bu, cihazın pil ömrünü kısaltabilir.

Çevresel Faktörler: Parlaklık, ışık koşulları ve çevresel faktörler, AR uygulamalarının performansını etkileyebilir. Özellikle açık hava kullanımında bu faktörler daha da kritik hale gelebilir. Özellikle zemin tanımlama düşük ışıklarda olumsuz etkilenmektedir.

AR uygulamaları, gerçek dünyayı analiz etmek ve sanal nesneleri yerleştirmek için karmaşık görüntü işleme ve bilgisayar görüşü tekniklerini kullanır. Bu, işlemci ve grafik kartının daha fazla yük taşımasına neden olabilir.

Kullanıcıların hareket kabiliyeti ve fiziksel sınırlamaları, AR uygulamalarının kullanılabilirliğini etkileyebilir. Örneğin, bir kullanıcının belirli bir konuma ulaşması veya belirli bir mesafede durması gerekebilir.

Bu sınırlamalar, AR geliştirme araçları kullanırken dikkate alınmalıdır. Bu sınırlamaları anlamak ve AR deneyimlerini optimize etmek için uygun stratejileri benimsenmelidir. [40]

3.4. Xcode ile Geliştirme Yapılmamasındaki Sebep

Bu tez çalışması için en başta XCode editörü ortamında Swift dili ile MacBook bilgisayarda uygulama geliştirmek için çalışmalara başlanmıştır. Bu dil ve ortam ile ios platformuna uygulama geliştirebilmek mümkündür. Fakat görülmüştür ki, hem swift dilinde olan hakimiyet eksikliği nedeni ile hem de Xcode editörünün öğrenme eğirişi Unity oyun motorundan zor oluşu nedeniyle tez uygulamasının geliştirilmesinde güçlük yaşanmıştır. Ayrıca ios platformuna özgü AR ve 3D kütüphanelerine de hakimiyet gereklidir. Bu kütüphanelerin kendine özgü kuralları vardır. Bu kurallar içerisinde kalmak gereklidir. Bu sınırlamalar nedeniyle bvh dosyasının parse edildikten sonra iskelet oluşturma ve animasyon oynatma da güçlüklerle karşılaşıldı. Her bir karenin oynatma süresinin ayarlanması gibi konfigürasyonlara izin vermemesinden dolayı swift dile uygulama istenilen düzeye ulaşamadı. Bundan dolayı tez uygulamasındaki çalışmalara Unity oyun motoru ile devam edildi.

3.5. MediaPipe Pose Estimation Hesaplama Sınırlamaları

MediaPipe Pose Estimation, birçok uygulama için kullanışlı bir teknoloji olsa da, belirli koşullar altında doğruluk ve performans sınırlamalarına tabi olabilir. Bu nedenle, uygulama senaryosuna ve kullanım durumuna göre uygun bir şekilde dikkate alınmalıdır. Bu teknolojinin bazı sınırlılıkları şunlardır:

MediaPipe Pose Estimation, görüntü kalitesine duyarlıdır. Düşük çözünürlüklü veya bulanık görüntülerde doğru sonuçlar üretme yeteneği sınırlıdır.

Görüntü açısı önemlidir. Görüntüleme açısı ve bakış açısı, tahminin doğruluğunu etkileyebilir. Özellikle tam profil görüntüler veya yüzeyin düzgün bir şekilde görünmediği durumlar, yanıltıcı sonuçlara yol açabilir.

Görüntülerdeki nesnelerin bazı bölümleri görünmüyorsa (örneğin, bir elin diğer elin arkasında olması), tahmin doğruluğu olumsuz etkilenebilir.

İnsan yüzü görünmüyorsa tahminleme yapamaz. Tahminleme yapabilmesi için mutlaka insan yüzünün kameraya görünmesi gereklidir. Çünkü ilk önce yüzü algıladıktan sonra diğer uzuvların tahminini yüze göre yapmaktadır.

MediaPipe Pose Estimation, aynı anda birden fazla nesnenin pozunu tahmin etmekte sınırlıdır. Çoklu insanlar veya nesneler varsa, bunları ayrıştırmak ve doğru tahminler yapmak daha zor olabilir.

Görüntüdeki renk ve aydınlatma değişiklikleri, tahmin doğruluğunu olumsuz etkileyebilir. Özellikle dış mekan koşullarında veya farklı aydınlatma koşullarında çalışırken dikkate alınmalıdır. [41]

Bazı karmaşık veya eşzamanlı vücut pozları, tahmin algoritmasının sınırlarına ulaşabilir ve doğruluk düşebilir.

3.6. Uygulamanın Kullanım Pratiği Üzerine Tartışmalar

Uygulamanın kullanımı için cep telefonunun sabitlenmesi gerekmektedir. Kameranın yeterli alanı sağlaması için belli bir derinlik mesafesi gereklidir. Kişi eğer tek başına ise telefondan uzaklaşması gerektiği anlamına gelir. Bundan dolayı uygulamanın kişiyi uzaktan yönlendirme yapabilmesi için kişiye sesli komutlar aracılığı ile çeşitli bilgilendirmeler yapması gerekebilir.

Kişi yaptığı hareketlerin analizleri görmek için gözlemleyebileceği bir monitöre ihtiyaç duyacaktır. Uygulamadan gelen görüntülerin bu monitöre yansıtılması gerekecektir. Bu gözlemi telefonun ekranını herhangi bir monitöre yansıtarak yapması mümkündür. Bunun için ihtiyaç duyacağı ekrana yansıtma teknolojilerini kullanması gereklidir.

Yansıtacak bir ekran yok ise telefonun ön kamerası görüntü almak için kullanılabilir. Böylece kişi telefonun ekranından izleme yapabilir hale gelir. Fakat ekranın boyutu ve kişinin mesafesi hareket takibini önemli ölçüde etkileyecektir. Kişi tek başına değil ise diğer kişi telefonun ekranından bu analizi görüntüleyebilir ve yönlendirebilir. Anlık analiz gerekmiyorsa hareket analizi kayıt altına alınıp daha sonra bu görüntülenip değerlendirilme şansı vardır.

3.7. Nerelerde Kullanılabileceği ve Faydaları Hakkında Tartışma

Bu tez çalışmasında geliştirilen teknolojinin eğitim alanında birçok farklı şekilde kullanılabileceği ve faydaları öngörülmektedir.

Bu teknolojinin eğitim alanındaki faydası yüksektir. Malezya da bir üniversitede yapılan çalışma gösteriyor ki teknolojinin fiziksel eğitimde kullanılmasının öğrencilerin spor becerilerini geliştirmelerine yardımcı bir unsuru vardır. Öğrencilerin, teknoloji entegrasyonu sayesinde spor becerileri hakkında daha fazla bilgi edindikleri ve bu teknolojileri kullanmaktan memnun oldukları belirtiliyor. Teknolojinin, öğrencilerin spor becerilerini geliştirmelerine yardımcı olan önemli bir araç olduğu sonucuna varılıyor. Teknoloji kullanımının öğrencilerin motivasyonunu ve katılımını artırdığı, öğrenci merkezli bir öğrenme yaklaşımını teşvik ettiği ve öğretmenlerin geri bildirimlerinin etkinliğini artırdığı vurgulanıyor. [47]

Motor becerileri, günlük yaşamda ve spor performansında önemli bir rol oynar. Bu becerilerin geliştirilmesi ve optimize edilmesi, bireylerin pratik kazanmasına ve performanslarını arttırmasına yardımcı olabilir. Motor becerileri, tekrar ve deneyimle geliştirilen becerilerdir. Pratik kazanma süreci, doğru ve etkili bir şekilde yapılan tekrarlarla mümkündür. Ancak, bu süreçte ortaya çıkan hataları doğru bir şekilde tanımlamak ve düzeltmek, etkili bir öğrenme için kritik öneme sahiptir. Bu süreç, beynin motor kontrolü ve koordinasyonunu iyileştirmesine yardımcı olur[48].

Bu teknoloji artırılmış geri bildirim aracılığı ile hataların düzeltilmesinde ekstra bir katkı sunar. Artırılmış geribildirim, bireyin performansı hakkında ek bilgiler sağlama

amacı taşıyan bir tekniktir. Geleneksel geribildirim yöntemlerine ek olarak, artırılmış geribildirim teknolojik araçlar, sensörler ve artırılmış gerçeklik gibi ileri teknolojileri içerir. Bu teknolojiler, bireylere daha spesifik ve anlamlı geribildirimler sağlama potansiyeline sahiptir. [49, 50]

Bu teknoloji, bir bireyin motor becerilerini ayrıntılı bir şekilde değerlendirir ve olası hataları tanımlar. Ayrıca hataları tespit etmenin yanı sıra düzeltme sürecine de rehberlik eder. Analiz sonuçlarına dayanarak, bireylere spesifik geribildirimler verilebilir ve doğru teknikleri öğrenmeleri için yönlendirme sağlanabilir. Bu da motor becerilerini daha etkili bir şekilde geliştirmelerine yardımcı olur. Motor becerilerinin pratik kazanmada oynadığı rol, bu teknoloji aracılığıyla daha net bir şekilde anlaşılabilir. Hata teşhisi ve düzeltme sürecinde hareket analizinin kullanılması, bireylerin performanslarını arttırmalarına ve daha sağlıklı motor alışkanlıkları geliştirmelerine olanak tanır.

Kademelendirerek öğreterek beceri kazandırma, bilgilerin veya becerilerin zaman içinde artan zorluk seviyeleriyle öğretilmesini sağlayan bir öğrenme modelidir. Bu teori, öğrenenin önce temel becerileri kazanmasını ve ardından bu becerileri daha karmaşık durumlar için uyarlamasını teşvik eder. Hareket becerilerinin kademelendirilmiş bir şekilde öğrenilmesi, öğrencilere temel hareketlerin yanı sıra bu hareketlerin nasıl uygulanacağı konusunda da deneyim kazandırır. Bu, öğrencilerin beceri setlerini genişletmelerine ve gerçek dünya uygulamalarına daha etkili bir şekilde adapte olmalarına olanak tanır. [50]

Bu teknoloji kademelendirilmiş öğrenme süreçlerini destekleyebilir. Öğrencilere kontrollü bir ortamda farklı zorluk seviyelerinde pratik yapma ve becerilerini geliştirme imkanı sunar. Ayrıca öğrencinin zamanla nereden nereye geldiğinin izlenmesi için de bir ortam sağlanmış olur. Becerisinin zamanla nasıl geliştiğinin kaydedilmesi ve daha sonra izlenebilmesine olanak sağlar. Böylece gelişim ile ilgili bir veritabanı da oluşturulmuş olur. Daha sonra bu veritabanlarındaki bilgiler çeşitli araştırmalar için işlenebilir ve bu işlenmiş verilerden daha sonra kullanılmak üzere anlamlı bilgiler elde edilebilir. Veritabanındaki bu bilgiler demografik veriler ile birleştirilerek çeşitli başka

alanlar için karşılaştırma verisi de oluşturabilir. Örneğin öğrencilerin gelişimleri ile sosyo-ekonomik durumları arasında bir ilişki kurulabilmesine olanak sağlar.

Yapay zekanın bu teknoloji ile bir araya gelmesi ile birlikte önemli faydalar sağlanmış olur. Yapay zeka algoritmaları sayesinde bu bilgiler girdi verisi olarak kullanıldıktan sonra, hareketin analizi, beceri kazandırılması, hatalar ve geri bildirim gibi anlamlı neden ve sonuçlar oluşturarak eğitmene ve öğrenciye gelişimi için yardımda bulunulması daha kolay hala gelir. Bireyselleştirilmiş egzersiz programları önerebilir. Yapay zeka, bu verileri kullanarak bireylere özel egzersiz programları oluşturabilir, böylece kullanıcılar maksimum verimlilikle çalışabilirler. Öğrenci performans değerlendirmesinde yapay zeka kullanılarak öğrencilerin fiziksel aktivitelerini değerlendirmek ve öğrenci performansını analiz etmek için kullanılabilir. Yapay zeka, bu verileri işleyerek öğrenciye özel geri bildirimler sağlayabilir.

Ayrıca bu teknoloji ile birlikte oyunlaştırmanın kullanılması öğrenmeye katkı sağlar. Genel olarak bakacak olursak oyunlaştırma, oyun öğeleri ve mekaniklerinin eğitim süreçlerine entegre edilmesi anlamına gelir. Öğrencilerin motivasyonunu artırmak, öğrenmeyi daha eğlenceli hale getirmek ve uzun vadeli katılımı teşvik etmek amacıyla kullanılır. Oyunlaştırma, öğrenme sürecini rekabetçi ve ödüllendirici bir deneyime dönüştürebilir. Oyunlaştırmanın katacağı bazı faydalar şunlar olacaktır:

Kişiselleştirilmiş geri bildirim verilebilir. Hareket analizi ile elde edilen veriler, öğrencilere kişiselleştirilmiş geri bildirim sağlamak için kullanılabilir. Bu geri bildirim, öğrencilerin performanslarını anlamalarına ve geliştirmelerine yardımcı olabilir.

Zorluk seviyeleri ayarlanabilir. Hareket öğrenme sürecinde, oyunlaştırma araçlarıyla öğrencilere uygun zorluk seviyeleri sunulabilir. Bu, öğrencilerin kendilerini sürekli geliştirmelerini teşvik edebilir. Rekabet ve iş birliği yaratır. Öğrenciler arasında rekabeti artırmak veya iş birliğini teşvik etmek amacıyla oyunlaştırma unsurları eklenerek, öğrenme süreci daha etkili hale getirilebilir. Ödüllendirme sistemi getirir. Hareket analizi ve hareket öğrenmeyle entegre edilen oyunlaştırma, öğrencilere başarılarını ödüllendirmek için bir sistem sağlayabilir. Bu, öğrencilerin motivasyonunu artırabilir.

3.8. Hareket Analizi ve Karşılaştırma

Model verisi ile poz tahminlemeden gelen x,y,z koordinatları kaydedilip daha sonrasında sunucuya metin formatında aktarılmıştır. Bu veriler ilerde kullanılarak poz tahminleme yöntemi ile yakalanan insan hareketinin, modelin tekrar eden hareketi ile ne kadar örtüştüğü ile ilgili bilgi vermesi ve karşılaştırma analizinin yapılabilmesi mümkündür.

4. KAYNAKÇA

[1] Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1990). "Computer Graphics: Principles and Practice." Addison-Wesley.

[2] Hughes, J. F., van Dam, A., McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S. K., & Akeley, K. (2013). "Computer Graphics: Principles and Practice." Addison-Wesley.

[3] Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1997). "Computer Graphics: Principles and Practice in C." Addison-Wesley.

[4] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). "Recent Advances in Augmented Reality." IEEE Computer Graphics and Applications, 21(6), 34-47.

[5] Schmalstieg, D., & Hollerer, T. (2016). "Augmented Reality: Principles and Practice." Addison-Wesley.

[6] Azuma, R. T., Julier, S. J., & Baillot, Y. (2001). "Recent Advances in Augmented Reality." IEEE Computer Graphics and Applications, 21(6), 34-47.

[7] Reinhard, J., & Weiskopf, D. (2008). "Interactive Rendering and Visualization of Complex Real-Time Musculoskeletal Simulation Data." Computers & Graphics, 32(4), 450-458.

[8] Hsu, W., Lee, K. M., Pulli, K., & Lee, F. S. (2009). "DReal-Time Human Motion Analysis with Physically-Based Human Models." IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(4), 832-845.

[9] Moeslund, T. B., Hilton, A., & Krüger, V. (2006). "A Survey of Advances in Vision-Based Human Motion Capture and Analysis." Computer Vision and Image Understanding, 104(2-3), 90-126.

[10] Ceseracciu, Elena, Zimi Sawacha, and Claudio Cobelli. "Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept." PloS one 9.3 (2014): e87640.

[11] https://www.comparesportstech.com/compare-motion-capture-systems (Erişim tarihi: **20 Ekim 2023**).

[12] Josyula, Rohit, and Sarah Ostadabbas. "A review on human pose estimation." arXiv preprint arXiv:2110.06877 (2021).

[13] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping

Luo. Whole-body human pose estimation in the wild, 2020.

[14] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[15] S. Gary ve D. Johnson, "OpenCV: An Open Source Computer Vision Library," in International Workshop on Computer Vision, 2020

[16] Lugaresi, Camillo, et al. "Mediapipe: A framework for perceiving and processing reality." Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR). Vol. 2019. 2019.

[17] Marques, Oge, and Oge Marques. "Computer Vision and Image Analysis with the Vision Framework." Image Processing and Computer Vision in iOS (2020): 41-50.

[18] Apple Vision Framework. https://apple.co/37V2bxg, 2018. Accessed: 2023-11-27.

[19] Vaughan, C. L., Davis, B. L., & O'Connor, J. C. (1999). "Dynamics of Human Gait." Human Kinetics.

[20] Harris, C., & Stephens, M. (1988). "A Combined Corner and Edge Detector." In Alvey Vision Conference, 50, 147-151.

[21] Bartlett, R. M., & Wheat, J. S. (2019). "Analysis of Sports Performance: Advances and Challenges." Journal of Sports Sciences, 37(14), 1557-1560.

[22] Calvert, T. W., Mahler, M., Stevens, R. D., & Lee, W. O. (2004). "Real-time Motion Capture and Character Animation." In Computer Graphics Forum (Vol. 23, No. 3, pp. 591-600).

[23] Sikora, T., & Hilton, A. (2008). "Towards Consistent Mapping of Articulated Shapes." ACM Transactions on Graphics, 27(3), 38.

[24] Song, S. J., & Hodgins, J. K. (2011). "A Data-Driven Approach for Real-Time Full Body Pose Animation." ACM Transactions on Graphics, 30(4), 56.

[25] Lazor, Milica, et al. "Automation of the avatar animation process in FBX file format." FME Transactions 47.2 (2019): 398-403.

[26] Bozman, Michael, et al. "COLLADA: Digital Asset Schema Specification 1.4.1." Khronos Group, 2008.

[27] Liu, Hong and Nigg, Benno M. "A Unified Parametric Framework for the Analysis of Human Gait." Human Movement Science, 2000

[28] J. Bender, "An Overview of Modern Game Engines," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, 2016.

[29] A. Oliker, "Game Engines: A Survey of Current Technology," in IEEE Computer Graphics and Applications, vol. 31, no. 3, pp. 84-87, 2011.

[30] D. Shreiner, "The History of Game Engines," in The ACM Digital Library, 2013.

[31] T. Wend, "Comparative Analysis of Modern Game Engines," in International Journal of Computer Science and Information Security, vol. 14, no. 5, 2016.

[32] https://docs.unity3d.com/Manual/ExecutionOrder.html (Erişim tarihi: **11 Ekim 2023**).

[33] Linowes, Jonathan. *Augmented Reality with Unity AR Foundation*. Packt Publishing, 2021.

[34] Chaudhry, Tanisha, Ash Juneja, and Shikha Rastogi. "AR foundation for augmented reality in unity." *Int. J. Adv. Eng. Manage.* 3.1 (2021): 1-7.

[35] Linowes, Jonathan, and Krystian Babilinski. *Augmented reality for developers: Build practical augmented reality applications with unity, ARCore, ARKit, and Vuforia*. Packt Publishing Ltd, 2017.

[36] Linowes, Jonathan, and Krystian Babilinski. *Augmented reality for developers: Build practical augmented reality applications with unity, ARCore, ARKit, and Vuforia*. Packt Publishing Ltd, 2017.

[37] Fowler, Allan, and Allan Fowler. "The Unity ARKit." *Beginning iOS AR Game Development: Developing Augmented Reality Apps with Unity and C#* (2019): 57-102.

[38]https://storage.googleapis.com/mediapipe-assets/Model%20Card%20BlazePose%20GHUM%203D.pdf (Erişim tarihi: **11 Ekim 2023**).

[39] Buyuksalih, Ismail, et al. "3D modelling and visualization based on the unity game engine–advantages and challenges." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017): 161-166.

[40] Takrouri, Khaled, Edward Causton, and Benjamin Simpson. "AR Technologies in Engineering Education: Applications, Potential, and Limitations." *Digital* 2.2 (2022): 171-190.

[41] Latreche, Ameur, et al. "Reliability and validity analysis of MediaPipe-based measurement system for some human rehabilitation motions." *Measurement* 214 (2023): 112826.

[42] https://b4mind.com/dijital-pazarlama/sanal-gerceklik-ile-artirilmis-gerceklik-arasindaki-5-kritik-fark-2/ (Erişim tarihi: **11 Ekim 2023**).

[43] https://animost.com/ideas-inspirations/pros-and-cons-of-motion-capture-the-development-of-motion-capture-asia/ (Erişim tarihi: **11 Ekim 2023**).

[44] https://www.cursorinsight.com/sensor-video-analysis/ (Erişim tarihi: **15 Ekim 2023**).

[45] https://www.yazilimperver.com/index.php/2019/07/04/oyun-motorlarina-hizli-bir-bakis/ (Erişim tarihi: **20 Ekim 2023**).

[46] https://www.mshowto.org/oyun-motoru-nedir-unity-ve-unreal-engine-genel-ozellikleri-nelerdir.html (Erişim tarihi: **22 Ekim 2023**).

[47] Zulkifli, Ahmad Fahim, and Ajau Danis. "Technology in physical education: Using movement analysis application to improve feedback on sports skills among undergraduate physical education students." *Social Sciences & Humanities Open* 6.1 (2022): 100350.

[48] Coker, Cheryl. Motor learning and control for practitioners. Routledge, 2017.

[49] Schmidt, Richard A., and Craig A. Wrisberg. *Motor learning and performance: A situation-based learning approach*. Human kinetics, 2008.

[50] Magill, Richard, and David I. Anderson. *Motor learning and control*. New York: McGraw-Hill Publishing, 2010.