

**YENİ NESİL KINECT KAMERA İLE 3B ORTAMLARDA
NESNE SEÇME**

**OBJECT SELECTION WITH NEW GENERATION KINECT
CAMERA IN 3D ENVIROMENT**

ÖMER FARUK ÇANGIR

PROF. DR. HAŞMET GÜRÇAY

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Grafiği Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2016

Ömer Faruk ÇANGIR' ın hazırladığı “**Yeni Nesil Kinect Kamera İle 3B Ortamlarda Nesne Seçme**” adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR GRAFİĞİ ANABİLİM DALI**' nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Yrd. Doç. Dr. Serdar ARITAN

Başkan

Prof. Dr. Haşmet GÜRÇAY

Danışman

Yrd. Doç. Dr. Ufuk ÇELİKCAN

Üye

Yrd. Doç. Dr. Emre SÜMER

Üye

Yrd. Doç. Dr. Selen PEHLİVAN

Üye

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

Prof. Dr. İlyas ÇİÇEKLİ
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 tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

.../06/2016

Ömer Faruk ÇANGIR

ÖZET

YENİ NESİL KINECT KAMERA İLE 3B ORTAMLARDA NESNE SEÇME

Ömer Faruk ÇANGIR

Yüksek Lisans, Bilgisayar Animasyonu ve Oyun Teknolojileri Bölümü

Tez Danışmanı: Prof. Dr. Haşmet GÜRÇAY

Haziran 2016, 75 sayfa

İnsan bilgisayar etkileşimi; uygulamalar, oyunlar, benzetimler ve diğer birçok ortamlar için en temel unsurlardan biridir. İnsan bilgisayar etkileşimi üzerinde birbirinden farklı senaryolar için, birbirinden farklı araçlar kullanılarak geliştirilen pek çok yöntem mevcuttur. Bu yöntemler ekrandaki imleci hareket ettirme, nesne seçme gibi alanlarda yoğunlaşmıştır. Nesne seçme, insan bilgisayar etkileşiminde en çok çalışma yapılan ve en çok kullanılan alanlardan biridir. Önceleri 2B ortamlar üzerinde çeşitli yöntemler kullanılarak geliştirilmiş olan nesne seçme işlemi, daha sonraları 3B ortamlar üzerinde de geliştirilmeye başlanmıştır. Fakat 2B ortamlara özel geliştirilen yöntemler 3B ortamlara uyarlanırken pek çok farklı problem ortaya çıkmıştır. Bu nedenle 3B ortamlarla etkileşim için yeni araçlar geliştirilmiş ve bu araçlar yardımıyla etkileşim sağlanmıştır. 3B ortamlara özel olarak geliştirilen yöntemlerde bu araçların kullanılması ile beraber; kurulum süresi, işaretleyici giyme ya da taşıma zorunluluğu gibi problemler kendini göstermiştir.

Çalışma kapsamında 3B ortamlarda kurulum, işaretleyici giyme ya da taşıma gereksinimi olmadan en uygun nesne seçme yönteminin tespit edilmesi amaçlanmıştır. Bunun yanı sıra tespit edilen yöntemlerin kullanım kolaylığı, hız ve kesinlik açısından daha verimli hale getirilmesi hedeflenmiştir. Bu bağlamda, bahsedilen parametrelere göre seçilen üç farklı

yöntemin kıyaslanabilmesi için ‘el serbest’ biçimde seçme işlemi yapılabilen deneysel bir 3B oyun geliştirilmiştir. Oyunda insan bilgisayar etkileşimini sağlamak amacıyla Kinect kamera kullanılmıştır. Geliştirilen oyun Kinect ikinci sürüm kamera ile entegre edilmiş ve Kinect’in gelişmiş özellikleri kullanılarak kullanıcı etkileşiminin daha verimli hale getirilmesi sağlanmıştır. Kinect kameranın derinlik algılayan sensörleri sayesinde üç boyuttaki kullanıcı hareketleri kolayca alınarak oyunda işlenebilmiştir.

Oyunda Kinect kamera aracılığıyla yapılan nesne seçme işleminde kullanılan yöntemler, bu yöntemlerin farklı ortamlardaki davranışlarının tespit edilebilmesi amacıyla ‘nesne az yoğun’, ‘nesne çok yoğun’ ve ‘nesne hareketli’ ortamlarda test edilmiştir. Ayrıca farklı büyüklükteki nesnelerin seçim hızına etkisinin tespit edilebilmesi amacıyla test için oluşturulan ortamlardaki nesneler farklı boyutlarda seçilmiştir. Geliştirilen ortamlarda yöntemlerin kıyaslanabilmesi için gereken veriler, bu oyunu oynayan kullanıcı verilerinden elde edilmiştir. Her testin sonunda kullanıcılar oyun değerlendirme anketini doldurmuş ve ankette yer alan sorulara verilen yanıtlar incelenerek kullanıcıların yöntemler hakkındaki değerlendirmeleri analiz edilmiştir. Ardından, elde edilen veriler kullanılarak Fitts analizi yapılmış ve zorluk göstergesi ve seçme süresindeki doğru orantı gösterilmiştir. Bunun yanı sıra Fitts analizi ile yöntem ve ortamlar için performans göstergeleri hesaplanmış ve birbirleri ile kıyaslanmışlardır. Seçilen ilk iki yöntem yeni geliştirilen ortama uygun çalışabilmesi için güncellenmiş, bunun yanı sıra iki farklı yöntemden esinlenilerek geliştirilen yeni bir hibrit yöntem oluşturulmuştur. Geliştirilen bu hibrit yöntem ‘nesne az yoğun’ ve ‘nesne hareketli’ ortamlarda diğer yöntemlere yakın performans sergilerken ‘nesne çok yoğun’ ortamlarda diğer yöntemlere göre daha iyi seçme performansı sağlamıştır. Benzer şekilde katılımcı değerlendirmeleri analiz edildiğinde ‘nesne az yoğun’ ve ‘nesne hareketli’ ortamlarda kullanıcı yöntem tercihlerinde anlamlı bir fark olmadığı ancak ‘nesne çok yoğun’ ortamda kullanıcıların geliştirilen hibrit yöntemi tercih ettikleri görülmüştür.

Anahtar Kelimeler: insan bilgisayar etkileşimi, el serbest seçme, 3B ortamda nesne seçme, Kinect sensör

ABSTRACT

OBJECT SELECTION WITH KINECT CAMERA IN 3D ENVIRONMENT

Ömer Faruk ÇANGIR

Master Of Science, Computer Animation and Game Technologies

Supervisor: Prof. Dr. Haşmet GÜRÇAY

April 2016, 75 pages

Human computer interaction is one of the most fundamental element for applications, games, simulations and many other environments. There are many methods on human computer interaction to use that are developed by using different tools in different scenarios. These methods are concentrated in areas such as move the cursor on the screen and object selection. Object selectin is one of the most studied and used area in computer interaction. Object selection process, which was developed using a variety of methods on 2D environments previously, have been developed on 3D environments subsequently. While specially developed methods on 2D environments are tailored to work on 3D environments, many different problem has emerged. For this reason, new tools are developed in order to interact with 3D environments and interaction is completed with the assist of these tools. With the use of new developed tools on methods which specially developed on 3D environments; many problem is emerged such as setup time, having to carry or wear marker.

The scope of the study is determining the most appropriate method for object selection in 3D environment without the need for wearing or carrying the marker object or setup the environment. In addition, making identified methods more efficient in terms of ease of

user, speed and precision is aimed. In this context, to compare three different selected methods according to selected parameters, an 3D experimental game which allows selecting objects with free hand gestures has been developed. Kinect camera was used to ensure the human computer interaction in the experimental game. Developed game is integrated with second version Kinect camera and using the advanced features of the camera are provided to make user interaction more efficient. Thanks to the Kinect camera depth sensors that detect in three dimension movements of the user could be handled easily in the game.

The methods used in the process of selecting object via Kinect camera in the game, has been tested on sparse, dense and mobile environments in order to determine the behavior of different environments. Moreover, in order to calculate the impact on speed of different sized objects, objects are selected in different sizes in the environment created for testing. Necessary data to compare the methods are obtained from the data of the users who played the game. At the end of each test, each user has filled the evaluation survey of the game and answers to the questions in the survey were analyzed by examining the evaluation of users. Then, Fitts analysis performed using data obtained from tests and linearity between index of difficulty and selection time has been shown. First two selected methods have been updated to operate in accordance with the newly developed environment and also new hybrid method inspired by two different methods has been created. While hybrid method exhibits close performance to other methods on sparse and object moving environments, it has provided better selection performance than other methods on dense environment. Similarly, when evaluation of participants analyzed, it has been detected that there is no significant difference on method preference of participants on sparse and object moving environments but more participants preferred the hybrid method on dense environments.

Keywords: human computer interaction, hand free selection, object selection in 3D environment, Kinect sensor

TEŐEKKÜR

Tez konusunun belirlenmesinden sonuçların alındığı ana kadar tüm çalışmalarında değerli yardımlarıyla tez çalışmasında yol gösteren Sayın Prof. Dr. Haşmet Gürçay' a, çalışma boyunca desteğini benden esirgemeyen Dr. Yusuf Öztürk' e ve bana her yönden destek olan sevgili eşim Fatma Çangır' a sonsuz teşekkürü bir borç bilirim.



İÇİNDEKİLER

	<u>Sayfa</u>
ETİK.....	ii
ÖZET	iii
ABSTRACT	v
TEŞEKKÜR	vii
İÇİNDEKİLER.....	viii
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ.....	xiv
SİMGELER VE KISALTMALAR	xv
1. GİRİŞ.....	1
1.1. Problem Tanımı	1
1.2. Tez Çalışmasının Hedefi ve Kapsamı.....	3
1.3. Çalışma Planı	3
1.4. Tez Çalışmasının Uygulama Alanı	4
2. İLİŞKİLİ ÇALIŞMALAR.....	5
2.1. 2B Ortamlarda Seçme İşlemi	5
2.2. 3B Ortamlarda İzleme Aletleri İle Seçme İşlemi.....	6
2.2.1. Bubble Cursor Yöntemi.....	9
2.2.2. Depth Ray Yöntemi	10
2.2.3. Squad Yöntemi	11
2.3. 3B Ortamlarda El Serbest Seçme İşlemi.....	12
3. ÖN BİLGİ.....	14
3.1. Kinect Sensör	14
3.1.1. Genel Özellikler.....	14

3.1.2.	Kinect Sensör Hareket Yakalama.....	15
3.2.	Unity Oyun Motoru	16
3.2.1.	Genel Özellikler.....	16
3.2.2.	Unity ile Oyun ya da Uygulama Geliştirme	17
4.	DENEYSEL OYUN GELİŞTİRME	19
4.1.	DeneySEL Oyun Tasarımı	19
4.2.	3B Ortamlarda Kinect ile Kullanıcı Etkileşimi.....	21
4.3.	Geliştirme.....	21
4.3.1.	Eklem Koordinatlarının Alınması	21
4.3.2.	Koordinat Sistemi Çevriminin Yapılması	22
4.3.3.	Eklem Konumu Yumuşatma	23
4.3.4.	El Serbest Seçme İşlemi	24
4.3.5.	Ortamların Geliştirilmesi.....	25
4.3.6.	Uyarlanmış Bubble Cursor Yöntemi	26
4.3.7.	Uyarlanmış Depth Ray Yöntemi	28
4.3.8.	Uyarlanmış Hibrid Squad Yöntemi	30
4.3.9.	Kullanıcı Verilerinin Kaydedilmesi	34
5.	KULLANICI TESTLERİNİN UYGULANMASI VE VERİLERİN ANALİZ EDİLMESİ	36
5.1.	Kullanıcı Testlerinin Uygulanması.....	36
5.2.	Fitts Analizi.....	37
5.3.	Katılımcı Oyun Verilerinin Analizi	41
5.4.	Katılımcıların Daha Önceki Tecrübelerinin Analizi.....	45
5.5.	Katılımcıların Oyun Değerlendirmelerinin Analizi	47
6.	SONUÇLAR VE ÖNERİLER	51
	KAYNAKLAR.....	53

EK-1: KULLANICI DEĞERLENDİRME ANKETİ.....	56
ÖZGEÇMİŞ.....	59



ŞEKİLLER DİZİNİ

Şekil 1 Lazer silahı yönteminin basit bir mekanik parça ile çalışması (Resim [12] numaralı kaynaktan alınmıştır)	7
Şekil 2 Grossman ve Balakrishnan'ın geliştirdikleri 3B hacimsel ortam (Resim [26] numaralı kaynaktan alınıp düzenlenmiştir)	8
Şekil 3 2B ortamda Bubble Cursor yönteminde nesne seçimi (Resim [23] numaralı kaynaktan alınmıştır) a)Bubble Cursor yönteminde nesne seçilmeden önceki durum b) Bube Cursor yönteminde nesne seçiminin yapıldığı durum.....	9
Şekil 4 3B ortamda Bubble Cursor yönteminde nesne seçme işlemi (Resim [24] numaralı kaynaktan alınmıştır) a) Bubble Cursor yönteminde hedefin görünmediği durum b) Bubble Cursor yönteminde hedefin önündeki nesnelere yarı geçirgen hale getirildiği için hedefin görüldüğü durum	10
Şekil 5 Depth Ray yönteminin geliştirildiği fanus biçimindeki özel ortam ve gönderilen ışın aracılığıyla gerçekleştirilen seçme işlemi (Resim [22] numaralı kaynaktan alınmıştır)	10
Şekil 6 Depth Ray yönteminde aynı ışın üzerinde kalan nesnelere seçimi a) İşaretleyici belirlenen ilk konumda bulunuyorken b) İşaretleyici fanustan uzaklaştırıldığında c) İşaretleyici fanusa yaklaştırıldığında (Resim [22] numaralı kaynaktan alınmıştır)	11
Şekil 7 Squad yöntemi nesne seçim adımları (Resimler [25] numaralı kaynaktan alınmıştır) Resimler sırasıyla seçilmek istenilen nesnenin bulunduğu çember biçiminde bir alanın seçilmesi ve bu alandaki nesnelere farklı bir arayüzde gruplar halinde sınıflandırılması işlemlerini belirtmektedir.	12
Şekil 8 Kinect sensör parçaları (Resim [33] numaralı kaynaktan alınıp düzenlenmiştir) ...	15
Şekil 9 Kinect Eklem Türleri Listesi(Kırmızı renkle yazılan eklemler Kinect ilk nesil sensörlerden alınamayıp yeni nesil Kinect sensörlerden alınabilen eklemleri belirtmektedir)	16
Şekil 10 Unity Ana Düzenleyici Pencere	17
Şekil 11 3B Nesne Seçme oyunu genel sistem mimarisi.....	20

Şekil 12 Kinect Kamera koordinat sistemi (Resim [40] numaralı kaynaktan alınmıştır) ...	22
Şekil 13 Kinect sensör, ekran ve dünya koordinat sistemleri.....	23
Şekil 14 Nesne seçme işleminde kullanılan el pozisyonları a) El açık pozisyon herhangi bir seçme işlemi yapılmak istenmediği durumlarda kullanılır b) El yumruk pozisyonu el açık pozisyonda iken nesne seçimi yapılmak istendiğinde kullanılır	24
Şekil 15 a) Nesne az yoğun ortam b) Nesne çok yoğun ortam c) Nesne hareketli ortam ...	25
Şekil 16 Nesne yoğun ortam farklı açılardan görünüşleri a)Önden görünüş b)Sağdan görünüş c)Arkadan görünüş d)Soldan görünüş e)Üstten görünüş f)Alttan görünüşü	25
Şekil 17 Uyarlanmış 3B Bubble Cursor Yöntemi seviye ekranı.....	26
Şekil 18 Uyarlanmış 3B Depth Ray Yöntemi seviye ekranı	28
Şekil 19 Uyarlanmış 3B Hibrit Squad Yöntemi seviye ekranı.....	31
Şekil 20 Uyarlanmış Hibrit Squad yönteminde sol kol seçim işlemi(solda) ve seçim bölgeleri(sağda)	33
Şekil 21 Veritabanı tablo ilişkileri.....	35
Şekil 22 Kullanıcı testlerinin yapıldığı test ortamı	37
Şekil 23 Nesne az yoğun ortamda zorluk göstergesine bağlı seçim süresi değişimi.....	39
Şekil 24 Nesne yoğun ortamda zorluk göstergesine bağlı seçim süresi değişimi	39
Şekil 25 Geliştirilen yöntemler için zorluk göstergesine bağlı işlem süresi değişimi.....	40
Şekil 26 Seviye tamamlama ortalamalarının yöntemlere ve ortamlara göre dağılımı	42
Şekil 27 Seviye tamamlama ortalamalarının yöntemlere göre dağılımı.....	44
Şekil 28 Seviye tamamlama ortalamalarının ortamlara göre dağılımı	45
Şekil 29: Katılımcıların Video Oyunu Oynama Sıklığı.....	46
Şekil 30 3B ortamlar için geliştirilmiş oyun oynama sıklığı.....	46
Şekil 31 Uzaktan oyun kontrolüne izin verilen kısımlar olan oyun oynama sıklığı	46
Şekil 32 Kinect kamera kullanılarak uzaktan oyun kontrolüne izin verilen kısımlar olan oyun oynama sıklığı	47
Şekil 33 Katılımcıların kendi oyun becerilerini sınıflandırma değerleri.....	47

Şekil 34 Katılımcılar tarafından yöntemlerin işlevsel olarak değerlendirilme grafiği 48

Şekil 35 Katılımcılar tarafından ortam bazlı olarak tercih edilen yöntem grafiği..... 49



ÇİZELGELER DİZİNİ

Tablo 1 Etkileşim Sistemlerinin özellikleri	2
Tablo 2 User tablosu alanları ve özellikleri.....	34
Tablo 3 Experiment tablosu alanları ve özellikleri.....	35
Tablo 4 Experiment_step tablosu alanları ve özellikleri	35
Tablo 5 Testler esnasında seçilen nesnelerin koordinat ve seçim alanı büyüklükleri	38
Tablo 6 Seviye tamamlama ortalamalarının karşılaştırılması	41
Tablo 7 Yönteme göre gruplanmış veriler üzerinde yapılan Tukey HSD post hoc çoklu karşılaştırma sonuçları	43
Tablo 8 Ortama göre gruplanmış oyun verileri üzerinde yapılan Tukey HSD post hoc çoklu karşılaştırma sonuçları	44
Tablo 9 Ortama göre gruplanmış değerlendirme verileri üzerinde yapılan Tamhane post hoc çoklu karşılaştırma sonuçları	49

SİMGELER VE KISALTMALAR

Kısaltmalar

3B	Üç Boyut
2B	İki Boyut
SDK	Software Development Kit
API	Application Programming Interface
RGB	Red Green Blue
APS	Active Pixel Sensor
DOF	Degrees of Freedom
DEMO	Demonstration
ANOVA	Analysis of Variance
ID	Index of Difficulty
IP	Index of Performance

1. GİRİŞ

1.1. Problem Tanımı

Bilgisayar grafiđi, kısaca bilgisayar kullanılarak grafiklerin oluşturulması ve özel grafik donanım yazılımlarından bilgisayarla elde edilen resim bilgilerinin ifade edilmesi olarak tanımlanabilir. Bilgisayar grafikleri kullanılarak ortaya çıkarılan sanal ortamlar bilgisayarlar, oyun konsolları, tabletler, akıllı telefonlar gibi pek çok farklı ortamda kullanılırlar. Sanal ortamların gelişimi ile birlikte günlük hayatta kullanımı yaygınlaşmış ve bununla birlikte insanlarla bilgisayarların etkileşiminin gerekliliđi ortaya çıkmıştır. Bilgisayar kullanıcı etkileşimi bilgisayar grafikleri sayesinde oldukça kolaylaşmıştır. Bilgisayar kullanıcı etkileşimi arttıkça pek çok türde etkileşim aracı ortaya çıkmıştır.

Bilgisayar tarafından üretilen grafikler iki boyutlu (2B), üç boyutlu (3B) ve canlandırılmış grafikler olarak ele alınır. Önceleri daha çok 2B grafikler yaygın olarak kullanılmaktayken teknolojinin gelişimi ile birlikte 3B bilgisayar grafikleri daha çok kullanılmaya başlanmıştır. Bu durum, 3B grafikler ile etkileşim için kullanılan ve kullanımı daha kolay olan araçlara ihtiyacı arttırmaktadır. Klavye, fare ya da dokunmatik etkileşim araçları, genel olarak 2B arayüzler için tasarlanmıştır. Dolayısıyla bu araçlar 3B arayüzler üzerinde yeterli seviyede kullanışlı değildir. Bu nedenle 3B grafiklerle çalışılırken daha etkili, hızlı, hassas ve kullanımı kolay donanımlar geliştirilmeye ve kullanılmaya başlanmıştır.

Bilgisayar grafiđi hangi tür ortamda kullanılırsa kullanılsın, seçme işlemi kullanıcı bilgisayar etkileşiminin sağlanabilmesi için kullanılan en temel unsurlardan biridir [1]. Bu nedenle 3B grafiklerle insan bilgisayar etkileşiminin geliştirilmesi amacıyla 3B grafikler ile çalışan pek çok seçme yöntemi araştırılmaya başlanmıştır. Yapılan araştırmalar ve geliştirmeler sonucunda 3B grafiklerle etkileşim sağlanması amacıyla 3B izleme araçları ile seçme işlemi kullanılmaya başlanmıştır.

İzleme araçları vücudun belirli bölgelerine giyilen ya da el ile tutulan işaretleyicilerdir. İzleme araçları ile çalışan sistemler, işaretleyicileri izlerken sistemdeki ilgili nokta ile işaretleyicileri eşleştirerek çalışır. Bunun yanı sıra işaretleyiciler ile çalışmak pek çok güçlüğü beraberinde getirir. Etkileşim yapılmadan önce işaretleyicilerin yerleştirilme zorunluluđu, kullanıcıya özel kalibrasyon yapılma gereksinimi ve işaretleyicilerin kişi hareketlerini kısıtlaması gibi nedenlerden dolayı bu tarz yöntemler tercih edilmemektedir. Buna rağmen insan bilgisayar etkileşimi amacıyla geliştirilen bu yöntemler, kullanımı daha kolay ve ayar gerektirmeyen yöntemlere ön ayak olmuştur.

Dam ve diğerleri [2] ‘el serbest yöntem’ aracılığıyla üzerinde gezme, bastırma, tutma gibi seçme tekniklerinin yanı sıra pek çok farklı işlemin de yapılabildiğini göstermiştir. İlgili çalışmada el serbest teknikle pek çok işlemin yapılabileceği gösterilmesine karşın bu işlemlerin performans olarak kullanıcıların aşına olduğu diğer yöntemlere göre yetersiz olduğu tespit edilmiştir. Buna rağmen teknolojinin gelişmesi ve yeni yöntemlerin ortaya çıkması ile el serbest yöntemlerin performansı, bir işaretleyici giymeyi ya da tutmayı gerektiren yöntemler karşısında artmaktadır.

2B arayüzler kullanılarak gerçekleştirilen, 3B arayüzler üzerinde izleme araçları kullanılarak gerçekleştirilen ve 3B arayüzler üzerinde el serbest biçimde gerçekleştirilen etkileşim sistemleri ile ilgili genel özellikler Tablo 1’de belirtilmiştir.

Tablo 1 Etkileşim Sistemlerinin özellikleri

Etkileşim Sistemi	Gerçekleştirim	Üstün Yanları	Düşük Yanları
2B arayüzler üzerinde gerçekleştirilen	Klavye, fare ya da dokunmatik etkileşim araçları kullanılarak gerçekleştirilir.	Uzun zamandır kullanılması ve üzerinde pek çok farklı çalışma yapılması nedeniyle kullanım kolaylığı, etkileşim hızı ve imleç kesinliği açısından yüksek performans sergiler.	Nesne yoğunluğunun yüksek olduğu durumlarda iki boyuta sahip olduğundan dolayı etkileşim zorluğu problemini ortaya çıkarır.
3B arayüzler üzerinde izleme araçları kullanılarak gerçekleştirilen	İşaretleyicileri izlerken sistemdeki ilgili nokta ile işaretleyicileri eşleştirerek çalışır.	3B ortamlarla etkileşimi, 2B ortamlar için geliştirilen yöntemlere göre kolaylaştırmıştır.	Etkileşim yapılmadan önce işaretleyicilerin yerleştirilme zorunluluğu, kullanıcıya özel kalibrasyon yapılma gereksinimi ve işaretleyicilerin kişi hareketlerini kısıtlaması problemlerini beraberinde getirir.
3B arayüzler üzerinde el serbest biçimde gerçekleştirilen	Sensörler aracılığıyla kullanıcı hareketleri izlenerek sistemle etkileşim yapılarak	İşaretleyicilerin yerleştirilme zorunluluğunu, kullanıcıya özel kalibrasyon yapılma gereksinimini ve	Etkileşim hızı ve imleç kesinliği açısından 2B ortamlar için geliştirilen sistemler kadar yüksek

	çalışır.	işaretleyicilerin kişi hareketlerini kısıtlaması problemlerini ortadan kaldırmıştır.	performans sergileyemez.
--	----------	--	--------------------------

1.2. Tez Çalışmasının Hedefi ve Kapsamı

İnsan bilgisayar etkileşimi birkaç temel işlev aracılığıyla kolaylıkla sağlanabilmektedir. Bowman ve Hodges [3] yaptıkları çalışma ile insan bilgisayar etkileşiminin gezinme, seçme ve düzenleme ana başlıkları altında ele alınabileceğini belirtmiştir. İnsan bilgisayar etkileşiminin sağlanmasında nesne seçimi hemen hemen tüm arayüz etkileşimlerinde kullanılan temel bir işlemdir. Bu nedenle tez çalışmasında, insan bilgisayar etkileşimini daha verimli hale getirmek için kullanılabilir yöntemleri tespit etmede nesne seçme işlemi temel alınmıştır. Bu kapsamda, Kinect yeni nesil kamera kullanarak 3B ortamlarda nesne seçme işleminin daha kolay kullanılabilir, daha hızlı ve kesin bir biçimde yapılabilmesini sağlamak amacıyla kullanılabilir yöntemlerin tespiti ve geliştirilmesi hedeflenmiştir. Bu bağlamda geliştirilen yöntemlerin farklı ortamlardaki performansları incelenerek; hangi ortamda, hangi yöntemin, kullanım kolaylığı, hız ve kesinlik özellikleri bakımından uygun olacağı tespit edilmiştir.

1.3. Çalışma Planı

Tez çalışmasının hedef ve kapsamı belirlendikten sonra bu hedef doğrultusunda çalışma planı çıkarılmıştır. Tez çalışmasının başında belirlenen ve izlenen adımlar aşağıda belirtilmiştir;

Araştırma:

- Daha önceleri yapılmış olan insan bilgisayar etkileşimi çalışmalarının incelenmesi
- İnsan bilgisayar etkileşimlerinden seçme alanındaki çalışmaların incelenmesi varsa DEMO'larının yapılması
- Geliştirilmiş olan el serbest seçme yöntemlerinin incelenmesi ve derinlik algılayan kameraların yeteneklerinin öğrenilmesi
- Önceki çalışmalardaki yöntem kıyaslamalarını da inceleyerek 3B ortamda el serbest seçme işlemine en uygun yöntemlerin tespit edilmesi ve kullanılacak derinlik algılayan kameranın belirlenmesi

- Seçilen en uygun yöntemlerin iyi ve kötü yönleri araştırılarak kullanım kolaylığı, kesinlik ve hız gibi alanlarda diğerlerinden daha iyi çalışabilecek bir yöntem önerisi yapılması

Geliştirme:

- El serbest etkileşimde kullanılacak olan derinlik algılayan kameranın SDK'sı üzerinde çalışma yapıp oyunda kullanılabilir kod kısımlarının geliştirilmesi
- Geliştirilecek oyunun modüllerinin belirlenmesi ve bu modüllerin geliştirilmesine izin verecek bir 3B oyun motoru tercihinin yapılması
- Tercih edilen 3B oyun motoru ile derinlik algılayan kamera arasındaki iletişimin kurulması ve yalnızca el hareketini algılayan ve ekranda bir cisim hareket ettiren bir deneme oyununun geliştirilmesi
- Oyunda bulunacak olan birbirinden farklı ortamların oluşturulması,
- Belirlenen tasarıma göre yöntemlerde ihtiyaç duyulacak kod kısımlarının yazılması
- Yöntemlerin uygulamaya entegre edilmesi
- Test verilerinin tutulacağı veri tabanı tasarımının yapılması ve geliştirilen oyunla çalışabilir hale getirilmesi

Yorumlama:

- Test işlemlerinin yapılacağı katılımcılara sorulacak soruların belirlenmesi
- Katılımcılara geliştirilen oyun oynatılarak sonuçların elde edilmesi
- Sonuçların yorumlanabilmesi için veri tabanı ile yorumlamada kullanılacak aracın entegrasyonunun yapılması
- Elde edilen sonuçların yorumlanması

1.4. Tez Çalışmasının Uygulama Alanı

3B ortamlarda nesne seçimi, günlük hayatımızı çevreleyen ve teknolojinin kullanıldığı pek çok alanda yararlanılabilecek bir yöntemdir. Günümüzde pek çok bilgisayar grafiği uygulaması 2B ortamdan 3B ortama geçiş yapmaktadır. Bu kapsamda 3B bilgisayar grafiğinin kullanıldığı tüm alanlarda nesne seçimi önemli bir işlev olmuştur. Nesne seçme işlemini birçok açıdan iyileştirmeye çalıştığımız tez çalışması ile birlikte, eğitim, sağlık, reklamcılık, turizm, animasyon, oyun gibi uygulama alanlarının tümüne katkı sağlayacak bir sonuç elde edilmiş olacaktır.

2. İLİŞKİLİ ÇALIŞMALAR

2.1. 2B Ortamlarda Seçme İşlemi

İlk ortaya çıkarılan bilgisayar grafikleri ile insan etkileşimi sağlanırken; seçme işlemi, en sık kullanılan etkileşim yöntemlerinden biridir. Seçme işlemi için kullanılan yöntemlerden ilki seçim yapılabilmesi için imlecin tam olarak seçilecek nesnenin üzerine gelmesi kuralı ile çalışan nokta imleç seçme yöntemidir. Bu yöntem masaüstü bilgisayarlar ya da dokunmatik ekranlar üzerinde en çok kullanılan tekniklerden biridir. Buna rağmen farklı platformlarda, farklı grafik ortamları için yeni yöntemlere olan gereksinim artmıştır.

2B yüzeylerde hedeflerin üst üste binmesi nesne seçimini zorlaştırmaktadır. Bu amaçla Yatani ve diğerleri [4] üst üste binen nesnelere her biri için farklı yönlerdeki ok işaretleri ile işlemiştir. Dokunmatik 2B yüzeylerdeki nesne seçimini kolaylaştırmak amacıyla geliştirilen bu yöntemde, kullanıcı parmağını ekrana dokundurduktan sonra seçmek istediği nesnenin ekranda belirtilen yönüne doğru parmağını sürükleyerek seçme işlemi kolayca yapılabilmektedir. Geliştirilen bu yöntem sayesinde çok fazla nesnenin üstü üste gelmesi zorunluluğu olan pek çok uygulama dokunmatik yüzeylerde de kolaylıkla kullanılabilir. Bunun yanı sıra artırılmış gerçeklik ortamlarında üstü üste binen nesnelerin kolaylıkla seçilebilmesi amacıyla Mossel ve diğerleri ekranda dokunulan yerden sanal ortama doğru dik bir biçimde ışın göndererek çalışan ve ışınla kesişen nesnelerin farklı bir ekranda listelenerek daha kolay seçilmesini sağlayan 'DrillSample' yöntemini geliştirmişlerdir [5].

Büyük ebatlı ekranlarda seçme işlemi kolaylaştırmak ve seçim süresini düşürmek amacıyla Bragdon ve Ko [6], kullanıcının kendisinden uzakta bulunan ve seçmek istediği nesneye doğru düz bir çizgi çizmesiyle başlayan ve daha sonra seçmek istediği nesne için ekranda beliren sembolü çizerek seçme işlemi tamamlayan bir yöntem geliştirdiler. Bu yöntem sayesinde büyük ebatlı ekranlarda kullanıcı yer değiştirme zahmetine girmeden uzaktaki bir nesneyi kolaylıkla seçebilmektedir.

İmleci hareket ettirme hızı da sanal ortamlarda seçim süresine etki eden faktörlerdendir. Chapuis ve diğerleri [7] imlecin hareket hızına göre seçim alanının değiştiği 'DynaSpot' yöntemini geliştirmişlerdir. Bu yöntemde imleç yavaş hareket ediyorken ya da hareket etmiyorken, nesne seçim alanı küçülmekteyken; imleç hareket hızı arttıkça nesne seçim alanı büyümektedir. Chapuis ve diğerleri imleç hızının da seçme işleminde hesaba katılması ile nesne seçim süresinin kısaldığını göstermişlerdir.

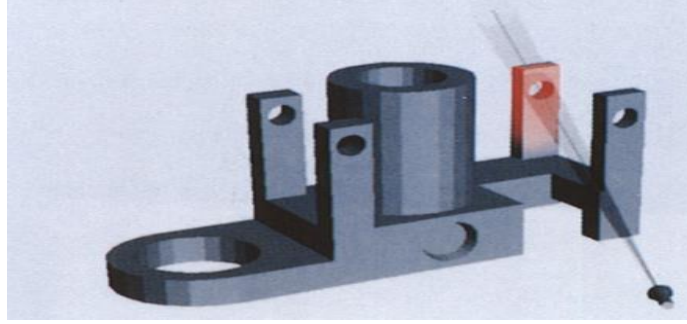
Wobbrock ve Gajoz [8] nokta imleç yönteminin motor nöron rahatsızlığı bulunan insanlarda başarısız sonuçlar verdiğini göstermişlerdir. Bu amaçla hedefin direkt olarak seçilmesi ile değil, çaprazlanması ile seçilebilmesini sağlayan bir yöntem geliştirmişlerdir. Geliştirilen bu yöntem ile motor nöron bozukluğu olan kullanıcıların da akıcı ve istikrarlı bir biçimde seçme işlemini gerçekleştirebilmeleri sağlanmıştır. Yine engelli kullanıcılar için seçme işlemini kolaylaştırma ve hızlandırma amacıyla Findlateri ve diğerleri [9], imleç tıklama ve çaprazlama ile seçme işlemi yapılabilen bir yöntem geliştirmişlerdir. Bu yöntemde imleç ile seçme işlemi yapıldığında imlecin etrafında seçilmek istenilen alandaki nesnelere bulunduğu çember biçiminde bir menü belirmektedir. Daha sonra seçilmek istenilen nesneye doğru çaprazlama yapılarak nesne seçimi gerçekleştirilmektedir. Geliştirdikleri yöntem üzerinde yaptıkları test sonuçları engelli kullanıcılar için seçme zamanını kısalttığını göstermektedir.

2.2. 3B Ortamlarda İzleme Aletleri İle Seçme İşlemi

3B sanal ortamlar pek çok alanda yoğun olarak kullanılmakla birlikte her geçen gün yeni alanlarda da kullanılmaktadır. Bu durum 3B ortamlarda insan bilgisayar etkileşiminin geliştirilmesi ihtiyacını ortaya çıkarmıştır. Etkileşime yeni bir boyut daha eklenmesi, etkileşimin zorluğunu arttırmıştır. Bu sorunu ortadan kaldırmak amacıyla pek çok yöntem geliştirilmiştir. Bu alanda yapılan öncü çalışmalardan biri Zhai ve diğerlerinin [10] ortaya çıkardığı ‘Silk Cursor’ yöntemidir. Bu yöntem etkileşimde kullanılacak imlecin dikdörtgenler prizması şeklinde hacimsel bir hale getirilmesi ile uygulanmıştır. Yapılan bu çalışma ile hacimsel imleç kullanılarak yapılan seçme işleminin nokta imleç kullanılarak yapılan seçme işlemine göre daha kısa sürede tamamlandığı gösterilmiştir.

3B nesnelere etkileşim amacıyla kullanılan yöntemlerden en yaygın olanlarından biri ‘Ray Casting’ yöntemidir. İlk olarak Roth tarafından geliştirilen ‘Ray Casting’ yöntemi [11] gönderilen ışık ışınları aracılığıyla katı cisimlerin modellenmesi ve aydınlatılması amacıyla kullanılmıştır. Bu yöntem birçok araştırmacıya daha farklı alanlarda da ilham kaynağı olmuştur. Nesne seçme alanında da etkileri görünen bu yöntemi kullanarak Liang ve Green lazer silahı adını verdikleri bir yöntem geliştirmişlerdir [12]. Bu yöntemde seçme işlemi el ile tutulan bir cihazdan gönderilen ışınlar aracılığı ile gerçekleştirilir ve ışının içinden geçtiği nesne seçilir. El fenerinin çalışma mekanizmasına benzer biçimde çalıştığı için bu

yöntem 'Flashlight Selection' olarak da adlandırılmaktadır. Çalışma prensibi Şekil 1'de gösterilen bu yöntem nesne seçme işlemi üzerine yapılacak çalışmaların önünü açmıştır.



Şekil 1 Lazer silahı yönteminin basit bir mekanik parça ile çalışması (Resim [12] numaralı kaynaktan alınmıştır)

Ortega [13] 'Flashlight Selection' çalışma mantığına benzer biçimde çalışan fakat sezgisel yaklaşım kullanan 'Hook' yöntemini ortaya çıkarmıştır. 'Hook' yöntemi ekrandaki nesnelerin imlece uzaklıklarına göre sürekli olarak puanlanması yöntemi ile çalışmaktadır. Cashion ve LaViola [14] 'Hook' yöntemini ve imlecin hareket hızına göre seçim alanının değiştiği 'DynaSpot' yöntemini geliştirmiş ve her ikisinin de bir arada kullanıldığı bir yöntem önermişlerdir.

'Ray Casting' yöntemi kullanılarak geliştirilen bir başka yöntem ise Mine [15] tarafından geliştirilmiştir. Bu çalışmada Mine, hareketin yönünü belirlemek için kullanılacak ışın gönderme, ilgili yöne bakma gibi yöntemleri incelemiş ve bu yöntemlerin hareket, seçme, değiştirme ve ölçekleme gibi işlemlerde kullanımlarını incelemiştir. Bunun yanı sıra Mine ve diğerleri [16] iki el kullanarak ışın aracılığıyla hareket yönünün belirlenebileceği bir yöntem önermişlerdir. Ayrıca bahsedilen yöntemler gibi 'Ray Casting' yöntemi temel alınarak ve el tarafından ışın gönderilerek etkileşim yapılabilen birçok farklı çalışma yapılmıştır [17] [18].

El kullanılarak gönderilen ışın aracılığıyla yapılan etkileşim, nesnelerin birbirlerini örtmesi gibi durumlarda çok etkin çalışmayabilir. Bu durumun önüne geçebilmek için elin yanı sıra göz ile de seçimi destekleyecek etkileşim çalışmaları ortaya çıkmıştır [19] [20] [21]. Bu sayede arkada kalmış ya da daha az kısmı görünen nesnelerin de kolaylıkla seçilebilmesi sağlanmıştır.

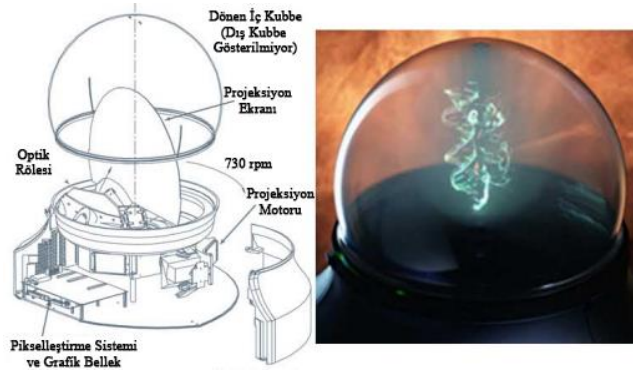
Nesne yoğun ortamlarda ışın aracılığıyla seçme işlemi sırasında ortamda bulunan nesnelerin yoğunluğuna bağlı olarak ışın birden fazla nesne ile kesişebilir. Grossman ve

Balakrishnan [22] bu sorunu çözmek için ışın üzerinde hareket eden bir noktaya en yakın olan nesnenin seçilmesi prensibine dayalı çalışan 'Depth Ray' yöntemini geliştirmişlerdir. Bu yöntemi yine geliştirdikleri özel ortamda test etmişler ve ortalama 'Depth Ray' yönteminin yine kendi geliştirdikleri diğer yöntemlere göre daha hızlı olduğunu göstermişlerdir.

3B grafik ortamlarda nesne seçmek için geliştirilen bir başka teknik ise 'Bubble Cursor' yöntemidir. İlk olarak 2B grafik ortamı için geliştirilmiş olan ve nesnelerin tam olarak üzerine geldiğinde değil de belirli bir algoritmaya göre yakınında bulunduğu da seçilmesini sağlayan bu yöntem seçme işlemini, kesin bir biçimde üzerine gelerek seçme yöntemine göre bariz bir biçimde hızlandırmıştır [23]. Daha sonra yöntem elle tutulan bir nesne yardımıyla elin konumunu 3B ortama yansıtarak çalışacak biçimde güncellenmiş ve 3B ortamlarda seçme işlemine yeni bir soluk getirmiştir [24]. Bu yöntemin yine kendi geliştirdikleri 'Depth Ray' yöntemine yakın performans verdiğini hatta az yoğun ortamlarda biraz daha verimli çalıştığını göstermişlerdir.

Nesne yoğun ortamlarda o ana kadar geliştirilen yöntemlerin yetersiz kaldığını fark eden Kopper ve diğerleri [25] 'Squad' yöntemini geliştirmişlerdir. Bu yöntem ile nesne kalabalık ortamda nesne seçimi, birden fazla adıma çıkarılmasına rağmen daha kesin hale getirilmiştir.

Tüm bunların yanında Grossman ve Balakrishnan [26] Şekil 2'de görüldüğü gibi özel bir ortamda yaptıkları çalışmalarında, 3B hacimsel ortamlarda nesne seçmek için işaretleyici ile yapılan ileri geri hareketi ile seçme işleminin diğer yönlere doğru yapılan harekete göre daha yavaş olduğunu tespit etmişlerdir. Bu sayede geliştiricilerin ortamları geliştirirken bu duruma dikkat etmeleri gerekliliği ortaya konulmuştur.

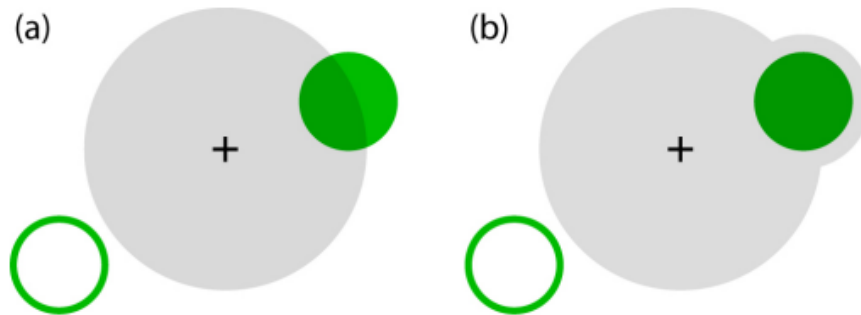


Şekil 2 Grossman ve Balakrishnan'ın geliştirdikleri 3B hacimsel ortam (Resim [26] numaralı kaynaktan alınıp düzenlenmiştir)

3B takip araçları ile seçim işlemi için pek çok yöntem geliştirilmiş ve uygulanmış olmasına rağmen, bu yöntemler bahsedilen sorunlardan ötürü çok yaygın kullanım alanına sahip olmamıştır. Bu duruma çözüm olarak el serbest hareketleri algılayabilen ve daha düşük bütçe gerektirdiğinden dolayı ticari olarak kullanımı yaygınlaşan kameralar (Microsoft Kinect, Asus xtion vb..) ve bunların kullanıldığı teknikler geliştirilmeye başlanmıştır. Bu türdeki araçlar kullanılarak geliştirilen yöntemler, vücuda herhangi bir araç, işaretleyici giyme ya da işaretleyicileri el ile tutma gereksinimini ortadan kaldırmıştır. Aynı zamanda bu yöntemler daha az kurulum ayarı gerektirmektedir.

2.2.1. Bubble Cursor Yöntemi

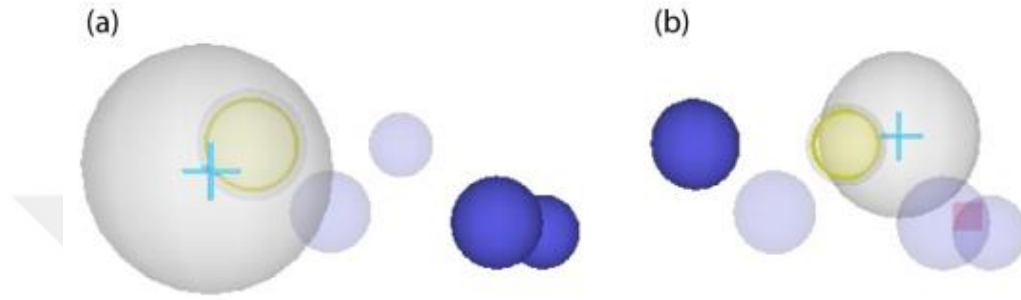
Ortamda daima tek bir nesnenin seçili olmasını esas alan bu yöntem ilk olarak iki boyutlu ortamlar için geliştirilmiştir. Alışlagelmişin dışında imlecin daima seçilecek nesnenin üzerinde bulunması zorunluluğunu ortadan kaldırarak seçilmek istenilen nesneye yaklaşılmasıyla o nesnenin seçilebilmesini sağlayan bir yöntemdir [23]. İmleç hangi konumda olursa olsun ekrandaki nesnelere her zaman yalnız bir tanesinin seçili olmasını gerektirir. Bu özelliğin sağlanabilmesi için imlecin etrafında dinamik olarak boyutları değişen bir seçim çemberi oluşturulur ve bu çember o anda seçili olan nesneyi tamamiyle kaplayacak büyüklüktedir. Çemberin boyutları imlecin o anki konumuna en yakın olan iki nesnenin seçilmesinin ardından daha yakın olanın çevresindeki en uzak nokta ile daha uzak olanın çevresindeki en kısa noktaya olan uzaklıkların karşılaştırılması sonucu daha kısa mesafenin seçilmesi prensibine göre ayarlanır. Burada en yakın iki nesne seçilirken nesnelere merkezinin imlece en yakın olduğu noktalar dikkate alınır. Yönteme göre nesne seçimi Şekil 3’te görülebilir.



Şekil 3 2B ortamda Bubble Cursor yönteminde nesne seçimi (Resim [23] numaralı kaynaktan alınmıştır) a)Bubble Cursor yönteminde nesne seçilmeden önceki durum b)

Bube Cursor yönteminde nesne seçiminin yapıldığı durum

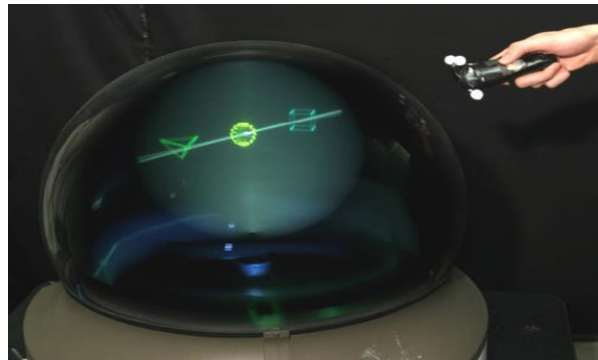
Bu yöntem daha sonraları elle tutulan bir nesne yardımıyla elin konumunu 3B ortama yansıtarak çalışacak biçimde güncellenmiş ve 3B ortamlarda da seçme yapabilecek biçimde geliştirilmiştir [24]. Yöntem üç boyuta genişletilirken en yakın iki noktanın bulunması işleminde çember yerine dairesel seçici kullanılmıştır. Ayrıca 3B ortama geçildiğinde karşılaşılan örtme problemine karşı Şekil 4’te görüldüğü gibi imlecin belirli bir mesafe uzağındaki tüm nesnelere yarı geçirgen hale getirilmiş ve bu sayede başka nesnelere tarafından örtülen nesnelere bu yöntemle seçilebilmesi sağlanmıştır.



Şekil 4 3B ortamda Bubble Cursor yönteminde nesne seçme işlemi (Resim [24] numaralı kaynaktan alınmıştır) a) Bubble Cursor yönteminde hedefin görünmediği durum b) Bubble Cursor yönteminde hedefin önündeki nesnelere yarı geçirgen hale getirildiği için hedefin görüldüğü durum

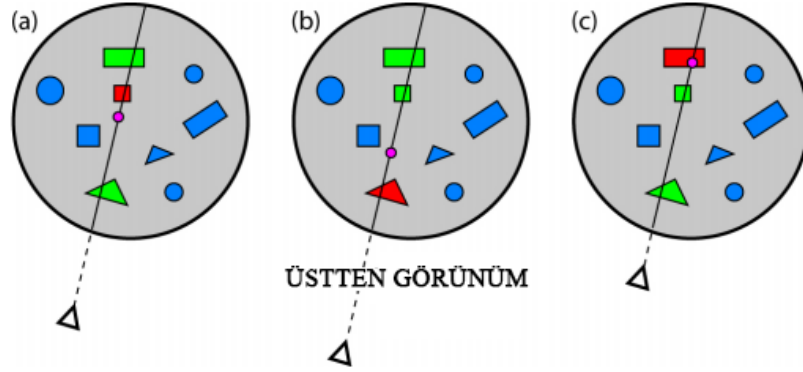
2.2.2. Depth Ray Yöntemi

‘Depth Ray’ yöntemi ışın üzerinde hareket eden bir noktaya en yakın ve ışın üzerinde olan nesnenin seçilmesi prensibine dayalı bir yöntemdir. Bu yöntem Grossman ve Balakrishnan tarafından geliştirilen Şekil 5’te görülen fanus biçimindeki bir üç boyutlu ortama taşınmış ve geliştirilen özel bir aletle ışın gönderilerek seçme işlemi uygulanmıştır [22].



Şekil 5 Depth Ray yönteminin geliştirildiği fanus biçimindeki özel ortam ve gönderilen ışın aracılığıyla gerçekleştirilen seçme işlemi (Resim [22] numaralı kaynaktan alınmıştır)

Işın fanus biçimindeki hacimli ekranın farklı yönlerinden gönderilebilmektedir ve bu sayede birbirini örten nesnelere farklı açılardan ışın gönderilerek daha kolay biçimde seçilebilmektedir. Bu kolaylığın yanı sıra Şekil 6’da görüldüğü gibi ışının üzerinde birden fazla nesne kalması durumunda kullanıcı el ile tuttuğu aleti fanustan uzağa ya da yakına hareket ettirerek ışın üzerindeki imleci hareket ettirebilir ve ışın üzerinde bulunan imlece en yakın nesneyi daha kolay biçimde seçebilmektedir.



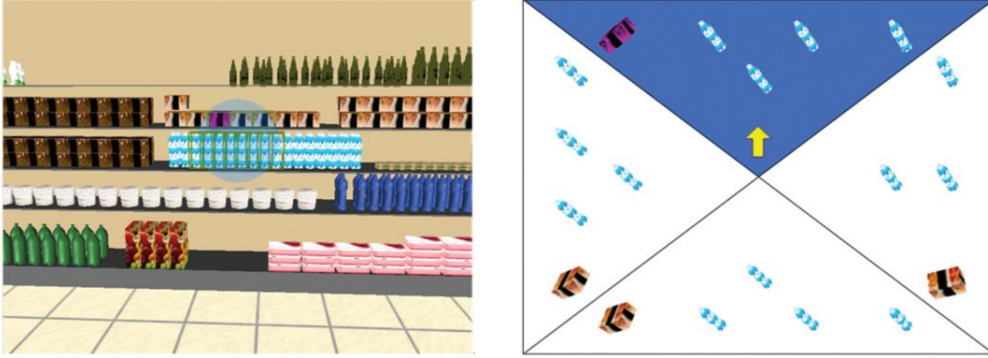
Şekil 6 Depth Ray yönteminde aynı ışın üzerinde kalan nesnelerin seçimi a) İşaretleyici belirlenen ilk konumda bulunuyorken b) İşaretleyici fanustan uzaklaştırıldığında c) İşaretleyici fanusa yaklaştırıldığında (Resim [22] numaralı kaynaktan alınmıştır)

Bu yöntem daha sonraları Vanacken ve diğerleri [24] tarafından elle tutulan bir nesne yardımıyla elin konumunu 3B ortama yansıtarak çalışacak biçimde güncellenmiştir. Bunun yanı sıra yöntem, elin sanal ortamda bulunduğu konuma ışın gönderildikten sonra algoritma prensibine göre seçim yapabilecek şekilde düzenlenmiştir.

2.2.3. Squad Yöntemi

2B ortamlarda nesne sayısının arttığı durumlarda daha kolay nesne seçimini sağlamak amacıyla geliştirilmiştir. Elle tutulan bir alet yardımıyla iki boyutlu bir ekran üzerinde seçilen çember biçiminde bir alanın içinde bulunan tüm nesnelerin farklı bir arayüzde görüntülenmesi ve daha küçük parçalara ayrılarak seçilmesi esasına dayanır [25]. Çember biçiminde olan alandaki ilk seçme işleminden sonra ekranın tamamı Şekil 7’de görüldüğü gibi çember biçiminde olan alandaki nesnelerin farklı gruplar halinde gruplandırılmış görüntüsü ile doldurulur. Her seferinde bir alt grup seçilerek bu işlem tekrarlanarak devam ettirilir. Birkaç adımdan sonra seçme işlemi tamamlanır. Nesne az yoğun ortamlar için

gereksiz seçme adımları getiren bu yöntem nesne yoğun ortamlardaki seçme karışıklığının önüne geçmiştir.



Şekil 7 Squad yöntemi nesne seçim adımları (Resimler [25] numaralı kaynaktan alınmıştır)

Resimler sırasıyla seçilmek istenilen nesnenin bulunduğu çember biçiminde bir alanın seçilmesi ve bu alandaki nesnelerin farklı bir arayüzde gruplar halinde sınıflandırılması işlemlerini belirtmektedir.

2.3. 3B Ortamlarda El Serbest Seçme İşlemi

El serbest etkileşim üzerine çalışan Guimbretiere ve Nguyen [27] yakın mesafeden ilk nesil Kinect sensör kullanarak işaretli menü seçimi üzerine çalışmışlar ve iki el kullanılarak uygulanan bir seçme tekniği önermişlerdir. El serbest 3B etkileşim üzerinde önemli bir aşama kaydetmelerine rağmen bu yöntem kameranın belirli açılarda ayarlanmasını gerektirdiğinden dolayı çok fazla tercih edilmemiştir.

Cashion [28] nesne 3B ortamlarda nesne yoğun ortamlardaki seçme problemini aşmak için 'Zoom' ve 'Expand' hibrit yöntemlerini önermiştir. 'Zoom' yöntemi nesne yoğun ortamların karmaşıklığını azaltarak seçim yaptırmayı hedeflerken 'Expand' yöntemi belirlenen alandaki nesnelerin tümünü ekrana dağıtarak seçme işlemini kolaylaştırmıştır. Ayrıca önerdiği yöntemlerin seçme süresi, kullanım kolaylığı ve kesinlik gibi alanlarda 'Ray Cast' ve 'Squad' yöntemlerinden daha avantajlı olduğunu göstermişlerdir.

Kinect kamera kullanılarak kullanıcı etkileşim sağlanan birçok oyun geliştirilmiştir [29]. Geliştirilen bu oyunlarda seçme işleminin yanı sıra oynanan oyunun ihtiyacına göre farklı etkileşim teknikleri de kullanılmıştır. Kinect kamera kullanılarak geliştirilen oyunlar etkileşim performansı yüksek olduğundan geniş oyuncu kitleleri tarafından tercih edilmektedir.

Bu çalışma için Argelageut ve Andujar [30]'ın tarihçesinden bahsettiği birçok nesne seçme yöntemi değerlendirilmiş ve bunların içerisinde Kinect sensör ile kullanılabilir el serbest etkileşime ve 3B ortama en uygun olanlarına karar verilmiştir. Değerlendirmelerimiz sonucunda 'Bubble Cursor', 'Depth Ray' ve 'Squad' yöntemlerinin diğerlerinden daha uygun olacağına karar verilmiştir.



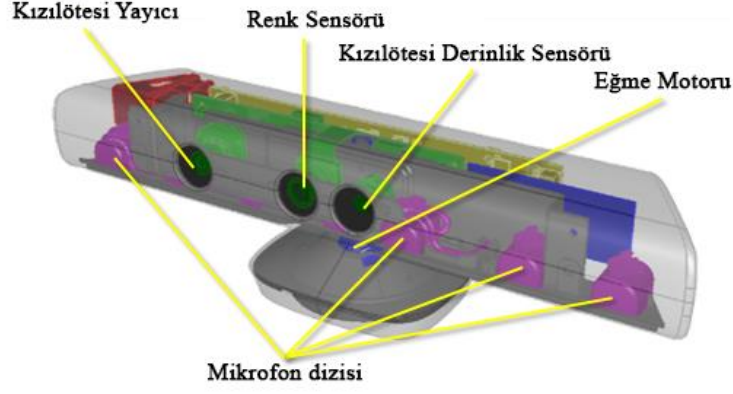
3. ÖN BİLGİ

3.1. Kinect Sensör

3.1.1. Genel Özellikler

Kinect, Xbox360 ve Xbox One oyun konsolları ve Windows bilgisayarlar için Microsoft tarafından geliştirilen hareket algılayan bir girdi aracıdır [31]. Araç insanların doğal kullanıcı hareketleri ve ses komutları aracılığıyla, bilgisayar ya da oyun konsolu ile etkileşim yapabilmesini sağlar. Xbox 360 oyun konsolu için geliştirilen ilk nesil Kinect sensör 2010 yılında tanıtılmıştır. Windows işletim sistemi için olan sürümü ise ilk olarak 2012 yılında ortaya çıkmıştır. 2014 yılında ise Windows ortamlar için yeni sensör kullanılarak geliştirilen ikinci nesil Kinect sensör ortaya çıkmıştır. Ayrıca 2011 yılında geliştiricilerin Kinect kullanarak C++/CLI, C# ya da Visual Basic .NET programlama dilleri üzerinde geliştirme yapabilmelerine olanak sağlamak amacıyla ilk Kinect SDK yayınlanmıştır. Ayrıca geliştiricilerin sensör özelliklerini daha verimli kullanabilmelerini sağlamak amacıyla Kinect SDK yeni sürümleri ortaya çıkarılmaktadır [32].

Kinect sensör, sensörün kullanılacağı ekranın alt ya da üst kısmına yerleştirilebilmesi amacıyla yatay bir taban üzerine yerleştirilmiştir. Araç bir RGB kamera, kızılötesi yayıcı, kızılötesi derinlik sensörü ve bir dizi mikrofonun birbirleri ile entegrasyonu yapılarak ortaya çıkarılmıştır [33]. Kinect sensör öğeleri Şekil 8'de gösterilmiştir. Bu sayede 3B hareket yakalama, yüz ve ses tanıma işlemlerinin gerçekleştirilmesinde kullanılabilir. Sensörde bulunan RGB kamera üç kanal verisini 1280x960 çözünürlükte yakalar ve bu sayede renk verisinin kullanılabilmesine imkan verir. Entegre edilen diğer parçalardan kızılötesi yayıcı, ışık ışınları yayarken; kızılötesi derinlik sensörü, kızılötesi ışınları yakalar. Yakalanan bu ışınlar derinlik verisine çevrilir ve bu sayede nesnelerin sensöre olan uzaklıkları belirlenir. Bunun yanında kullanılan kızılötesi ışık ışınları sayesinde görüntü verileri her tür ışık altında elde edilebilmektedir. Sensörde bulunan bir diğer parça grubu olan mikrofon dizisi ise dört farklı mikrofondan oluşur ve bu sayede ortamdaki ses verisinin elde edilmesinin yanında sesin geldiği konumun belirlenmesinde de kullanılabilir. Ayrıca aracın en alt kısmında duruş açısının ayarlanabilmesi için eğme motoru bulunur. Bu sayede en uygun görüş açısı kolaylıkla ayarlanabilmektedir.



Şekil 8 Kinect sensör parçaları (Resim [33] numaralı kaynaktan alınıp düzenlenmiştir)

Yeni nesil kameranın ortaya çıkmasıyla birlikte donanımsal özellikler artış göstermiş ve bu sayede Kinect sensör daha fazla alanda kullanılmaya başlanmıştır. Yeni nesil Kinect kamera aşağıda belirtilen donanım özelliklerine sahiptir [34].

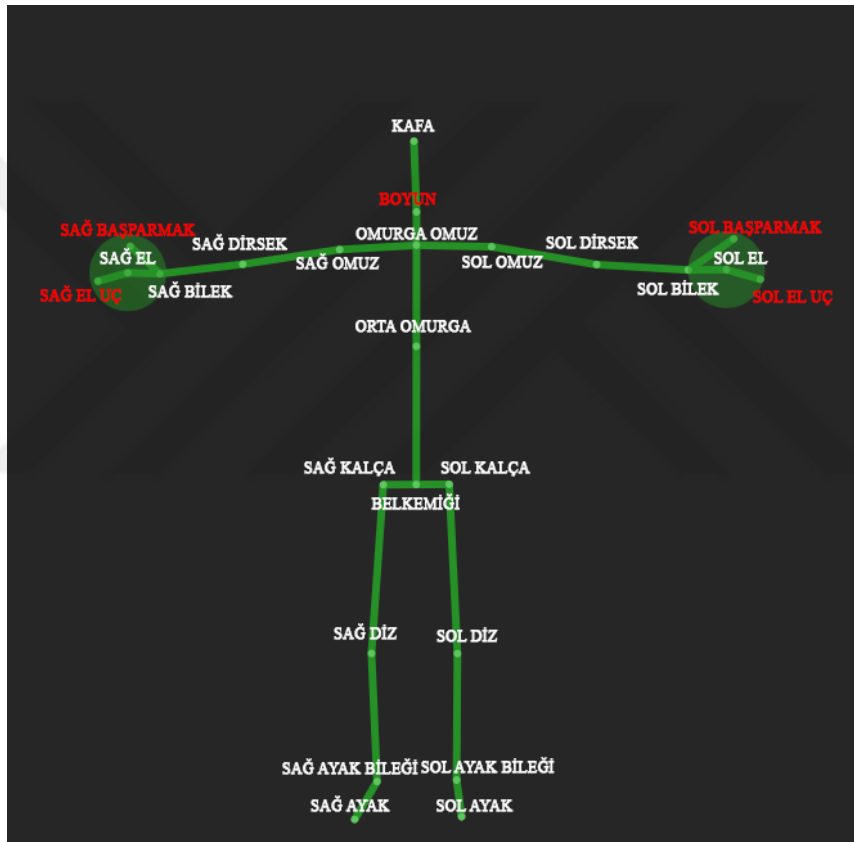
- 6 serbestlikte(DOF) 6 insan iskeletini ve her bir iskelet için 25 eklemi izleyebilir.
- 4.5 metre derinlikteki küçük nesnelere dahi algılayabilir.
- 30 Hz ve 1080p kalitedeki görüntüyü ekrana aktarabilir.
- Yeni aktif kızılötesi yeteneği sayesinde ışıktan bağımsız olarak çalışabilir.
- Geniş görüş açısı ile takip ve tanıma yapabilir.

Yeni sürüm SDK'lar ses ve yüz tanıma, hareket yakalama özelliklerindeki iyileştirmelerin haricinde, Windows mağaza ve Unity oyun motoru desteği sağlamaktadır. Ayrıca geliştiricilerin SDK'yı daha hızlı öğrenebilmeleri amacıyla Kinect Studio uygulaması geliştirilmiş ve bu uygulama içerisinde birçok farklı Kinect uygulaması kaynak kodları ile beraber paylaşılmıştır [35].

3.1.2. Kinect Sensör Hareket Yakalama

Kinect sensörlerden alınan eklem koordinatları sayesinde birden fazla insan hareketi yakalanabilmektedir. Kinect SDK aracılığıyla geliştirilen uygulamaya aktarılabilen kameranın bulunduğu yere göre alınan üç boyutlu eklem koordinatları, geliştirilen oyun ya da uygulamada işlenerek kullanılabilir. Kinect ilk nesil sensörlerden 20 farklı eklem koordinatı alınabiliyorken yeni nesil Kinect sensörlerden 25 farklı eklem koordinatı alınabilmektedir. Bu güncelleme ile Şekil 9'da görülebildiği gibi ilk paylaşılan SDK'lardan her bir el için bilek ve bir koordinat alınabiliyorken yeni SDK'dan bilek ve 3 koordinat

alınabilmektedir. Bu sayede kullanıcı el hareketleri daha kolay ve kesin bir biçimde işlenebilmektedir. İlk nesil Kinect sensörler kullanılarak geliştirilen oyun ve uygulamalar el durumunu yakalayabilmek için SDK'dan paylaşılan iki konum arası uzaklığı ölçüyor ve buna göre elin durumuna karar veriyorlardı ancak bu kadar az veriyle yapılan bu değerlendirme çok fazla hatalı durum içeriyordu. Yeni nesil Kinect kameralarla geliştirilen oyun ve uygulamalar ise dört farklı konumun birbirlerine göre uzaklığını ölçerek birçok farklı el durumuna karar verebilmektedir. İlk nesil sensörden alınması mümkün olmayan el durumlarının yeni nesil sensörden alınabilmesinin yanı sıra elin açık ya da kapalı olduğu bilgisi yeni nesil sensörlerden daha kesin bir biçimde alınabilmektedir.



Şekil 9 Kinect Eklem Türleri Listesi(Kırmızı renkle yazılan eklemler Kinect ilk nesil sensörlerden alınamayıp yeni nesil Kinect sensörlerden alınabilen eklemleri belirtmektedir)

3.2. Unity Oyun Motoru

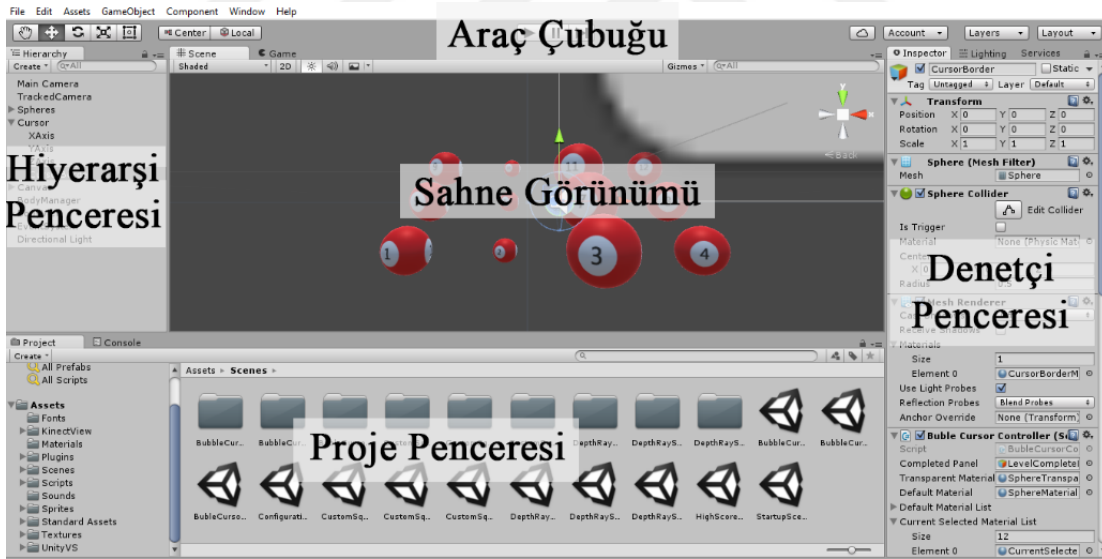
3.2.1. Genel Özellikler

Unity 2B ve 3B oyunlar ve uygulamalar oluşturmaya yardımcı olan çapraz platform bir oyun motorudur. Geliştirilen oyun ya da uygulamaların mobil cihazlar, web tarayıcılar, bilgisayarlar ve konsollarda çalışmasına olanak verir. Farklı platformlarda çalışabilen oyun

ya da uygulama çıkarılabilmesine izin vermek amacıyla Unity oyun motoru OpenGL, Direct 3D ve OpenGL ES gibi grafik API'lerini kullanmaktadır. En son sürüm Unity, bilgisayarlarda: Windows, Windows Store, Mac ve Linux; sanal gerçeklik ve artırılmış gerçeklik cihazlarında: Oculus Rift, Google Cardboard, Steam VR, Playstation VR, Gear VR ve Microsoft Hololens; konsollarda: Playstation 4, Playstation vita, Xbox one, Xbox 360, Wii U ve Nintendo 3Ds; web de: WebGL; akıllı televizyonlarda: Android TV, Samsung Smart TV ve TvOS oyun ve uygulamaları geliştirmeye izin vermektedir [36].

3.2.2. Unity ile Oyun ya da Uygulama Geliştirme

Unity, 2B ve 3B oyun ya da uygulama geliştirmeyi kolaylaştıran düzenleyici grafik arayüzü desteği vermektedir. Bu ana düzenleyici pencerede geliştiricinin isteğine göre ayarlanabilir birçok sekme mevcuttur. Bu sayede geliştirilen farklı projelere ve geliştiricilere en uygun ayarlar kolayca yapılabilir. Ana düzenleyici pencerede araç çubuğu, hiyerarşi penceresi, sahne görünümü, denetçi penceresi ve proje penceresi bulunur. Örnek bir ana düzenleyici pencere Şekil 10'da görülebilir.



Şekil 10 Unity Ana Düzenleyici Pencere

Araç çubuğu, en temel çalışma özelliklerine erişimi sağlar. Çubuğun sol tarafında sahne görünümündeki nesnelerin konumunu, ölçeğini ve rotasyonu değiştirmeyi sağlayan basit araçlar bulunur. Çubuğun orta kısmında ise oyun görünümünü başlatmayı, adım adım ilerletmeyi ve durdurmaya sağlayan araçlar bulunur. Çubuğun sağ tarafında ise Unity bulut servislerine ve Unity hesabına erişim sağlayan araçlar, katman gizlilik menüsü ve yerleşim

menüsü bulunur. Hiyerarşi penceresi ise sahne görünümünde bulunan tüm nesnelerin hiyerarşik bir biçimde tutulduğu penceredir. Hiyerarşi penceresi sahne görünümüne bağlı bir penceredir. Denetçi penceresi ise sahnedeki nesnelere seçili olanlarının tüm özelliklerinin düzenlenebildiği penceredir. Sahne görünümü ise sahnenin görsel bir biçimde düzenlenebilmesini sağlar. Sahne görünümü geliştirilen projenin özelliğine göre 2B ya da 3B görünümde çalıştırılabilir. Proje penceresi ise projedeki tüm varlıkların görüntülenebilmesine, yeni varlık eklenmesine ve varlıkların silinebilmesine imkân veren penceredir [37].

Unity oyun motoru oyun ve uygulamaların en önemli özelliklerinden olan ışık, kamera, materyaller, gölgeler, dokular ve bunlar gibi daha birçok grafik özelliklerini kolay bir biçimde yönetme imkânı verir. Bunun yanında fizik motoru sayesinde nesnelerin birbirleri ile çarpışma, temas etme gibi durumlarında gerçeğe en yakın davranışın belirlenmesinde geliştiricilere büyük kolaylık sağlamaktadır. Geliştiriciler için kolay kullanılabilir betik desteği sağlaması da Unity oyun motorunun kolaylaştırıcı özelliklerindedir. C# ve Javascript betikleri ile geliştirme yapmaya imkân vermektedir. Ayrıca animasyon, ses, kullanıcı arayüzü gibi alanlarda da pek çok kolaylık sağlamaktadır. Tüm bu özellikleri ile Unity bir oyun ya da uygulama geliştirecek kişiler için en önde gelen oyun motorlarından biri olarak kabul görmektedir.

4. DENEYSEL OYUN GELİŞTİRME

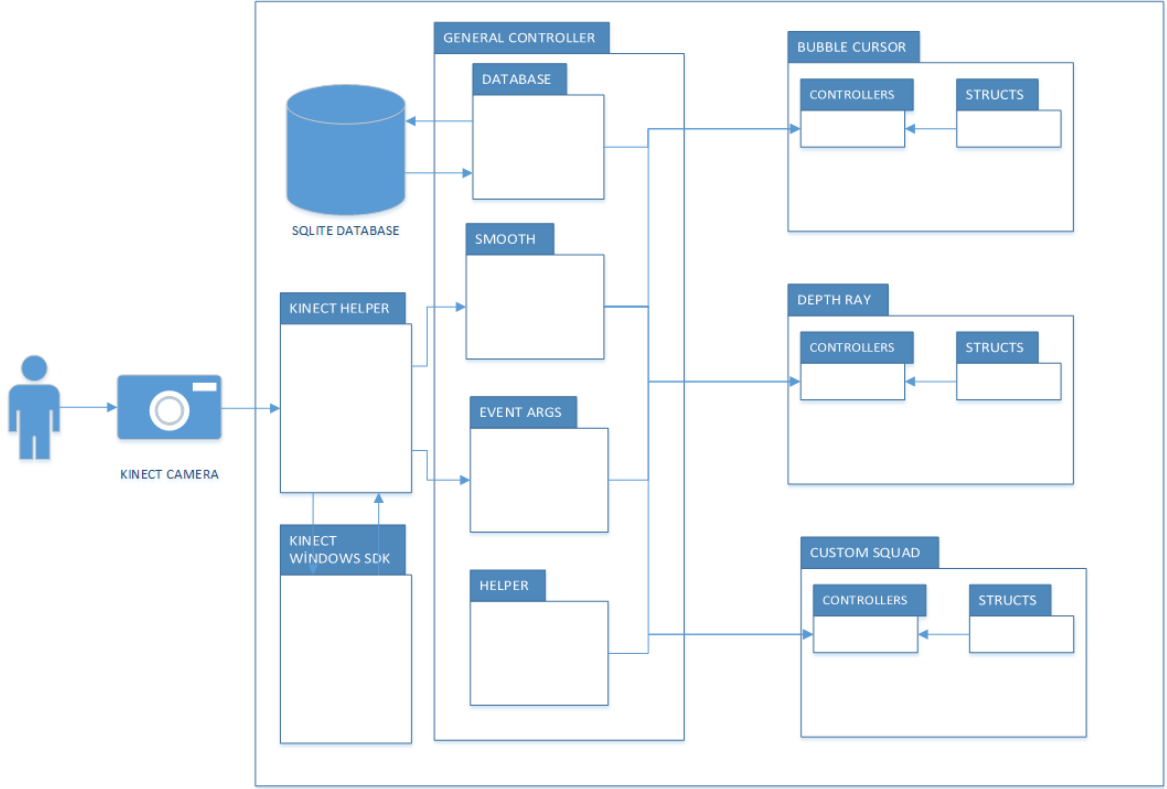
3B el serbest nesne etkileşimi çalışmalarında derinlik algılayıcı olarak Kinect yeni nesil kamera kullanılmış ve yöntemler bu kamera ile çalışacak biçimde geliştirilmiştir. Kullanıcıların üç boyutlu ortamlarda daha kolay ve hızlı bir biçimde nesnelere seçebilmelerini sağlamak amacıyla, daha önceleri farklı ortamlarda geliştirilen yöntemler yeni ortama uyum sağlayacak biçimde güncellenmiştir. Ardından uygulanan yöntemlerin birbirleri ile kıyaslanabilmeleri için aynı çevreler her bir yönteme uygulanmıştır. Geliştirilen yöntemlerin birbirleri ile kıyaslarında üç farklı durum ele alınmıştır. Bu bağlamda seçilecek nesnelere az yoğun olduğu ortam, çok yoğun olduğu ortam ve hareketli oldukları ortamlar oluşturulmuştur. Oluşturulan bu üç boyutlu ortamlar, yöntemlerin birbirinden farklı ortamlarda hız ve kullanım kolaylığı açısından kıyaslanmasında kullanılmıştır.

3B nesne etkileşimi konusunda bahsedilenlerden farklı olarak yenilik getiren pek çok çalışma mevcuttur. Argelageut ve Andujar [30] bu çalışmalar üzerine odaklanmış ve bunların birçoğunu tek bir makalede toplamışlardır. Bu yöntemlerden kullanımı en kolay ve kurulum gerektirmeyen yöntem olan el serbest yöntemler üzerinde geliştirme yapılmış ve farklı seviyelerden oluşan küçük bir oyun ortaya çıkarılmıştır. Bu oyunda 3B takip araçları ile geliştirilen 3 farklı yöntem, el serbest etkileşimde kullanılacak biçimde güncellenmiştir. Ayrıca daha kolay, hızlı ve kesin bir biçimde seçim yapılabilmesini sağlamak amacıyla yöntemlerin pek çok bölümü değiştirilmiş ve bu yöntemler kıyaslanmıştır. Bu kıyaslama yapılırken her bir yöntem için az yoğun, çok yoğun ve hareketli ortamlardaki sonuçlar incelenmiştir.

4.1. Deneysel Oyun Tasarımı

Kullanıcının el hareketlerinin algılanması için derinlik algılayıcı olarak Kinect ikinci nesil kamera seçilmiştir. Windows işletim sistemi için geliştirilen SDK eklem konumlarının 3 boyutlu olarak uygulamaya aktarılmasında kullanılmıştır. Bu nedenle uygulamanın çalışacağı işletim sistemi olarak Windows işletim sistemi tercih edilmiştir. Ayrıca seçilen ve geliştirilen yöntemlerin tümü bir oyun çatısı altında toplandığı için uygulamanın bir 3B oyun motoru üzerinde yapılması kararlaştırılmıştır. 3B oyun motorları içerisinde Unity 3B geliştirme kolaylığı, performansı ve yetenekleri açısından diğer alternatiflerden daha üstün olduğundan uygulamanın bu platformda yapılması kararlaştırılmıştır.

Geliştirilen sistemde eklem verileri Kinect kameradan SDK yardımıyla alındıktan sonra işlenerek el konumunu belirlemede ve seçme hareketini algılamada kullanılmıştır. Şekil 11’de görüldüğü gibi Kinect yardımcı kütüphanesi aracılığıyla alınan eklem verileri hareketin daha az titreşimli olarak elde edilebilmesi amacıyla yumuşatma (smooth) paketinden geçirildikten sonra kullanılmıştır.



Şekil 11 3B Nesne Seçme oyunu genel sistem mimarisi

Yumuşatma paketinden geçen eklem verileri o anda açık olan seviyeye bağlı olarak ilgili seviyenin kontrol paketinde işlenmiş ve ekranda imleç hareketi gerçekleştirilmiştir. Sistemde her seviye için farklı bir paket bulunmaktadır. Nesne seçim hareketinin algılanması ise olay (event) paketi yardımıyla gerçekleştirilmiştir. Bu paketten gönderilen seçilme komutu, ilgili yöntemin kontrol paketinde işlenerek nesne seçimi tamamlanmıştır. Ayrıca tüm yapılan seçme işlemlerinin veri olarak tutulabilmesi için veri tabanı paketi kullanılmıştır.

4.2. 3B Ortamlarda Kinect ile Kullanıcı Etkileşimi

3B ortamlarda insan bilgisayar etkileşiminin sağlanması işleminde, Kinect sensörlerin sağladığı eklem koordinatları büyük kolaylık sağlamaktadır. Geliştirilen oyunda imlecin hareket ettirilebilmesi için sağ el koordinatı temel alınmıştır. El koordinatının izlenmesi ve ekrana aktarılması için Kinect sensörden alınan omuz omurga kesişim eklemi, sağ omuz eklemi, sağ dirsek eklemi ve sağ el bilek eklemi koordinatları kullanılmıştır. Sol elin kullanılması gereken yöntemde ise sol el bilek eklemi koordinatları kullanılmıştır. Kullanım kolaylığının artırılması için omuz omurga eklemi merkez kabul edilmiştir ve imleç koordinatı el bilek eklemi koordinatının omuz omurga eklemine göre üç boyutlu uzaydaki konumuna göre belirlenmiştir. Bunun yanı sıra üç boyutlu seçim alanındaki nesnelerin konumu neresi olursa olsun kolayca seçilebilmesi için omuz el bileği mesafesi hesaplanmıştır. Bu mesafenin hesaplanması; omuz dirsek eklemleri arası uzaklık ve dirsek el bileği eklemi arası uzaklık toplanarak gerçekleştirilir. Hesaplanan bu uzunluk Poupyrev ve diğerlerinin [38] önerdiği ‘Go-go’ yönteminde olduğu gibi doğrusal olmayan eşleştirme ile ekran genişliğinin yarısı ile eşleştirilir ve kullanıcının sağ eli yana doğru tamamen açıldığında imleç ekrandaki en sağ noktaya yerleştirilir. Bu sayede köşelerdeki nesnelere kolaylıkla seçilebilmesi sağlanır. Bunun yanı sıra hesaplanan uzunluk sanal ortamın derinliği ile eşleştirilerek en arkada kalan nesnelere zorlanmadan seçilebilmesi sağlanır.

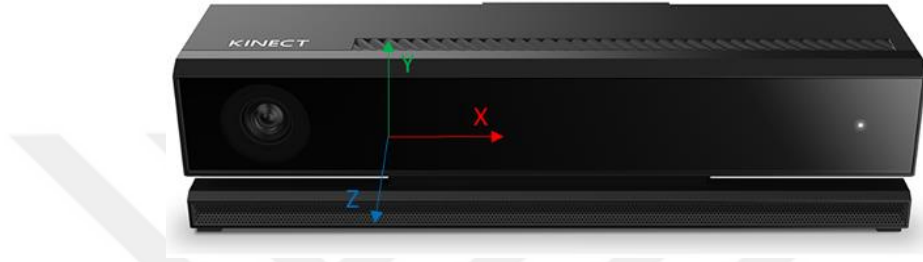
Nesne seçme işleminde ise Dam ve diğerlerinin [39] çalışmasında bahsettiği üzerinde durma, bastırma ve sabit tutma yöntemleri değerlendirilmiş ve bu yöntemlerin seçme süresini düşüreceği görülmüştür. Bu nedenle elin açık durumdayken kapatılması aracılığıyla nesne seçiminin yapılması yöntemi kullanılmıştır. Ayrıca imlecin hareket ettirildiği el kullanılarak seçme işleminin yapılması ile seçim kolaylığı ve hassasiyetinin sağlanması amaçlanmıştır.

4.3. Geliştirme

4.3.1. Eklem Koordinatlarının Alınması

Oyunun çalıştığı Windows işletim sisteminde kamera ile entegrasyonun kurulabilmesi için Kinect 2.0.1410 SDK’sı kurulu olmalıdır. Kullanılan SDK bir önceki sürümlerde olmayan Windows mağaza veya Unity oyun motoru entegrasyonlarını sağlamanın yanı sıra geliştirilmiş el hareketi algılama sistemi sayesinde daha verimli bir etkileşim sağlamaktadır.

Kinect sensörün üzerinde bulunan derinlik sensörü sayesinde veriler Şekil 12’de görüldüğü gibi x, y, z koordinat düzleminde elde edilir. Elde edilen bu verilerin oyuna aktarılıp kullanıcılar ile etkileşimli hale getirilmesi gerekmektedir. Bu entegrasyonun sağlanabilmesi için geliştirme yapılan oyun motoru Unity ile iletişimin kolaylaşması amacıyla Microsoft tarafından sağlanan Unity paketleri kullanılmıştır. Bu paketler C# betikleri ile yazılmış ve direkt olarak herhangi bir Unity betiği ile kolayca kullanılabilen sınıflardan oluşmaktadır. SDK tarafından sürekli olarak gönderilen eklem koordinat verileri betiklerinin sürekli çağrılan yöntemleri içerisinde alınarak etkileşim sağlanmıştır.



Şekil 12 Kinect Kamera koordinat sistemi (Resim [40] numaralı kaynaktan alınmıştır)

4.3.2. Koordinat Sistemi Çevriminin Yapılması

Eklem koordinatlarının SDK’dan alındıktan sonra ekran koordinatları ile eşleştirilmesi gerekmektedir. Bu amaçla eklem verileri öncelikle ekran koordinatlarına çevrilir. Eşitlik 1’de x ekseninde koordinat çevrimi gösterilmiştir. Eşitlikte bulunan C_x eşleştirilmiş x koordinatını, PW ekranın piksel olarak genişliğini, H_x el konumunun x değerini, S_x omuz konumunun x değerini ve D ise el ve omuz arası üç boyutlu uzaklığını belirtmektedir. Benzer biçimde Eşitlik 2’de de y ekseninde koordinat çevrimi gösterilmiştir. Eşitlik 2’de bulunan PH, ekranın piksel olarak yüksekliğini belirtmektedir.

$$C_x = \frac{PW}{2} * \left(1 + \frac{H_x - S_x}{D}\right) \quad (1)$$

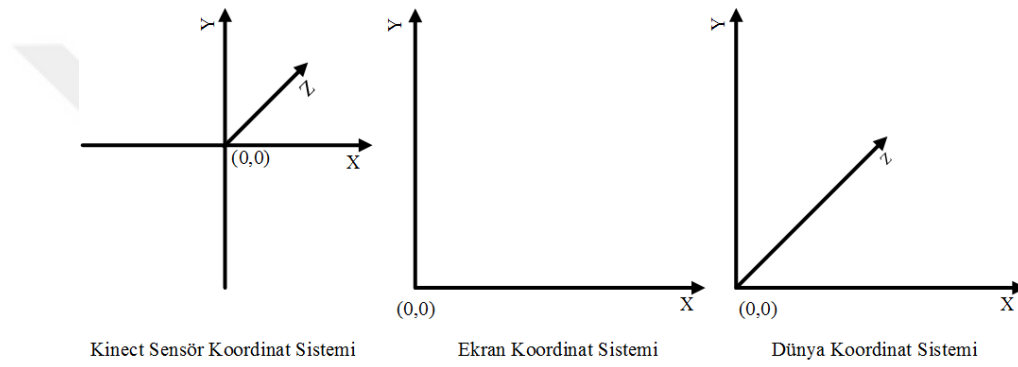
$$C_y = \frac{PH}{2} * \left(1 + \frac{H_y - S_y}{D}\right) \quad (2)$$

Z ekseninde eklem verisi piksel çevrimi, x ve y eksenleri üzerinde yapılan çevrimden daha farklı olmaktadır. Eklem koordinat sisteminden ekran koordinat sistemine z eksenine

üzerindeki geçiş Eşitlik 3’de gösterilmiştir. Eşitlikte SP_z belkemiği z koordinatını, H_z el konumu z koordinatını ve D ise el omuz arası üç boyutlu uzaklığı belirtmektedir. Eşitlikteki ZD değeri ise sahnede kullanılacak alanın derinliğini ifade etmektedir.

$$C_z = \frac{SP_z - H_z}{D} * ZD \quad (3)$$

Kinect koordinat sistemine çevrilen noktalar, deneysel 3B oyun sahnesinde çalışılabilmesi amacıyla dünya koordinat sistemine çevrilmiştir. Çevrim yapılan koordinat sistemleri Şekil 13’te görülebilir.



Şekil 13 Kinect sensör, ekran ve dünya koordinat sistemleri

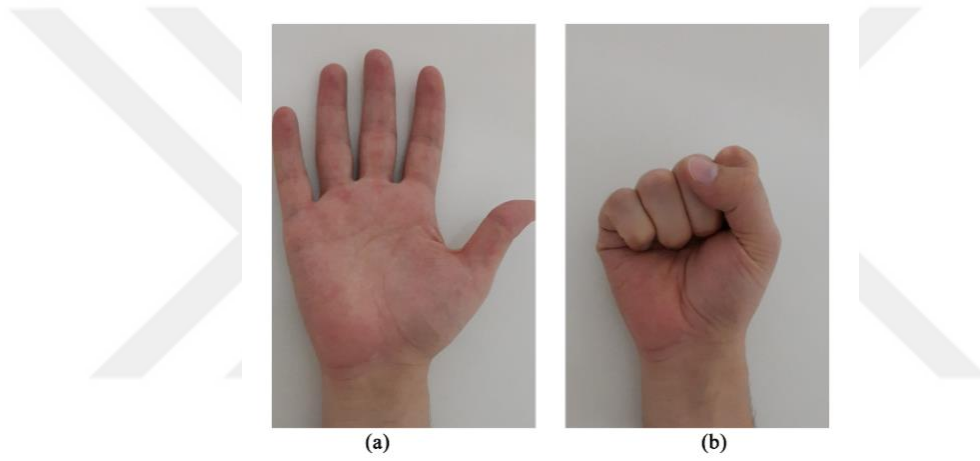
4.3.3. Eklem Konumu Yumuşatma

Kameradan alınan eklem verileri kullanıcının kameradan uzaklığına bağlı olarak çok fazla titreşim hareketi göstermektedir. Bunun yanı sıra SDK, eklemleri izlemeye çalışırken yeterince düzenli çalışmamakta ve bazı durumlarda eklemler birbirleri ile karışmaktadır. Bu nedenlerden dolayı SDK’den alınan koordinatlar direkt olarak kullanılamamaktadır. Bu durumun önüne geçebilmek için aritmetik ortalama ile yumuşatma işlemi kullanılmaktadır. Her bir eklem için en son gelen verisi, kendisinden önce belirli bir sayıdaki veri ile birlikte hesaplanır ve bu sayede hareketteki gürültünün önüne geçilmiş olur. Hesaplama tüm koordinatların üç boyutlu olarak toplanmasının ardından toplam değerinin eleman sayısına bölümü ile gerçekleştirilmiştir. Geçmiş hareketler eklem koordinat penceresinde tutularak bu işlem gerçekleştirilir. Hareket verisinin bu şekilde işlendikten sonra kullanılması her ne kadar titreşimi azaltıp kullanıcı etkileşimini iyileştirirse de hareketin oyunda yansımalarını bir

süre geciktirmektedir. Bu sorunun en aza indirilmesi amacıyla eklem koordinat penceresi boyutu 25 adet veri ile sınırlandırılmıştır.

4.3.4. El Serbest Seçme İşlemi

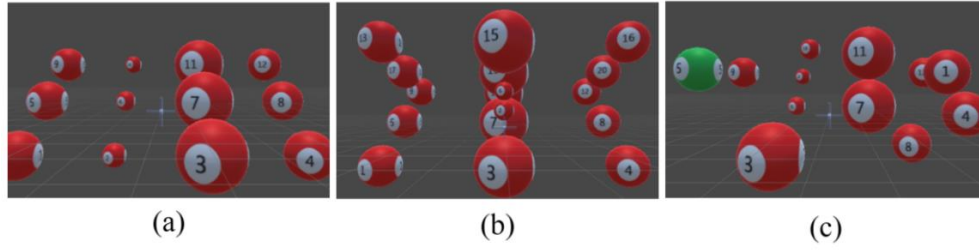
Kinect kamera aracılığıyla seçme işlemi yapılabilmesi için izlenen eklem verilerinin belirlenen bir hareketi yaptığı durumun yakalanması yolu izlenmelidir. Geliştirilen deneysel oyunda sağ el hali hazırda imleci hareket ettirmek için kullanıldığından; seçme işleminin aynı el aracılığıyla ve el hareket ettirilmeden yapılması gerekliliği doğmuştur. Bu nedenle seçme işleminin yapılabilmesi için Şekil 14’te görüldüğü gibi kullanıcının eli açıkken yumruk yapması yoluyla seçme işleminin algılanması tercih edilmiştir.



Şekil 14 Nesne seçme işleminde kullanılan el pozisyonları a) El açık pozisyon herhangi bir seçme işlemi yapılmak istenmediği durumlarda kullanılır b) El yumruk pozisyonu el açık pozisyonda iken nesne seçimi yapılmak istendiğinde kullanılır

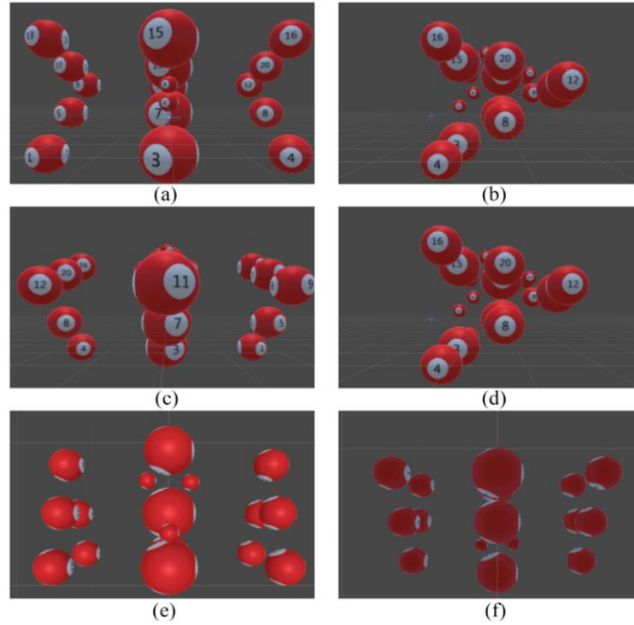
Kinect Unity paketi aracılığıyla el konumu alınabilmesine rağmen çok sık el durumu verisi gelmesi ve gelen verilerin her zaman doğru durumu vermemesi, kullanıcı etkileşiminde problemlere yol açmıştır. Bu sorunu gidermek amacıyla el durumlarının tutulduğu bir pencere veri yapısı oluşturulmuş ve elin durumu sürekli olarak bu penceredeki durumların çoğunluğu olarak kabul edilmiştir. Bu yöntem sayesinde, kullanıcıların ilgili nesneyi kolay bir biçimde seçebilmesi sağlanmıştır. Son olarak seçim yapıldığı bilgisi, oluşturulan özel sınıf kullanılarak ve bu sınıfa seçme dinleyicisi atanarak sağlanmıştır. Herhangi bir seçme geribildirimi geldiği anda aktif sahnedeki sürekli dinleyen kısım aktifleştirildikten sonra gerekli işlemler tamamlanmıştır.

4.3.5. Ortamların Geliştirilmesi



Şekil 15 a) Nesne az yoğun ortam b) Nesne çok yoğun ortam c) Nesne hareketli ortam

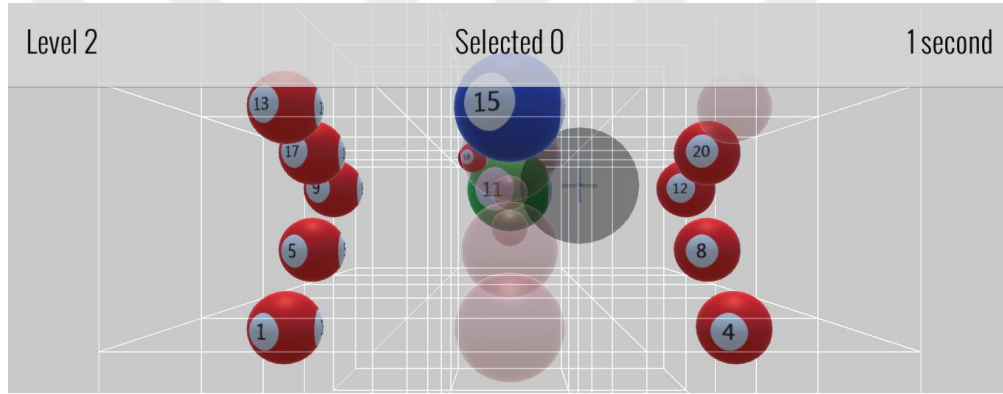
Geliştirilen deneysel oyunda farklı algoritmaların birbirinden farklı ortamlarda ele alınmalarını sağlayabilmek için seviye bazlı bir yöntem kullanılmıştır. Her bir yöntemin az yoğun, yoğun ve nesne hareketli ortamlardaki özelliklerini inceleyebilmek için Şekil 15’te görüldüğü gibi üç farklı seviye oluşturulmuştur. Bu sayede yöntemlerin birbirleri ile kıyaslanırken farklı yöntemlere göre kıyaslanarak ortam bazlı olarak karşılaştırılabilmesi sağlanmıştır. Nesne az yoğun ve nesne hareketli ortamlara 12, nesne çok yoğun ortama ise 18 seçilebilir nesne yerleştirilmiştir. Nesne hareketli ortamda, sanal ortamdaki nesnelerin belirlenen koordinatlar arasında sabit hızlarla hareket etmeleri sağlanmıştır. Ayrıca her bir seviyede farklı boyutlarda nesneler kullanılmış ve bu sayede nesne büyüklüğünün seçme işlemine etkisinin belirlenmesi amaçlanmıştır. Nesne yoğun ortamın altı farklı açıdan görünüşünün verildiği Şekil 16’da nesne büyüklüklerinin farkı net bir biçimde görülebilir.



Şekil 16 Nesne yoğun ortam farklı açılardan görünüşleri a)Önden görünüş b)Sağdan görünüş c)Arkadan görünüş d)Soldan görünüş e)Üstten görünüş f)Alttan görünüşü

4.3.6. Uyarlanmış Bubble Cursor Yöntemi

'Bubble Cursor' yönteminin uyguladığı algorithmda, Kinect kameradan alınan konum verisi yumuşatma filtresinden geçirildikten sonra ekrandaki imlecin konumu belirlenmiş ve 'Uyarlanmış 3B Bubble Cursor' yöntemi bu konum temel alınarak çalıştırılmıştır. Ayrıca imlecin etrafında, yöntemin çalışma prensibine göre boyutları belirlenen çember, üç boyutlu olarak gösterilmiştir. Bu çemberin büyüklüğüne göre en yakın nesne aktifleştirilip seçime hazır hale getirilmiştir. İmleç, ekranda gezdirilirken yöntemin çalışma prensibine göre o anda seçili olan nesne mavi renkle gösterilirken, seçilecek olan nesne yeşil renkle gösterilmiştir. Üst üste gelen ya da diğerinin arkasında kalan nesnelerin seçimini kolaylaştırmak amacıyla imlecin bulunduğu konumdan belirli mesafe uzaklıktaki nesneler yarı geçirgen hale getirilmiştir. Yöntemin çalıştığı örnek bir seviye ekranı Şekil 17'de görülebilir.



Şekil 17 Uyarlanmış 3B Bubble Cursor Yöntemi seviye ekranı

Yöntemin Şekil 17'de görüldüğü gibi çalışır hale getirilebilmesi için, biri seçme işlemini diğeri ise kalan tüm ayarlamaları kontrol eden iki fonksiyon kullanılmıştır. Bu fonksiyonların sözde kod yazımı aşağıda verilmiştir.

```
void function Clicked{
    if( seçili nesne = sıradaki seçilecek nesne){
        seçim bilgisini veri tabanına yaz;
        ekrandaki toplanan alanını güncelle;
        seçim sesini aktifleştir;
        bir sonraki seçilecek nesnenin rengini değiştir ya da seviye
        sonu panelini aktifleştir;
    }
}

void function Update{
    for( ; ; ){
        if(durdurulmuş){
            dön;
        }
    }
}
```

```

imleç pozisyonunu al;
foreach( nesne : tüm seçilebilir nesnelere){
    imlecin pozisyonu ile nesne arası uzaklığı hesapla;
    hesaplanan değeri nesnenin kesişim uzaklığı ve kapsayan
    uzaklığı alanlarına ata;
}

foreach( nesne : tüm seçilebilir nesnelere){
    imlece pozisyonuna göre en yakın iki noktayı al;
    if (imleç pozisyonuna en yakın noktanın kapsayan
    uzaklığı <= imleç pozisyonuna ikinci en yakın noktanın
    kesişen uzaklığı){
        en yakın nesnenin rengini değiştir;
    }
    else{
        en yakın ikinci nesnenin rengini değiştir;
    }

    if( nesne != seçili nesne){
        if( nesnenin imlece uzaklığı < belirlenen
        aralık){
            nesneyi yarı geçirgen yap;
        }
        else if( nesne yarı geçirgen){
            nesnenin yarı geçirgenliğini kaldır;
            nesnenin rengini eski haline getir;
        }
    }
}

imlecin etrafında seçili nesnenin durumuna göre yarı geçirgen
bir daire oluştur;
süre alanını güncelle;
}
}

```

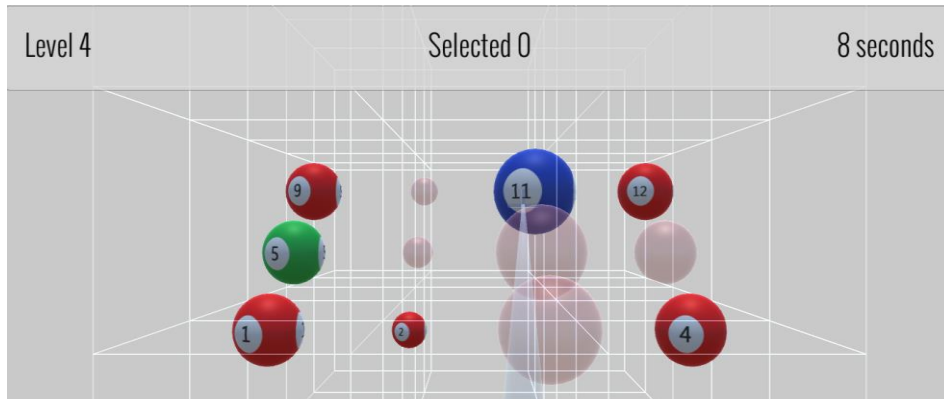
Nesnenin seçilme durumunu kontrol eden 'Clicked' fonksiyonu bir olay dinleyicisi olarak çalışmaktadır. Diğer fonksiyonlardan tamamen bağımsız olan bu fonksiyon, eklem koordinatlarının alındığı sınıfta el durumu açık durumdan kapalı duruma geçtiğinde tetiklenir. Ardından o anda sistemde aktif olarak seçili olan nesne, seçilecek nesnelere listesindeki sıradaki nesne ile karşılaştırılır. Bu karşılaştırma sonucunda her iki nesnede aynı nesne ise veri tabanına yazma, ekrandaki ilgili alanları güncelleme, seçim sesini aktifleştirme, bir sonraki seçilecek nesnenin rengini değiştirme ya da seviye sonu panelini aktifleştirme işlemleri yapılır.

Yöntemin diğer çalışma ayarlamalarını kontrol eden 'Update' fonksiyonu ise Unity tarafından her bir grafik çerçevesinde bir defa çalıştırılmaktadır. Bu sayede sahnede herhangi bir anda yapılmak istenilen tüm işlemler bu fonksiyonun içerisinde kolaylıkla yapılabilmektedir. 'Update' fonksiyonunda imleç pozisyonu hesaplandıktan sonra sahnede bulunan tüm seçilebilir nesnelere olan kapsayan ve kesişen uzaklıklar hesaplanır.

Ardından yöntemin çalışma prensibine uygun olarak imlece en yakın iki noktanın kapsayan ve kesişen uzaklıkları kıyaslanarak seçilecek nesne aktifleştirilir. Bunun yanı sıra belirlenen yarı geçirgenlik mesafesi içerisine giren tüm nesnelere yine bu fonksiyon içerisinde yarı geçirgen hale getirilir. Daha önceden rengi değiştirilmiş ya da yarı geçirgen hale getirilmiş olan nesnelerin eski hallerine getirilebilmeleri için gerekli olan kontrol ve işlemler de yine bu fonksiyon içerisinde gerçekleştirilir. Son olarak imlecin etrafında, seçilen nesneye göre yarıçapı belirlenen yarı geçirgen bir daire oluşturulup süre alanı güncellenerek işlemler tamamlanır.

4.3.7. Uyarlanmış Depth Ray Yöntemi

Geliştirilen uyarlanmış yöntemde, ışının gideceği nokta yumuşatma filtresinden geçen el konumu kullanılarak belirlenmektedir. Kameranın alt kısmında imlece doğru çizilen ışının üzerindeki nesnelere imlece en yakın olan aktifleştirilip seçme işlemine hazır hale getirilir. İmleç ekranda gezdirilirken yöntemin çalışma prensibine göre o anda seçili olan nesne, mavi renkle gösterilirken seçilecek olan nesne yeşil renkle gösterilmiştir. Üst üste gelen ya da diğerinin arkasında kalan nesnelerin seçimini kolaylaştırmak amacıyla gönderilen ışının üzerindeki ve imlecin belirli bir mesafe uzaklığındaki tüm nesnelere yarı geçirgen hale getirilmiştir. Yöntemin çalıştığı örnek bir seviye ekranı Şekil 18’de görülebilir.



Şekil 18 Uyarlanmış 3B Depth Ray Yöntemi seviye ekranı

Yöntemin Şekil 18’de görüldüğü gibi çalışır hale getirilebilmesi için biri seçme işlemi kontrol eden, diğeri ışının üzerinde kalan nesnelere bulan ve bir diğeri ise kalan tüm ayarlamaları kontrol eden üç yöntem kullanılmıştır. Bu yöntemlerin sözde kod yazımı aşağıda verilmiştir.

```

void function Clicked{
    if( seçili nesne = sıradaki seçilecek nesne){
        seçim bilgisini veri tabanına yaz;
        ekrandaki toplanan alanını güncelle;
        seçim sesini aktifleştir;
        bir sonraki seçilecek nesnenin rengini değiştir ya da seviye
sonu panelini aktifleştir;
    }
}

void function DepthRayRayController{
    for( ; ; ){
        kameradan imlece doğru bir ışın çiz;
        ışınla kesişen nesnelere bul ve bu nesnelere bir liste
oluştur.
    }
}

void function Update{
    for( ; ; ){
        if(durdurulmuş){
            dön;
        }
        imleç pozisyonunu al;
        foreach( nesne : tüm seçilebilir nesnelere){
            imleçin pozisyonu ile nesne arası uzaklığı hesapla;
            hesaplanan değeri nesnenin merkez uzaklık alanlarına
ata;
        }
        foreach( nesne : tüm seçilebilir nesnelere){
            ışının üzerinde bulunan nesnelere al ve ilgili
alanlarına bu bilgiyi ata;
        }

        foreach( nesne : tüm seçilebilir nesnelere){
            imleç pozisyonuna göre ışının üzerindeki en yakın
noktayı al;
            en yakın nesnenin rengini değiştir;

            if( nesne != seçili nesne){
                if( nesne ışının üzerinde || nesnenin imleç
uzaklığı < belirlenen aralık){
                    nesneyi yarı geçirgen yap;
                }
                else{
                    nesnenin yarı geçirgenliğini kaldır;
                    nesnenin rengini eski haline getir;
                }
            }
        }

        süre alanını güncelle;
    }
}

```

Nesnenin seçilme durumunu kontrol eden 'Clicked' fonksiyonu bir olay dinleyicisi olarak çalışmaktadır. Diğer fonksiyonlardan tamamen bağımsız olan bu fonksiyon, eklem

koordinatlarının alındığı sınıfta el durumu açık durumdan kapalı duruma geçtiğinde tetiklenir. Ardından o anda sistemde aktif olarak seçili olan nesne, seçilecek nesnelere listesindeki sıradaki nesne ile karşılaştırılır. Bu karşılaştırma sonucunda her iki nesnede aynı nesne ise veri tabanına yazma, ekrandaki ilgili alanları güncelleme, seçim sesini aktifleştirme, bir sonraki seçilecek nesnenin rengini değiştirme ya da seviye sonu panelini aktifleştirme işlemleri yapılır.

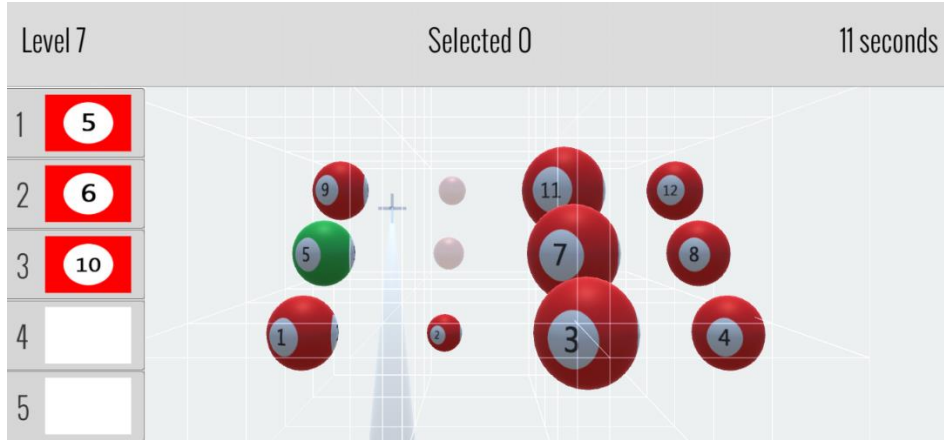
Gönderilen sanal ışının üzerinde bulunan nesnelere tespit eden 'DepthRayRayController' fonksiyonu ise Unity'nin fizik motorunu kullanarak gönderilen ışınla çarpışan tüm seçilebilir nesnelere bir listeye atar. Bu sayede ışının üzerinde bulunan nesnelere tespit edilerek üzerlerinde istenilen işlemler gerçekleştirilebilir.

Yöntemin diğer çalışma ayarlamalarını kontrol eden 'Update' fonksiyonu ise Unity tarafından her bir grafik çerçevesinde bir defa çalıştırılmaktadır. Bu sayede sahnede herhangi bir anda yapılmak istenilen tüm işlemler bu fonksiyonun içerisinde kolaylıkla yapılabilmektedir. 'Update' fonksiyonunda imleç pozisyonu hesaplandıktan sonra sahnede bulunan tüm seçilebilir nesnelere olan merkez uzaklık hesaplanır. Uzaklıklar hesaplandıktan sonra ışının üzerinde bulunan nesnelere bir listeye atılarak kullanıma hazır hale getirilir. Ardından yöntemin çalışma prensibine uygun olarak ışın üzerinde imleç pozisyonuna en yakın olan nesne aktifleştirilir. Bunun yanı sıra belirlenen yarı geçirgenlik mesafesi içerisine giren tüm nesnelere yine bu fonksiyon içerisinde yarı geçirgen hale getirilir. Daha önceden rengi değiştirilmiş ya da yarı geçirgen hale getirilmiş olan nesnelere eski hallerine getirilebilmeleri için gerekli olan kontrol ve işlemler de yine bu fonksiyon içerisinde gerçekleştirilir.

4.3.8. Uyarlanmış Hibrid Squad Yöntemi

Geliştirilmiş olan yöntemde hem 'Squad' yöntemi iki boyuttan üç boyuta çıkarılmış hem de seçme adımları birkaç adımdan tek adıma indirilmiştir. Bunun gerçekleştirilebilmesi için 'Depth Ray' yöntemindeki gibi sanal bir ışın kullanılmıştır. Sağ el ışının bitişiğindeki imlecin konumunu kontrol ederken, sol el seçilebilecek nesnelere istenileni seçmeyi belirlemede kullanılmıştır. Işının hareketine göre ışının üzerindeki ve imlecin belirli bir mesafe uzaklığındaki tüm nesnelere ekranın bir köşesinde listelenir ve sol elin konumuna göre bu nesnelere seçilebilir hale getirilir. Ardından sağ elle seçme komutu verildiğinde

ekranda istenilen nesne seçilmiş olur. Yöntemin çalıştığı örnek bir seviye ekranı Şekil 19'da görülebilir.



Şekil 19 Uyarlanmış 3B Hibrit Squad Yöntemi seviye ekranı

Yöntemin Şekil 19'da görüldüğü gibi çalışır hale getirilebilmesi için biri seçme işlemini kontrol eden, diğeri ışının üzerinde kalan nesnelere bulan ve bir diğeri ise kalan tüm ayarlamaları kontrol eden üç yöntem kullanılmıştır. Bu yöntemlerin sözde kod yazımı aşağıda verilmiştir.

```
void function UpdateSquadSelectNumber{
    sol el konumunu ve omuz konumunu al;
    sol el omuz vektörünü oluştur;
    sol el omuz vektörünün y değerine göre 1'den 5'e kadar olan
    gösterge değeri hesaplanır.
}

void function Clicked{
    sol elin durumuna göre gönderilen sayı bilgisini al;
    sol kanvasa bulunacak nesne listesinden ilgili gösterge
    değerindeki veriyi al;
    if( seçili nesne = sıradaki seçilecek nesne){
        seçim bilgisini veri tabanına yaz;
        ekrandaki toplanan alanını güncelle;
        seçim sesini aktifleştir;
        bir sonraki seçilecek nesnenin rengini değiştir ya da seviye
        sonu panelini aktifleştir;
    }
}

void function SquadRayController{
    for( ; ; ){
        kameradan imlece doğru bir ışın çiz;
        ışınla kesişen nesnelere bul ve bu nesnelere bir liste
        oluştur.
    }
}
```

```

void function Update{
  for( ; ; ){
    if(durdurulmuş){
      dön;
    }

    imleç pozisyonunu al;
    foreach( nesne : tüm seçilebilir nesnelere){
      imleçin pozisyonu ile nesne arası uzaklığı hesapla;
      hesaplanan değeri nesnenin merkez uzaklık alanlarına
      ata;
    }

    foreach( nesne : tüm seçilebilir nesnelere){
      ışının üzerinde bulunan nesnelere al ve ilgili
      alanlarına bu bilgiyi ata;
    }

    foreach( nesne : tüm seçilebilir nesnelere){
      if( nesne != seçili nesne){
        if( nesne ışının üzerinde || nesnenin imleç
        uzaklığı < belirlenen aralık){
          nesneyi yarı geçirgen yap;
        }else{
          nesnenin yarı geçirgenliğini kaldır;
          nesnenin rengini eski haline getir;
        }
      }

      if( nesne ışının üzerinde || nesnenin imleç uzaklığı <
      belirlenen aralık){

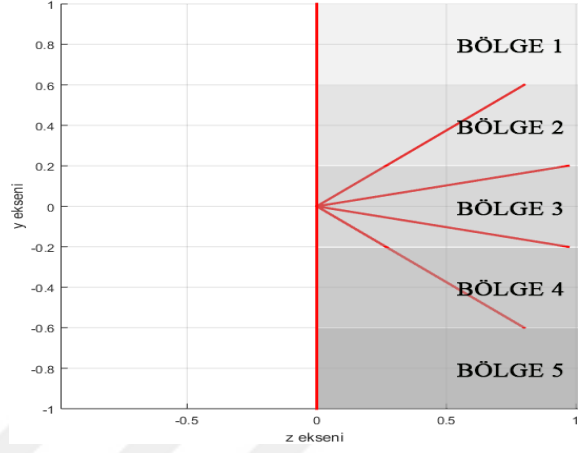
        if(nesne ışının üzerinde){
          sol kanvasa bulunacak nesne listesinin
          başına nesneyi ekle;
        }else{
          sol kanvasa bulunacak listenin sonuna
          nesneyi ekle;
        }
      }
    }
    else{
      nesnenin yarıgeçirgenliğini kaldır;
      nesnenin rengini eski haline getir;
    }
  }

  foreach( nesne : sol kanvasa bulunacak nesne listesi){
    nesnenin resim dokusunu al;
    sol kanvasa bulunan boşluklara resim dokusunu bastır;
  }

  süre alanını güncelle;
}
}

```


'UpdateSquadSelectNumber' fonksiyonu, sol kanvasta listelenen nesnenin seçilebilmesi için kullanılan sol el gösterge değerini almak için kullanılır. Sol el ve omuz konumları alındıktan sonra oluşturulan sol el omuz vektörünün y eksenindeki büyüklüğüne göre bir gösterge değeri döndürülür. Döndürülen gösterge değeri Şekil 20'de görülebilir.



Şekil 20 Uyarlanmış Hibrit Squad yönteminde sol kol seçim işlemi(solda) ve seçim bölgeleri(sağda)

Nesnenin seçilme durumunu kontrol eden 'Clicked' fonksiyonu bir olay dinleyicisi olarak çalışmaktadır. Diğer yöntemlerden tamamen bağımsız olan bu fonksiyon, eklem koordinatlarının alındığı sınıfta el durumu açık durumdan kapalı duruma geçtiğinde tetiklenir. Ardından o anda sol el gösterge değeri kullanılarak tespit edilen ve sistemde aktif olarak seçili olan nesne, seçilecek nesnelere listesindeki sıradaki nesne ile karşılaştırılır. Bu karşılaştırma sonucunda her iki nesne de aynı nesne ise veri tabanına yazma, ekrandaki ilgili alanları güncelleme, seçim sesini aktifleştirme, bir sonraki seçilecek nesnenin rengini değiştirme ya da seviye sonu panelini aktifleştirme işlemleri yapılır.

Gönderilen sanal ışının üzerinde bulunan nesnelere tespit eden 'SquadRayController' fonksiyonu ise Unity'nin fizik motorunu kullanarak gönderilen ışınla çarpışan tüm seçilebilir nesnelere bir listeye atar. Bu sayede ışının üzerinde bulunan nesnelere tespit edilerek üzerlerinde istenilen işlemler gerçekleştirilebilir.

Yöntemin diğer çalışma ayarlamalarını kontrol eden 'Update' fonksiyonu ise Unity tarafından her bir grafik çerçevesinde bir defa çalıştırılmaktadır. Bu sayede sahnede herhangi bir anda yapılmak istenilen tüm işlemler bu fonksiyonun içerisinde kolaylıkla

yapılabilmektedir. 'Update' fonksiyonunda imleç pozisyonu hesaplandıktan sonra sahnede bulunan tüm seçilebilir nesnelere olan merkez uzaklık hesaplanır. Uzaklıklar hesaplandıktan sonra ışının üzerinde bulunan nesnelere bir listeye atılarak kullanıma hazır hale getirilir. Ardından ışının üzerinde bulunan ya da imlece belirlenen mesafeden daha yakın mesafede olan nesnelere sol kanvasta bulunacaklar listesine atılır. Bu liste doldurulurken ışının üzerindeki nesnelere her zaman başta olacak şekilde listeye ekleme yapılır. Bunun yanı sıra belirlenen yarı geçirgenlik mesafesi içerisine giren tüm nesnelere yine bu fonksiyon içerisinde yarı geçirgen hale getirilir. Daha önceden rengi değiştirilmiş ya da yarı geçirgen hale getirilmiş olan nesnelere eski hallerine getirilebilmeleri için gerekli olan kontrol ve işlemlerde yine bu fonksiyon içerisinde gerçekleştirilir.

4.3.9. Kullanıcı Verilerinin Kaydedilmesi

Unity oyun motoru, geliştirilen oyun ya da uygulamaların birçok farklı platformda çalışmasını sağlayan özelliklere sahiptir. Bu sebeple geliştirilen ürünler eğer farklı platformlarda da çalıştırılmak isteniliyorsa, platform bağımsız bir veri depolama yöntemi kullanılmalıdır. Bu sebeple kullanıcı verilerinin toplanabilmesi için sqllite veri tabanı tercih edilmiştir. Sqllite veri tabanı ile iletişim kurulması amacıyla veri tabanı paketi oluşturulmuş ve gerekli sınıflar tanımlanmıştır. Oyunu oynayacak kullanıcı, seviyeleri açmadan önce isim soy isim bilgilerini girdikten sonra bu bilgiler veri tabanına kaydedilir. Ardından kullanıcıların oynadığı seviyeler ve seviyelerdeki tüm nesne seçme süreleri veri tabanına kaydedilir. Bu sayede yöntemlerin kıyaslanmasında seviye bazlı ya da nesne bazlı kıyaslanmanın kolaylıkla yapılması hedeflenmiştir.

Veri tabanı tabloları ve alanları oluşturulurken yöntemlerin en iyi biçimde kıyaslanmasını ve grafiklerin kolaylıkla elde edilebilmesini sağlayacak biçimde bir organizasyon yapılmıştır. Oluşturulan tablolar ve bu tabloların alanları Tablo 2, Tablo 3 ve Tablo 4'de gösterilmiştir.

Tablo 2 User tablosu alanları ve özellikleri

Alanları	Özellikleri
id	INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT

name	VARCHAR(50)
surname	VARCHAR(25)

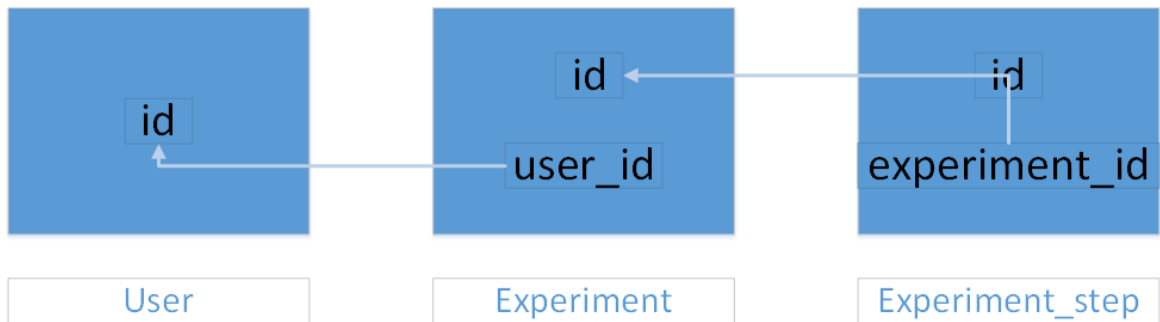
Tablo 3 Experiment tablosu alanları ve özellikleri

Alanları	Özellikleri
id	INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT
user_id	INTEGER NOT NULL

Tablo 4 Experiment_step tablosu alanları ve özellikleri

Alanları	Özellikleri
id	INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT
experiment_id	INTEGER NOT NULL
level	INTEGER NOT NULL
level_select_index	INTEGER NOT NULL
duration	FLOAT NULL

Veritabanı tablolarının birbirleri ile bağlantıları ise Şekil 21’de gösterilmiştir. Sistemde kullanılan üç tabloda birbirleri ile yabancı anahtarlar ile bağlı durumdadır.



Şekil 21 Veritabanı tablo ilişkileri

5. KULLANICI TESTLERİNİN UYGULANMASI VE VERİLERİN ANALİZ EDİLMESİ

5.1. Kullanıcı Testlerinin Uygulanması

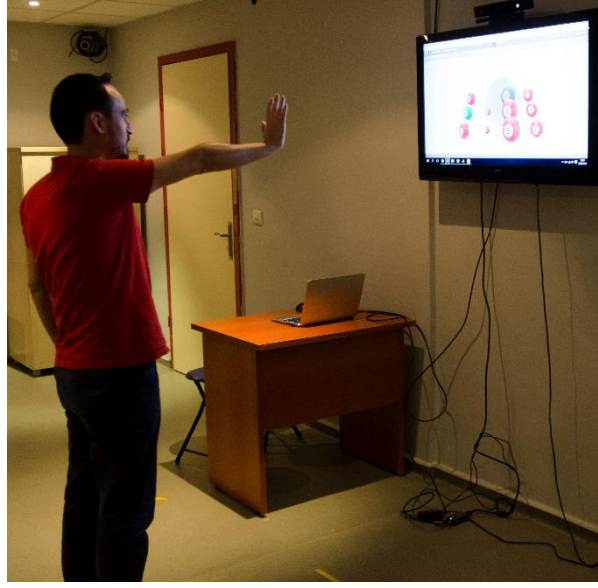
Yöntemlerin kıyaslanabilmesi için farklı kullanıcılarla test edilmesi gerekmektedir. Bu amaçla geliştirilen deneysel 3B oyun, yaşları 18 ile 37 arasında değişen ve yaş ortalamaları 22 olan 17 farklı kullanıcı (5 kadın, 12 erkek) ile test edilmiştir. Bir kullanıcı testi ortalama 20 dakika sürmüştür.

Kullanıcı testleri yapılmadan önce oyun ortamının ayarlanması, kameranın oyun alanını en iyi biçimde görebilecek bir yere yerleştirilmesi ve kamera kalibrasyonunun sağlanması işlemleri yapılmıştır. Testlerin yapılacağı oyun ortamı kameranın çalışmasını etkileyebilecek nesnelere arındırılmıştır. Bunun yanı sıra kullanıcıların kamera ile etkileşim içerisinde bulunabilecekleri en yakın ve en uzak mesafeler belirlenmiş ve kullanıcıların belirlenen aralıklarda oyun oynamaları sağlanmıştır.

Tüm katılımcılara hangi nesnelere seçilmesi gerektiği, ekrandaki farklı renkteki nesnelere neyi ifade ettiği, testin neyi amaçladığı gibi başlıklarla ilgili genel bilgilendirme yapılmıştır. Bu sayede katılımcıların oyuna daha kolay aşına olmaları amaçlanmıştır. Ayrıca tüm katılımcılara, kullanıcı etkileşiminin ve yöntemlerin çalışma prensiplerinin anlaşılabilirliği amacıyla her bir yöntem için en az bir seviye oynattırılmıştır. Bu aşamada katılımcılara herhangi bir süre kısıtlaması verilmemiş ve katılımcılar hazır olduklarını belirtene kadar beklenmiştir.

Hazırlık ve bilgilendirme işlemlerinin tamamlanmasının ardından geliştirilen arayüz yardımıyla katılımcıların sisteme kaydı yapılmıştır. Sisteme kaydı yapılan katılımcılar her bir seviyede 8 farklı nesne seçme işlemi olmak üzere toplamda 9 seviyede 72 nesne seçme işlemi yapmışlardır. Yapılan tüm seçme işlemi süreleri sistem tarafından ayrı ayrı olarak veri tabanına kaydedilmiştir.

Şekil 22'de görüldüğü biçimde gerçekleştirilen test uygulaması sona erdikten sonra katılımcılar daha önceki tecrübeleri ile ilgili 5 ve oynadıkları yöntemler ile ilgili 8 soru barındıran EK-1 anketini doldürmüşlardır. Bu sayede yöntemlerin kullanım kolaylığı, hız ve kesinlik özellikleri açısından farklı katılımcılar için değerlendirilmesi amaçlanmıştır.



Şekil 22 Kullanıcı testlerinin yapıldığı test ortamı

Testlerin tümü 8GB ram bulunduran Intel HD 4000 ekran kartına sahip Intel Core-i5 işlemcili dizüstü bilgisayarda çalışan oyunun 42 inç boyutunda bir televizyona yansıtılması ile tamamlanmıştır.

5.2. Fitts Analizi

Fitts Kanunu bir görevin tamamlanması için gereken süre ile hedef ve başlangıç noktaları arası uzaklık ve hedef nesnenin büyüklüğü arasındaki ilişkiyi inceler. Kanun 1954 yılında Paul Fitts tarafından geliştirilmiştir ve işaretleyici araçlar ile yapılan test işlemlerinin temelini oluşturmuştur [41]. Daha sonraları MacKenzie ve Buxton bu çalışma üzerinde bazı iyileştirmeler yapmış ve tek boyut için geliştirilen eşitliği ikinci boyuta genişletmişlerdir [42]. Ortaya çıkardıkları Eşitlik 4'te işaretleyicinin hedef alana taşınma süresi MT'nin başlangıç ve hedef noktası arası uzaklık D ve hedef büyüklüğü W arasındaki bağlantıyı ortaya koymuşlardır.

$$MT = a + b \log_2 \left(\frac{D}{W} + 1 \right) \quad (4)$$

Eşitlikte $(\log_2 \frac{D}{W} + 1)$ ifadesi zorluk göstergesi (ID) olarak ele alınmıştır. İlgili eşitlik yorumlandığında başlangıç ve bitiş noktası arası uzaklıklar arttığında ya da hedef küçüldüğünde görevin zorluk düzeyinin arttığı görülebilir. Aynı eşitlikten zorluk göstergesi ve geçen zamanın doğrusal bir ilişkisi olduğu çıkarılabilir (a ve b değeri ortam bazlı sabitler olarak belirlenmiştir).

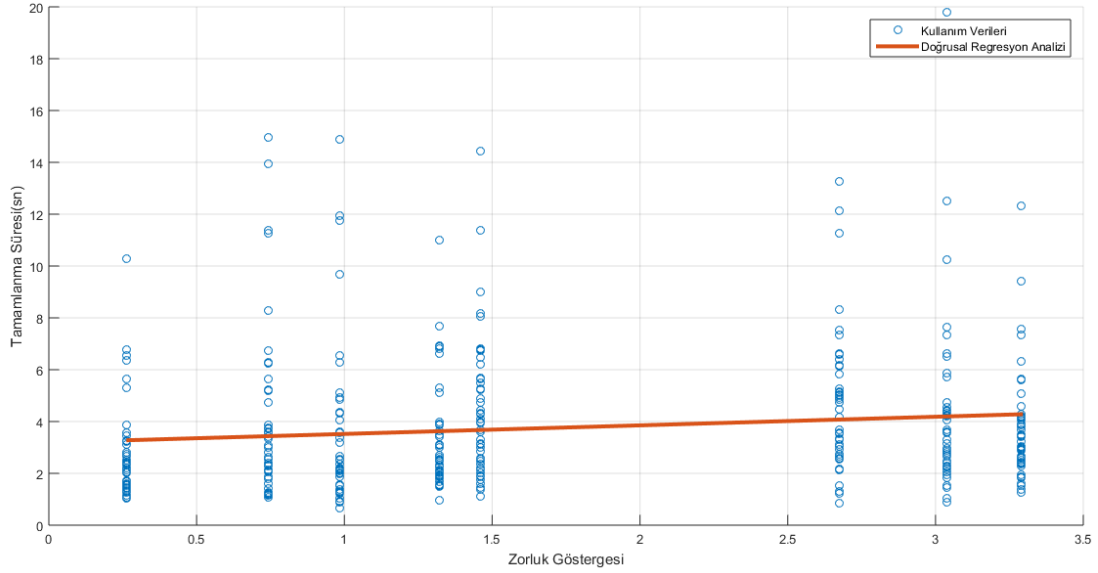
Geliştirilen ve kullanıcılara test ettirilen deneysel oyundan elde edilen test verileri Eşitlik 4'e göre analiz edilmiştir. Kullanıcı verilerinden zorluk göstergesi değerleri elde edilirken kullanılacak olan uzaklık ve hedef büyüklüğü, geliştirilen ortamlardaki nesnelerin konum ve yarıçap bilgilerinden elde edilmiştir. Ortamlardaki nesne sayılarının, konumlarının ve seçim sıralarının farklı olmalarından dolayı nesne az yoğun ve nesne yoğun ortamlar için Fitts analizi ayrı ayrı yapılmıştır. Nesne hareketli ortamlarda ise uzaklık değeri değişken olduğu için bu analiz yapılmamıştır. Bu analizlerde kullanılan koordinat ve hedef büyüklüğü değerleri Tablo 5'te verilmiştir. Bu değerler geliştirilen deneysel oyundaki seviyelerde bulunan nesnelere alınmıştır.

Tablo 5 Testler esnasında seçilen nesnelerin koordinat ve seçim alanı büyüklükleri

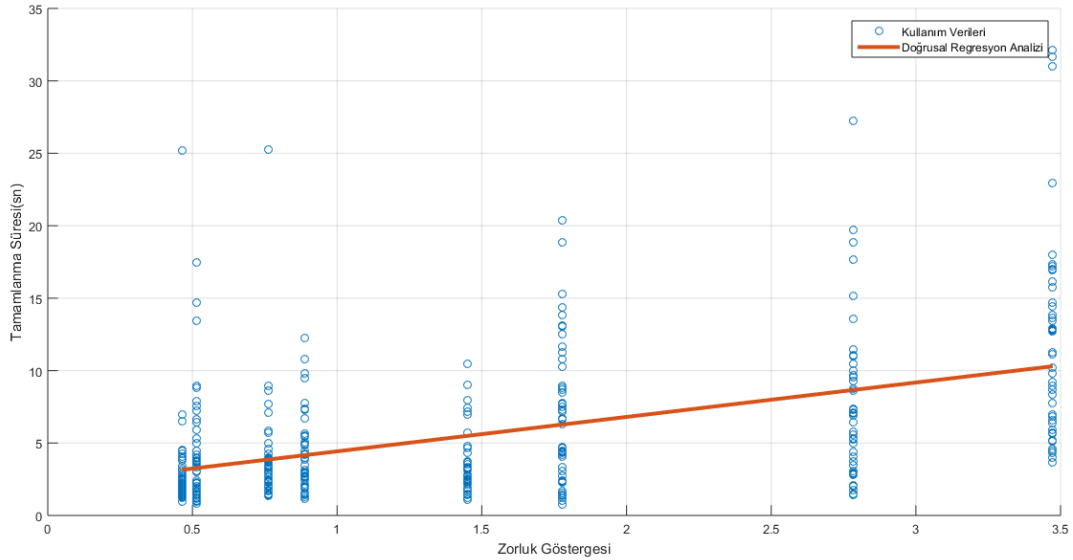
Ortam	Nesne Adı	Koordinat (x, y, z)	Hedef Büyüklüğü
Nesne Az Yoğun Ortam	Nesne 1	$x = -6, y = 0, z = 5$	8
	Nesne 2	$x = -2, y = 2, z = 8$	1
	Nesne 3	$x = -2, y = -2, z = 2$	1
	Nesne 4	$x = 2, y = 0, z = 5$	27
	Nesne 5	$x = 6, y = -2, z = 2$	8
	Nesne 6	$x = -6, y = 2, z = 8$	8
	Nesne 7	$x = 6, y = 2, z = 8$	8
	Nesne 8	$x = -2, y = 0, z = 5$	1
Nesne Çok Yoğun Ortam	Nesne 1	$x = 0, y = 2, z = 9$	27
	Nesne 2	$x = -1, y = 3, z = 7$	1
	Nesne 3	$x = 0, y = 1, z = 2$	1
	Nesne 4	$x = 6, y = -2, z = 2$	8
	Nesne 5	$x = -6, y = 0, z = 5$	8
	Nesne 6	$x = -6, y = -2, z = 2$	8
	Nesne 7	$x = -0, y = 5, z = 4$	1
	Nesne 8	$x = 6, y = 4, z = 2$	8

Eşitlik 4'te bulunan seçilecek nesnenin uzaklığı bilgisi bir önceki seçilmiş nesne ile aralarındaki Öklid uzaklığı olarak hesaplanmıştır. Seçilen ilk nesne için uzaklık

hesaplaması ise orijine göre yapılmıştır. Ayrıca ortamlarda bulunan nesnelere dairesel bir biçimde oldukları için yarıçapı 1, 2, 3 olan dairelerin hedef büyüklüğü 1, 8, 27 olarak kullanılmıştır. Tüm bu değerler kullanılarak zorluk göstergesi hesaplanmış ve nesne az yoğun ortamlar için Şekil 23, nesne yoğun ortamlar için Şekil 24 ortaya çıkarılmıştır. Şekiller incelendiğinde zorluk göstergesi arttıkça her iki ortam için de tamamlanma süresinin arttığı görülebilir.



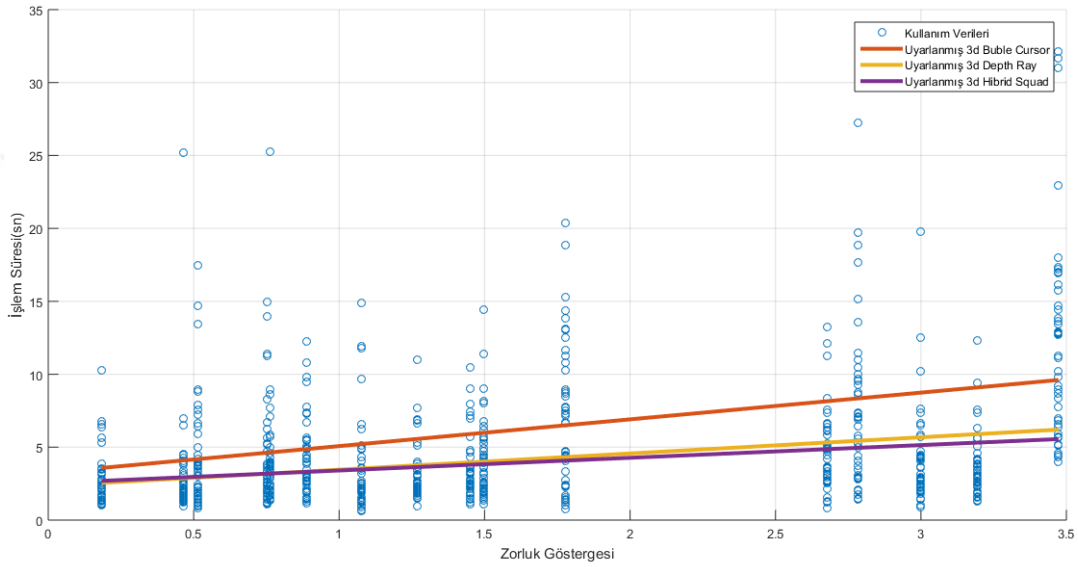
Şekil 23 Nesne az yoğun ortamda zorluk göstergesine bağlı seçim süresi değişimi



Şekil 24 Nesne yoğun ortamda zorluk göstergesine bağlı seçim süresi değişimi

Şekil 23 ve Şekil 24 oluşturulurken, ekranda belirtilen değerler ham verilerden elde edilmiştir. Sekizer farklı zorluk göstergesi için işlem süresi grafikleri çizilirken doğrusal

regresyon işlemi yapılmış ve 95% güven aralığında Şekil 23 için güven sınırları [2.66, 3.71] ve [0.07, 0.59], Şekil 24 için ise güven sınırları [1.24, 2.85] ve [2.93, 2.81] olarak hesaplanmıştır. Ayrıca doğrusal regresyon katsayıları Şekil 23 için a değeri 3.18, b değeri 0.33 ve Şekil 24 için a değeri 2.05 ve b değeri 2.37 olarak ortaya çıkmıştır. MacKenzie [43]'nin Fitts kanununun performans değerlendirmesi ile ilgili yaptığı çalışmada kullandığı performans göstergesi değeri $IP=1/b$ eşitliği kullanıldığında, nesne az yoğun ortamdaki performans değerinin(3.03) nesne yoğun ortamdaki performans değerinden(0.42) daha yüksek olduğu yorumu yapılabilir.



Şekil 25 Geliştirilen yöntemler için zorluk göstergesine bağlı işlem süresi değişimi

Bunun yanı sıra her bir yöntem için analizler ayrı ayrı gerçekleştirilmiştir. Yöntemler karşılaştırılırken her bir yöntemin nesne az yoğun ve nesne yoğun ortamlardaki değerleri tek bir kümede kabul edilip kıyaslama yapılmıştır. Zorluk göstergeleri için işlem süresi grafiği Şekil 25'te görüldüğü gibi çizilirken doğrusal regresyon işlemi yapılmıştır. Regresyon işlemi sonucunda 95% güven aralığında sırasıyla 'Uyarlanmış 3B Bubble Cursor', 'Uyarlanmış 3B Depth Ray' ve 'Uyarlanmış 3B Hibrit Squad' yöntemleri için alt ve üst güven sınırlarının sırasıyla [2.05 4.42] [1.21 2.45], [1.57 3.11] [0.70 1.51] ve [1.82 3.23] [0.50 1,23] olduğu tespit edilmiştir. Ayrıca doğrusal regresyon katsayıları 'Uyarlanmış 3B Cubble Cursor' yöntemi için a değeri 3.24 ve b değeri 1.83, 'Uyarlanmış 3B Depth Ray' yöntemi için a değeri 2.34 ve b değeri 1.11 ve 'Uyarlanmış 3B Hibrit Squad' yöntemi için a değeri 2.53 ve b değeri 0.87 olarak ortaya çıkmıştır. Bu veriler ışığında performans göstergesi değerleri kıyaslandığında performansı en yüksek yöntemin

1.14 performans göstergesi(IP) değeri ile ‘Uyarlanmış 3B Hibrit Squad’ yöntemi olduğu görülebilir.

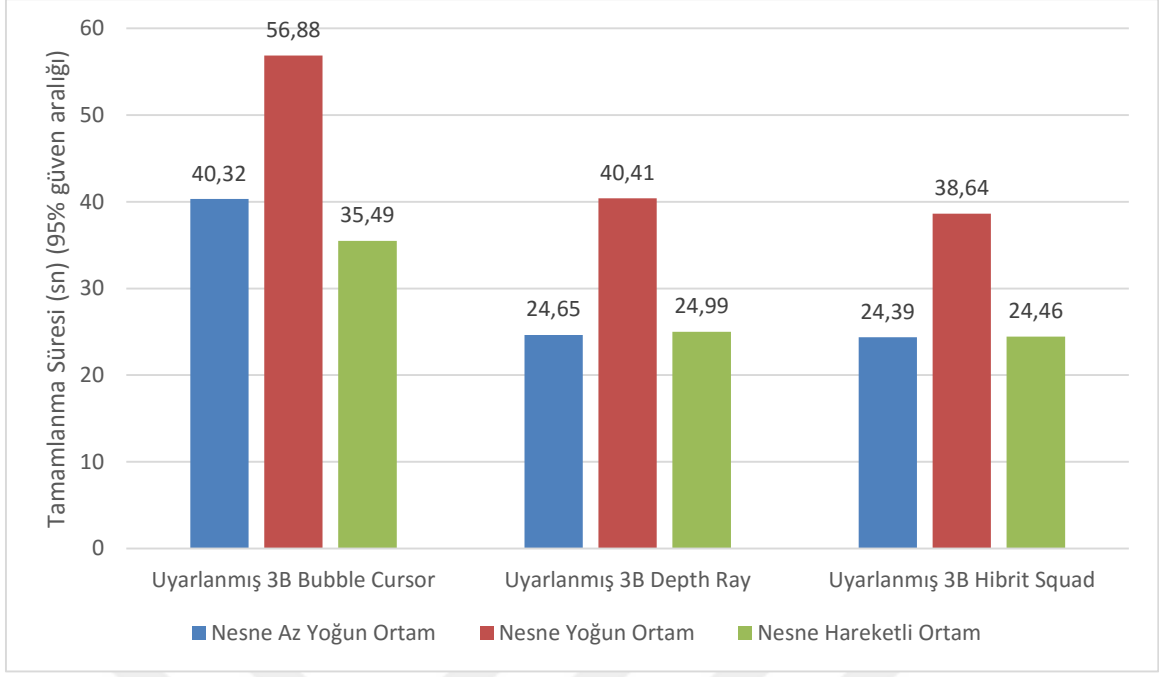
5.3. Katılımcı Oyun Verilerinin Analizi

Testlerden elde edilen verilerin tamamlanma süreleri bağımlı değişkeni üzerinde analizinin yapılabilmesi için önce normallik varsayımı yapılmıştır. Tüm seviye verileri üzerinde Kolomogorov-Smirnov normallik varsayımı yapılmış ve tüm seviyeler için p değerinin 0.05’den küçük olduğu ve bu durumda seviye verilerinin normal dağılıma uygun olduğu görülmüştür. Ayrıca varyansların homojen olup olmadığı testi yapılmış ve $p = 0.156$ değeri ile varyansların homojen olduğu tespit edilmiştir. Normal dağılımlı ve homojen olduğu tespit edilen veriler için 95% güven aralığında ortalama tamamlanma süreleri Tablo 6’te verilmiştir.

Tablo 6 Seviye tamamlama ortalamalarının karşılaştırılması

Seviye	Ortalama (sn)	Veri Adedi	Standart Sapma
Seviye 1	40,3197	17	13,08589
Seviye 2	56,8762	17	14,62668
Seviye 3	35,4934	17	10,12128
Seviye 4	24,6506	17	8,42693
Seviye 5	40,4128	17	13,21565
Seviye 6	24,9862	17	10,75386
Seviye 7	24,3904	17	7,76899
Seviye 8	38,6378	17	8,52657
Seviye 9	24,4620	17	7,48686
Tümü	34,4699	153	14,78722

Tablo 6’te bulunan verilerin ortamlara göre dağılımın grafiği ise Şekil 26’da görülebilir.



Şekil 26 Seviye tamamlama ortalamalarının yöntemlere ve ortamlara göre dağılımı

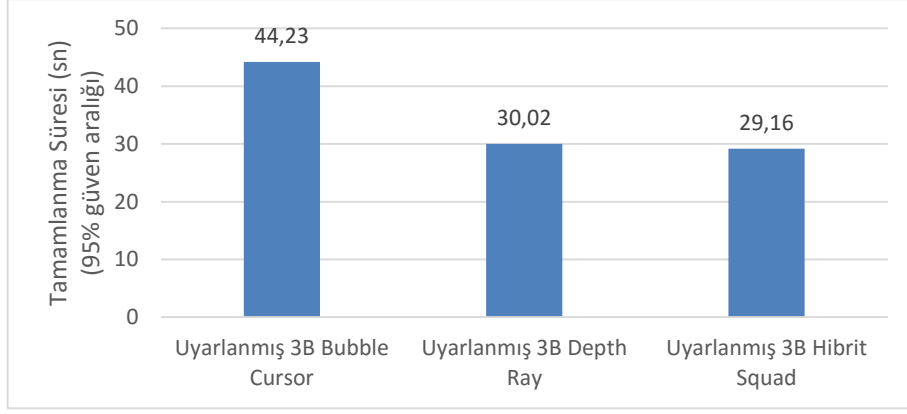
Şekil 26’da görüldüğü gibi ‘Uyarlanmış 3B Depth Ray’ ve ‘Uyarlanmış 3B Hibrit Squad’ yöntemleri ile birbirlerine çok yakın sürelerde seçme işlem yapılabilirken ‘Uyarlanmış 3B Bubble Cursor’ yöntemi ile bu işlem daha uzun sürede tamamlanabilmektedir. ‘Uyarlanmış 3B Hibrit Squad’ yönteminin ‘Uyarlanmış 3B Depth Ray’ yöntemine göre çok kısa süre farkla daha hızlı seçim yapılabilmesini sağladığı görünmesine rağmen bu farkın anlamlı olup olmadığının analiz edilmesi gerekmektedir. Bu amaçla ‘Uyarlanmış 3B Bubble Cursor’, ‘Uyarlanmış 3B Depth Ray’ ve ‘Uyarlanmış 3B Hibrit Squad’ yöntemleri gruplanmış ve gruplanan veriler üzerinde ANOVA analizi yapılmıştır.

ANOVA analizi yapılmadan önce veriler üzerinde Kolomogorov-Smirnov normallik varsayımı yapılmış ve tüm yöntemler için p değerlerinin 0.05’den büyük olduğu görülmüştür. Ayrıca yine gruplanmış veriler üzerinde homojenlik testi yapılmış ve $p = 0.078$ değeri ile verilerin homojen olduğu görülmüştür. Homojen ve normal dağılımlı verileri üzerinde ANOVA analizi yapılmış ve p değerinin 0.05’den küçük olduğu görülmüştür ($F = 21.130$). Bu veri ile birlikte yöntemler arasında tamamlanma süreleri açısından fark olduğu hipotezi doğrulanmıştır. Hangi yöntemler arasında fark olup hangileri arasında olmadığını belirlemek için Tukey HSD post hoc çoklu karşılaştırma testi yapılmış ve Tablo 7’de görülen sonuçlar ortaya çıkmıştır.

Tablo 7 Yönteme göre gruplanmış veriler üzerinde yapılan Tukey HSD post hoc çoklu karşılaştırma sonuçları

(I) Ortam	(J) Ortam	Ortalama Farkı (I-J)	Standart Hata	Anlamlılık Değeri (p)	95% Güven Aralığı	
					Alt Sınır	Üst Sınır
Uyarlanmış 3B Bubble Cursor	Uyarlanmış 3B Depth Ray	14,21323	2,60372	,000	8,0497	20,3768
	Uyarlanmış 3B Hibrid Squad	15,06634	2,60372	,000	8,9028	21,2299
Uyarlanmış 3B Depth Ray	Uyarlanmış 3B Bubble Cursor	-14,21323	2,60372	,000	-20,3768	-8,0497
	Uyarlanmış 3B Hibrid Squad	,85312	2,60372	,943	-5,3104	7,0167
Uyarlanmış 3B Hibrid Squad	Uyarlanmış 3B Bubble Cursor	-15,06634	2,60372	,000	-21,2299	-8,9028
	Uyarlanmış 3B Depth Ray	-,85312	2,60372	,943	-7,0167	5,3104

Tablo 7’de gösterilen veriler yorumlandığında ‘Uyarlanmış 3B Depth Ray’ ve ‘Uyarlanmış 3B Hibrit Squad’ yöntemleri arasında anlamlı bir fark olmadığı görülebilir. Yine aynı tabloda p değerlerine bakılarak ‘Uyarlanmış 3B Bubble Cursor’ yönteminin tamamlanma süresi bakımından diğer yöntemlerden farklı olduğu söylenebilir. Yöntemlere göre gruplanmış verilerin 95% güven aralığında tamamlanma süresi grafik biçiminde gösterimi ise Şekil 27’de görülebilir. Grafikte ‘Uyarlanmış 3B Bubble Cursor’ yöntemi kullanılarak yapılan seçme işleminin diğer ortamlarda yapılan seçme işlemlerine göre daha uzun sürede yapıldığı görülebilir.



Şekil 27 Seviye tamamlama ortalamalarının yöntemlere göre dağılımı

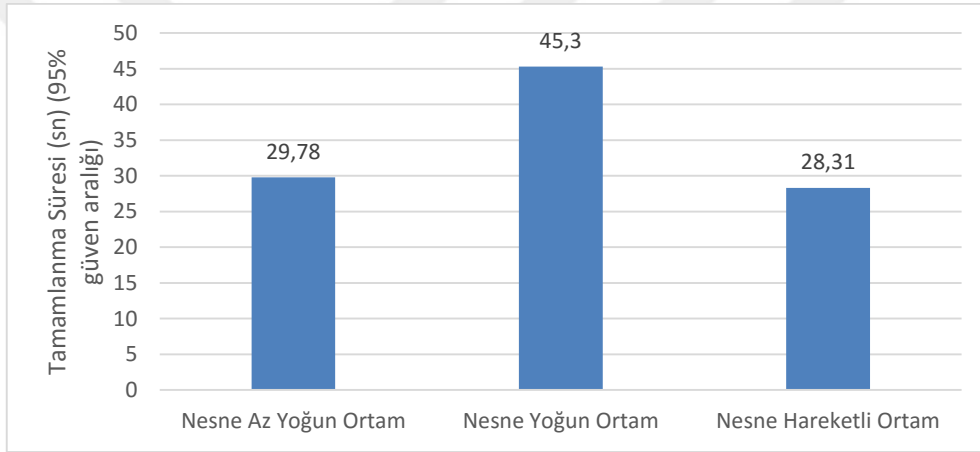
Ortamların nesne seçimi üzerindeki etkisini görmek için ise veriler ortamlara göre ‘Nesne Az Yoğun’, ‘Nesne Yoğun’ ve ‘Nesne Hareketli’ biçiminde sınıflandırılmıştır. Bu gruplandırmanın ardından veriler üzerinde Kolomogorov-Smirnov normallik varsayımı yapılmış ve tüm yöntemler için p değerlerinin 0.05’den büyük olduğu görülmüştür. Ayrıca yine gruplanmış veriler üzerinde homojenlik testi yapılmış ve $p = 0.378$ değeri ile verilerin homojen olduğu görülmüştür. Homojen ve normal dağılımlı verileri üzerinde ANOVA analizi yapılmış ve p değerinin 0.05’den küçük olduğu görülmüştür ($F = 28.033$). Bu veri ile birlikte yöntemler arasında tamamlanma süreleri açısından fark olduğu hipotezi doğrulanmıştır. Hangi yöntemler arasında fark olup hangileri arasında olmadığını belirlenmesi için Tukey HSD post hoc çoklu karşılaştırma testi yapılmış ve Tablo 8’de görülen sonuçlar ortaya çıkmıştır.

Tablo 8 Ortama göre gruplanmış oyun verileri üzerinde yapılan Tukey HSD post hoc çoklu karşılaştırma sonuçları

(I) Ortam	(J) Ortam	Ortalama Farkı (I-J)	Standart Hata	Anlamlılık Değeri (p)	95% Güven Aralığı	
					Alt Sınır	Üst Sınır
Nesne Az Yoğun	Nesne Yoğun	-15,52199	2,51499	,000	-21,4755	-9,5685
	Nesne Hareketli	1,47304	2,51499	,828	-4,4805	7,4265
Nesne Yoğun	Nesne Az Yoğun	15,52199*	2,51499	,000	9,5685	21,4755
	Nesne Hareketli	16,99503*	2,51499	,000	11,0415	22,9485

Nesne Hareketli	Nesne Az Yoğun	-1,47304	2,51499	,828	-7,4265	4,4805
	Nesne Yoğun	-16,99503*	2,51499	,000	-22,9485	-11,0415

Tablo 8’da gösterilen veriler yorumlandığında ‘Nesne Az Yoğun’ ve ‘Nesne Hareketli’ ortamlar arasında anlamlı bir fark olmadığı görülebilir. Yine aynı tabloda p değerlerine bakılarak ‘Nesne Yoğun’ ortamın tamamlanma süresi bakımından diğer yöntemlerden farklı olduğu söylenebilir. Yöntemlere göre gruplanmış verilerin 95% güven aralığında tamamlanma süresi grafik biçiminde gösterimi ise Şekil 28’de görülebilir. Grafikte ‘Nesne Yoğun’ ortamda yapılan seçme işleminin diğer ortamlarda yapılan seçme işlemine göre daha uzun sürede yapıldığı görülebilir.

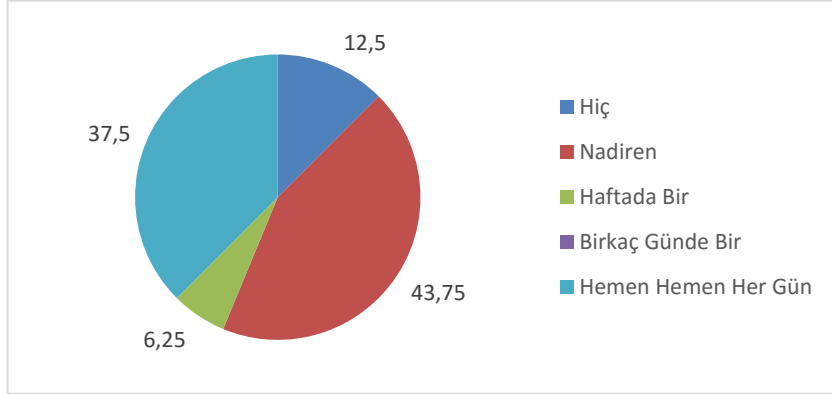


Şekil 28 Seviye tamamlama ortalamalarının ortamlara göre dağılımı

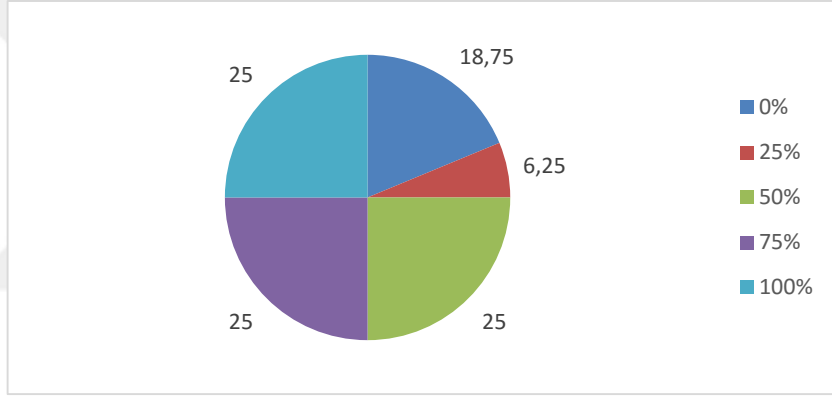
5.4. Katılımcıların Daha Önceki Tecrübelerinin Analizi

Katılımcıların doldurduğu anketin ilk grup soruları katılımcıların daha önceki tecrübelerinin belirlenmesine yönelik sorulardan oluşmaktadır. Katılımcıların bu sorulara verdikleri değerlendirmeler analiz edildiğinde 43.75% katılımcının haftada en az bir kez ya da daha sık video oyunu oynadıkları görülmüştür. Katılımcıların oynadıkları oyunların %75’ini 3B ortamlar için geliştirilmiş oyunlar oluşturmaktadır. Bunun yanı sıra katılımcıların oynadıkları oyunların 12.5% oranındaki kısmını uzaktan kontrole izin veren kısımlara sahip olan oyunlar oluşturmaktadır. Kinect kamera ile uzaktan kontrole izin veren oyun oynama oranı da 12.5% olarak görülmektedir. Tüm bunlarla birlikte katılımcıların %81.3’ü kendi oyun oynama becerilerini orta ve üstü olarak sınıflandırmıştır. Tüm bu önceki

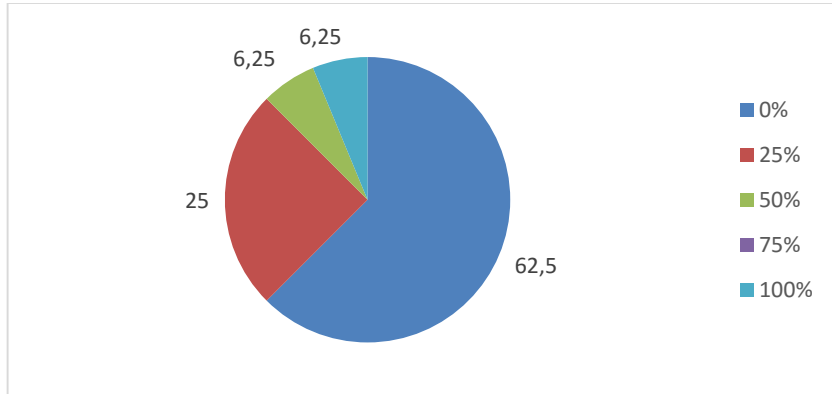
tecrübe verileri detaylı bir biçimde Şekil 29, Şekil 30, Şekil 31, Şekil 32 ve Şekil 33’de bulunan çizelgelerde görülebilir.



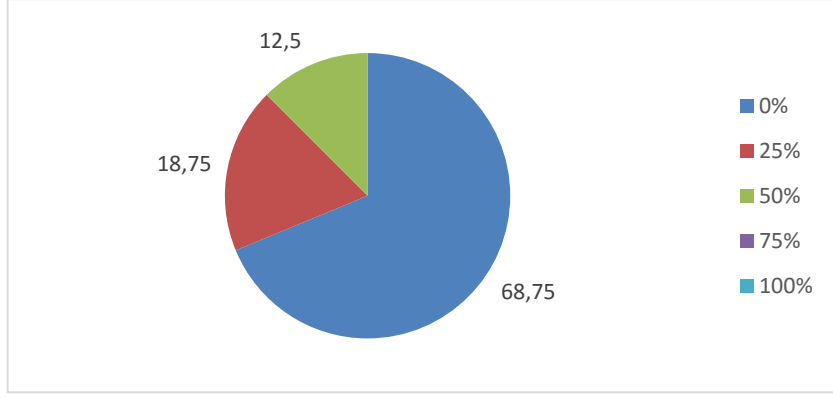
Şekil 29: Katılımcıların Video Oyunu Oynama Sıklığı



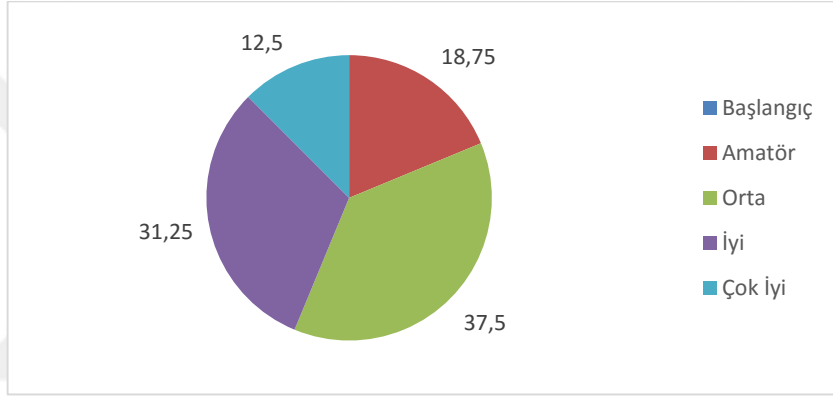
Şekil 30 3B ortamlar için geliştirilmiş oyun oynama sıklığı



Şekil 31 Uzaktan oyun kontrolüne izin verilen kısımlar olan oyun oynama sıklığı



Şekil 32 Kinect kamera kullanılarak uzaktan oyun kontrolüne izin verilen kısımlar olan oyun oynama sıklığı



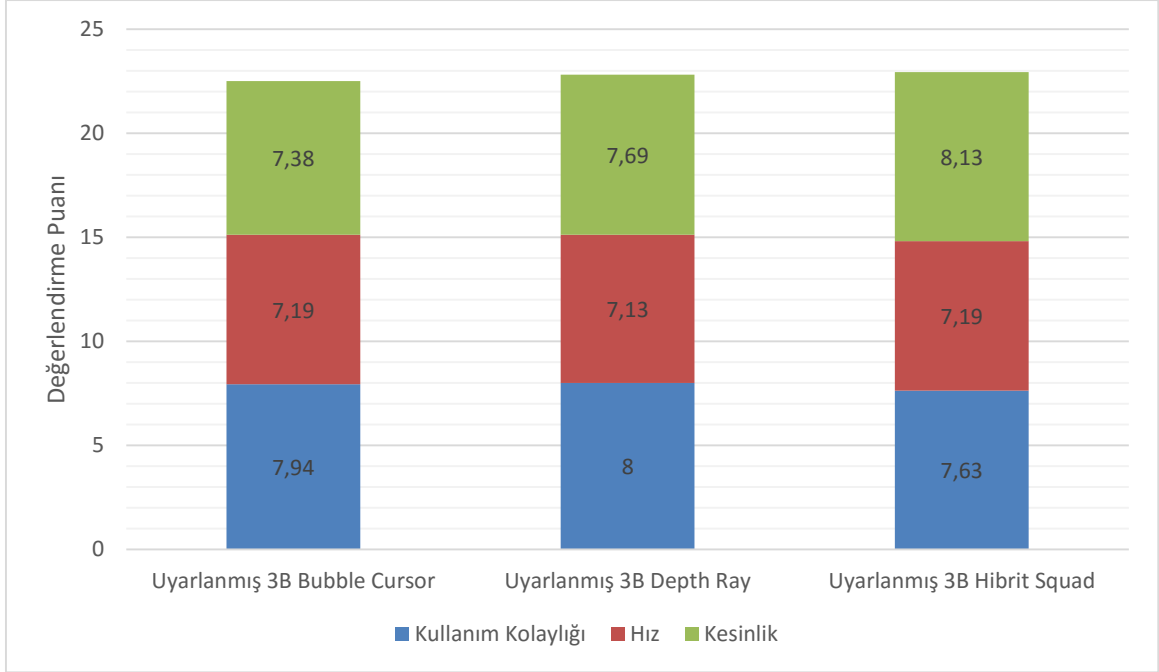
Şekil 33 Katılımcıların kendi oyun becerilerini sınıflandırma değerleri

5.5. Katılımcıların Oyun Değerlendirmelerinin Analizi

Kullanıcı tecrübe anketinde bulunan ve Kinect kameranın imleci hareket ettirme ve nesne seçme işlemlerindeki yeterliliğini tespit etmeyi amaçlayan ilk iki soruya verilen değerlendirmeler analiz edildiğinde her iki sorunun değerlendirme ortalamalarının 10 üzerinden 7 olduğu görülmüştür. Ayrıca 5'e kadar olan değerlendirmeleri yetersiz ve 5'ten büyük değerlendirmeleri yeterli kabul edersek her iki soru içinde katılımcılar %87.5 yeterli oyu kullanmışlardır. Bu istatistiklere bakılarak katılımcıların Kinect kamera ile etkileşimde çok fazla problem yaşamadıkları yorumu yapılabilir.

Kullanıcı değerlendirme anketinde bulunan bir diğer soru grubu ise yöntemlerin kullanım kolaylığı, hız ve kesinlik açısından değerlendirilmesine odaklanmaktadır. Bu değerlendirmelerin kıyaslanabilmeleri için değerlerin ortalamaları arasında kıyaslama yapılmıştır. Katılımcıların 10 üzerinden verdikleri puanların her bir yöntem için kullanım

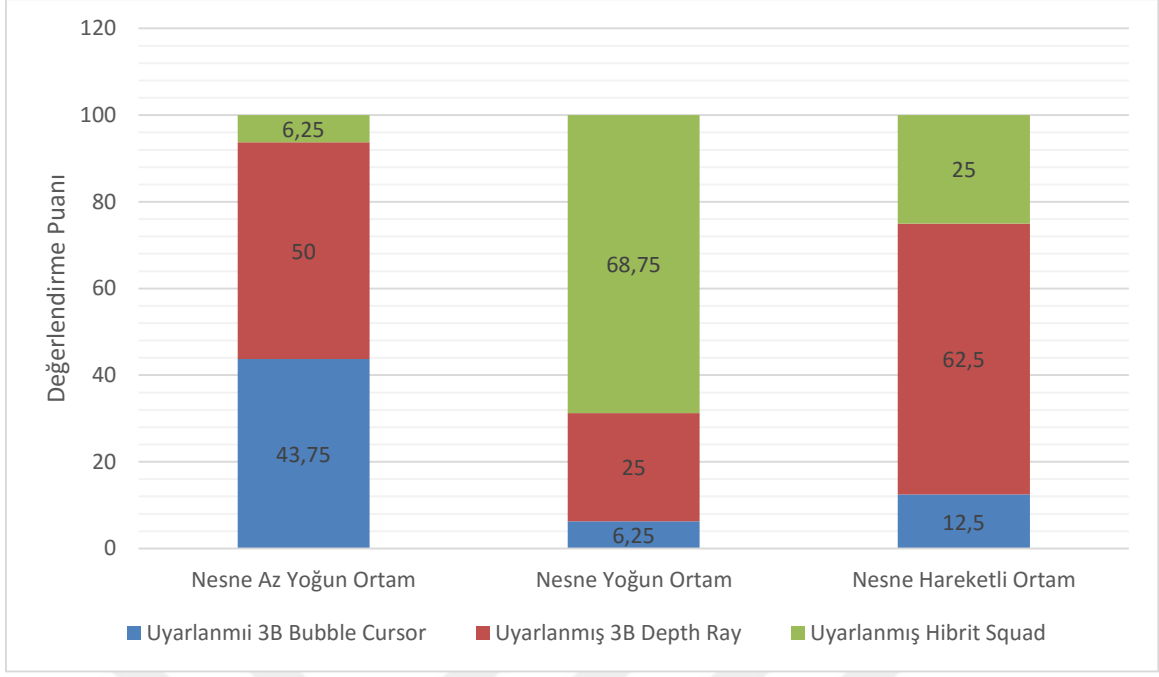
kolaylığı hız ve kesinlik açısından sonuçları Şekil 34’te verilmiştir. Bu grafikte ‘Uyarlanmış 3B Hibrit Squad’ yönteminin kesinlik açısından diğerlerinden daha yüksek değerlendirilmesi dolayısıyla toplamda diğer iki yöntemden daha yüksek değerlendirmeye sahip olduğu görülebilir.



Şekil 34 Katılımcılar tarafından yöntemlerin işlevsel olarak değerlendirilme grafiği

Ardından değerlendirmeler yöntemlere göre gruplanarak normallik ve homojenlik testi yapılmıştır. Kolomogorov-Smirnov testine göre verilerin normal dağılıma uygun olmadığı tespit edilmiş ancak homojen oldukları görülmüştür. Bu adımdan sonra ANOVA analizi yapılmış $p = 0,910$ olduğu görülmüştür ($F = 0,094$). Bu durumda 95% güven aralığında değerlendirmelerin birbirlerine göre farklarının anlamsız olduğu sonucuna ulaşılmıştır.

Yöntemlerin işlevselliğe göre değerlendirilmesinin ardından, ortamlarda tercih edilen yöntemlerin tespit edilmesi amacıyla Şekil 35’teki analiz yapılmıştır. Grafikteki veriler incelendiğinde ‘Nesne Az Yoğun’ ve ‘Nesne Hareketli’ ortamlarda ‘Uyarlanmış 3B Depth Ray’ yöntemi daha çok tercih edilirken ‘Nesne Yoğun’ ortamlarda ‘Uyarlanmış 3B Hibrit Squad’ yönteminin daha çok tercih edildiği görülebilir.



Şekil 35 Katılımcılar tarafından ortam bazlı olarak tercih edilen yöntem grafiği

Ortamlara göre gruplanmış Şekil 35'te görülen veriler üzerinde normallik ve homojenlik testi yapılmıştır. Kolomogorov-Smirnov testine göre verilerin normal dağılıma uygun olmadığı tespit edilmiş ancak homojen oldukları görülmüştür. Bu adımdan sonra ANOVA analizi yapılmış p değerinin 0,05'den küçük olduğu görülmüştür ($F = 10,43$). Bu durumda 95% güven aralığında değerlendirmelerin birbirlerine göre farklarının anlamlı olduğu sonucuna ulaşılmıştır. Hangi yöntemlerin birbirlerinden farklı olduğunu tespit etmek için ise normal dağılıma uygun olmayan veriler üzerinde Tamhane post hoc testi yapılmış ve tablo 9'deki değerler elde edilmiştir. Tablodaki anlamlılık değerleri incelendiğinde nesne az yoğun ve nesne hareketli ortamlar için anlamlı bir fark bulunmazken nesne yoğun ortamlardaki değerlendirmelerin diğerlerinden farklı olduğu görülebilir.

Tablo 9 Ortama göre gruplanmış değerlendirme verileri üzerinde yapılan Tamhane post hoc çoklu karşılaştırma sonuçları

(I) Ortam	(J) Ortam	Ortalama Farkı (I-J)	Standart Hata	Anlamlılık Değeri (p)	95% Güven Aralığı	
					Alt Sınır	Üst Sınır
Nesne Az Yoğun	Nesne Yoğun	-1,000	,219	,000	-1,55	-,45
	Nesne Hareketli	-,500	,219	,086	-1,05	,05

Nesne Yoğun	Nesne Az Yoğun	1,000	,219	,000	,45	1,55
	Nesne Hareketli	,500	,219	,086	-,05	1,05
Nesne Hareketli	Nesne Az Yoğun	,500	,219	,086	-,05	1,05
	Nesne Yoğun	-,500	,219	,086	-1,05	,05



6. SONUÇLAR VE ÖNERİLER

Teknolojinin ilerleme ile birlikte bilgisayarların kullanıldığı alanlar artış göstermiş ve bu durum insan bilgisayar etkileşimi çalışmalarının önemini artırmıştır. 3B ortamların yaygınlaşması ile birlikte ise 3B ortamlarda insan bilgisayar etkileşimi çalışmaları artış göstermiştir. 3B ortamlarda işaretleyici kullanılarak geliştirilen insan bilgisayar etkileşim yöntemleri araç taşıma ya da giyme zorunluluğu, kurulum süresi ve hareket kısıtlaması gibi nedenlerden dolayı çok fazla tercih edilmemiştir. Bu problemleri gidermek amacıyla el serbest etkileşim yöntemleri ortaya çıkmıştır. Bu yöntemlerde kullanıcılar herhangi bir harici donanım gereksizdir ellerini kullanarak direkt olarak sistemle etkileşim yapabilirler.

Tez çalışması kapsamında el serbest etkileşimle nesne seçme işlemine en uygun olan üç yöntem tespit edilmiş ve bu yöntemler yeni tasarlanan 3B sanal ortama uygun bir biçimde güncellenmiştir. Tasarlanan 3B sanal ortam kullanılarak deneysel bir oyun ortaya çıkarılmıştır. Deneysel oyunda el serbest etkileşimin sağlanması amacıyla Kinect yeni nesil kamera kullanılmıştır. Kinect kameranın gelişmiş yetenekleri sayesinde kullanıcı el hareketleri kolaylıkla sisteme aktarılmıştır. Ayrıca yöntemlerin farklı ortamlardaki davranışlarının gözlemlenmesi için ‘nesne az yoğun’, ‘nesne çok yoğun’ ve ‘nesne hareketli’ ortamlar ayrı ayrı oluşturulmuştur. ‘Nesne az yoğun’ ortam nesne sayısı daha az ve nesnelere arası mesafe daha uzun olacak biçimde tasarlanmıştır. ‘Nesne çok yoğun’ ortam ise sanal ortamdaki nesne sayısı daha fazla ve nesnelere arası mesafe daha az olacak biçimde tasarlanmıştır. ‘Nesne hareketli’ ortam ise ‘Nesne az yoğun’ ortamdaki nesnelere belirli doğrultularda hareket ettirilmesi ile oluşturulmuştur. Geliştirilen yöntemlerin tasarlanan ortamlarla eşleştirilmesi ile dokuz farklı seviye oluşturulmuştur.

Deneysel oyunda bulunan dokuz farklı seviye katılımcılara test ettirilmiş ve bu testin sonucunda tüm katılımcılar değerlendirme anketi doldürmüşlerdir. İlk olarak katılımcıların nesne seçme süreleri analiz edilmiş ve bu analiz neticesinde ‘Uyarlanmış 3B Depth Ray’ ve ‘Uyarlanmış 3B Hibrit Squad’ yöntemleri kullanılarak yapılan seçme işleminin ‘Uyarlanmış 3B Bubble Cursor’ yönteminden daha kısa sürede seçim yapılabilmesine imkan verdiği görülmüştür. ‘Uyarlanmış 3B Depth Ray’ ve ‘Uyarlanmış 3B Hibrit Squad’ yöntemleri kullanılarak yapılan seçme işleminde anlamlı bir fark görülmemiştir. Nesne seçim sürelerinin analiz edilmesinden sonra katılımcıların doldürdükleri anket kullanılarak yaptıkları değerlendirmeler analiz edilmiştir. Kullanım kolaylığı, hız ve kesinlik açısından

yapılan yöntem deęerlendirmelerinde anlamlı bir fark bulunmamıştır. Bunun yanı sıra ortamlara göre tercih edilen yöntemlerin analizinde ‘Nesne az yoğun’ ve ‘Nesne hareketli’ ortamlarda anlamlı bir fark bulunmamasına rağmen ‘Nesne çok yoğun’ ortamlarda ‘Uyarlanmış 3B Hibrit Squad’ yöntemi daha fazla tercih edilmiştir.

Tüm veriler birlikte yorumlandığında ‘Uyarlanmış 3B Hibrit Squad’ yöntemi kullanılarak yapılan seçme işlemlerinin diğer yöntemlerden herhangi bir deęerlendirmede geride kalmadığı söylenebilir. Bunun yanı sıra ‘Uyarlanmış 3B Hibrit Squad’ yönteminin ‘Nesne yoğun’ ortamlarda daha hızlı seçme işlemi sağlamış ve daha çok katılımcı tarafından tercih edilmiştir. Tüm durumlar ele alındığında 3B ortamda el serbest seçme işlemi uygulamalarında ‘Uyarlanmış 3B Hibrit Squad’ yönteminin kullanılmasının daha verimli sonuçlar ortaya çıkaracağı görülmüştür.

Tüm bu yöntemlere ek olarak tercih edilen yöntem ile seçme işlemi yapılırken ek olarak Kinect kameranın ses algılama yeteneęi de kullanılabilir. Aynı anda hem seçilen yöntem akışı uygulanırken hem de ses ile alınan belirli komutlar seçimi kolaylaştırmak amacıyla kullanılabilir. Örnek olarak ‘Uyarlanmış 3B Hibrit Squad’ yöntemi kullanılan seviyelerde ekranın sol kısmında listelenen nesnelerin seçilmesi için ses aracılığıyla sayı numarası alınarak seçme işlemi yapılabilir. Buna ek olarak yine sezgisel yöntemlerdeki puanlandırma sistemi de sistemlere entegre edilebilir. Sezgisel seçim sayesinde ‘Uyarlanmış 3B Hibrit Squad’ yöntemindeki sol taraftaki listede bulunan nesnelerin çok hızlı deęişmesi dezavantajı ortadan kaldırılmış olacaktır.

KAYNAKLAR

1. Bowman, D., Kruijff, E., LaViola, J., Poupyrev, I., *3D User Interfaces: Theory and Practice*, Addison Wesley Longman Publishing, Redwood City, **2004**.
2. Dam, P. F., Carvalho, F. G., Braz, P., Raposo, A. B., Haas, A., Hands-Free Interaction Techniques for Virtual Environments, *Proceedings of SBGames 2013*, Sao Paulo, 100-108, **2013**.
3. Bowman, D., Hodges, L., Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *Journal of Visual Languages and Computing*, 10(1), 37–53, **1999**.
4. Yatani, K., Partridge, K., Bern, M., Newman, W. M., Escape: a target selection technique using visually-cued gestures, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, Paris, 285–294, **2008**.
5. Mossel, A., Venditti B., Kaufmann, H., DrillSample: Precise Selection in Dense Handheld Augmented Reality Environments, *Proceedings of the Virtual Reality International Conference: Laval Virtual (VRIC '13)*, Laval, **2013**.
6. Bragdon, A., Ko, H. S., Gesture select: acquiring remote targets on large displays without pointing, *CHI Conference on Human Factors in Computing Systems (CHI '11)*, Vancouver, 187–196, **2011**.
7. Chapuis, O., Labrune, J., Pietriga, E., DynaSpot: SpeedDependent Area Cursor, *CHI Conference on Human Factors in Computing Systems (CHI '09)*, Boston, 1391-1400, **2009**.
8. Wobbrock, J. O., Gajos, K. Z., Goal crossing with mice and trackballs for people with motor impairments: performance, submovements, and design directions, *ACM Trans Access Computing*, New York, 1(1), **2008**.
9. Findlater L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., Wobbrock, J.O., Enhanced area cursors: reducing fine pointing demands for people with motor impairments. *The 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*, New York, **2010**.
10. Zhai, S., Buxton, W., Milgram, P., The “Silk Cursor”: Investigating Transparency for 3D Target Acquisition, *ACM Conference on Human Factors in Computer Systems (CHI '94)*, Boston, 459–464, **1994**.
11. Roth, S. D., Ray casting for modeling solids, *Computer Graphics and Image Processing*, 18, 109-144, **1982**.
12. Liang, J., Green, M., JDCAD: A highly interactive 3D modeling system, 3rd International Conference on CAD and Computer Graphics, Beijing, 18(4), 499-506, **1994**.
13. Ortega, M., Hook: Heuristics for Selecting 3D Moving Objects in Dense Target Environments, *IEEE 8th Symposium on 3D User Interfaces*, Orlando, 119-122, **2013**.
14. Cashion, J., LaViola, J. J., Poster: Dynamic adaptation of 3D selection techniques for suitability across diverse scenarios, *IEEE 8th Symposium on 3D User Interfaces*, 165–166, **2014**.

15. Mine, M., *Virtual environments interaction techniques*, University of North Carolina at Chapel Hill Chapel Hill, Chapel Hill, **1995**.
16. Mine, M. R., Brooks, F. J., Sequin, C., Moving objects in space: exploiting proprioception in virtual-environment interaction, *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH '97)*, Los Angeles, 19–26, **1997**.
17. Steed, A., Parker, C., 3d selection strategies for head tracked and non-head tracked operation of spatially immersive displays, *8th International Immersive Projection Technology Workshop*, 163–170, **2004**.
18. Lee, S., Seo, J., Kim, G., Park, C. Evaluation of pointing techniques for ray casting selection in virtual environments, *Proceedings of SPIE - The International Society for Optical Engineering*, Hangzhou, 38-44, **2002**.
19. Forsberg, A., Herndon, K., Zeleznik, R., Aperture based selection for immersive virtual environments, *Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96)*, New York, 95–96, **1996**.
20. Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., Mine, M., Image plane interaction techniques in 3D immersive environments, *Proceedings of the 1997 symposium on Interactive 3D graphics (I3D '97)*, New York, 39–ff, **1997**.
21. Argelaguet, F., Andujar, C., Trueba, R., Overcoming eye-hand visibility mismatch in 3d pointing selection, *Proceedings of the 2008 ACM symposium on Virtual reality software and technology (VRST '08)*, New York, 43–46, **2008**.
22. Grossman, T., Balakrishnan, R., The design and evaluation of selection techniques for 3D volumetric displays, *Proceedings of the 19th annual ACM symposium on user interface software and technology (UIST '06)*, New York, 3-12, **2006**.
23. Grossman, T., Balakrishnan, R., The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area, *CHI 2005 Conference on Human Factors in Computing Systems (CHI '05)*, Portland, 281-290, **2005**.
24. Vanacken, L., Grossman, T., Coninx, K., Exploring the effects of environment density and target visibility on object selection in 3D virtual environments, *2007 IEEE Symposium on 3D User Interfaces*, Charlotte, 115-122, **2007**.
25. Kopper, R., Bacim, F., Bowman, D. A., Rapid and accurate 3D selection by progressive refinement, *Proceedings of IEEE Symposium on 3D User Interfaces*, Singapore, 67-74, **2011**.
26. Grossman, T., Balakrishnan, R., Pointing at trivariate targets in 3D environments, *Proceedings of the SIGCHI conference on human factors in computing systems (CHI '04)*, New York, 447–54, **2004**.
27. Guimbretiere, F., Nguyen, C., Bimanual marking menu for near surface interactions, *Proceedings of the 2012 ACM annual conference on human factors in computing systems (CHI '12)*, New York, 825–828, **2012**.
28. Cashion J., Intelligent selection techniques for virtual environments, Doktora Tezi, Engineering and Computer Science at the University of Central Florida, Florida, **2014**.
29. <http://www.xbox.com/tr-tr/Search?q=Kinect+Sports> (Nisan **2016**).

30. Argelaguet, F., Andujar, C., A survey of 3D object selection techniques for virtual environments, *Computers and Graphics*, 37(3), 121-136, **2013**.
31. <https://en.wikipedia.org/wiki/Kinect> erişim tarihi (Nisan **2016**).
32. <https://msdn.microsoft.com/en-us/library/aa187916.aspx> (Nisan **2016**).
33. <https://msdn.microsoft.com/en-us/library/jj131033.aspx> (Nisan **2016**).
34. <https://developer.microsoft.com/en-us/windows/kinect/hardware> (Nisan **2016**).
35. <https://msdn.microsoft.com/tr-tr/library/dn782041.aspx> (Nisan **2016**).
36. <http://unity3d.com/unity/multiplatform/> (Mayıs **2016**).
37. <http://docs.unity3d.com/Manual/LearningtheInterface.html> (Mayıs **2016**).
38. Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T., The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR, *Ninth Annual Symposium on User Interface Software and Technology (UIST '96)*, Seattle, 79–80, **1996**.
39. Dam, P., Braz, P., Raposo, A., *Virtual Augmented and Mixed Reality, Designing and Developing Augmented and Virtual Environments*, Springer Berlin Heidelberg, Berlin, 139–148, **2013**.
40. <https://msdn.microsoft.com/en-us/library/dn785530.aspx> (Mayıs **2016**).
41. Fitts, P.M., The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement, *Journal of Experimental Psychology*, 47(6), **1954**.
42. MacKenzie, S., Buxton, W., Extending Fitts' law to two-dimensional tasks, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Monterey, 219–226, **1992**.
43. MacKenzie, I. S., Fitts' Law as a Performance Model in Human-Computer Interaction, Doktora Tezi, University of Toronto, Ontario, **2002**.

EK-1: KULLANICI DEĞERLENDİRME ANKETİ

Katılımcı Numarası: _____

Tarih: ___/___/_____



HACETTEPE ÜNİVERSİTESİ

BİLGİSAYAR GRAFİĞİ ANABİLİM DALI

YENİ NESİL KINECT KAMERA İLE 3B ORTAMLARDA NESNE SEÇME TESTİ
KULLANICI DEĞERLENDİRME ANKETİ

DEMOGRAFİK SORULAR

Ad Soyad: _____

Eğitim: _____

Cinsiyet(Bir tanesini yuvarlak içine alınız): Erkek / Kadın

Yaş: _____

Boy: _____

TECRÜBE SORULARI

1. Ne kadar sıklıkla video oyunu oynarsınız.
a) Hiç
b) Nadiren
c) Haftada bir
d) Birkaç günde bir
e) Hemen hemen her gün
2. Oynadığınız video oyunlarının yüzde kaçını 3B ortamlar için geliştirilmiş oyunlardır.
a) 0%
b) 25%
c) 50%
d) 75%
e) 100%
3. Oynadığınız 3B ortam için geliştirilmiş video oyunlarının yüzde kaçında uzaktan oyun kontrolüne izin veren kısımlar mevcuttur. (Wii, Kinect, Playstation kamera vb.)

- a) 0% b) 25% c) 50%
- d) 75% e) 100%

4. Oynadığınız 3B ortam için geliştirilmiş uzaktan kontrole izin vere video oyunlarının yüzde kaçında Kinect kamera ile oynadınız.

- a) 0% b) 25% c) 50%
- d) 75% e) 100%

5. Oyun oynama becerinizi nasıl sınıflandırırsınız.

- a) Başlangıç b) Amatör c) Orta
- d) İyi e) Çok İyi

TECRÜBE SORULARI

1. Uygulamadaki tecrübe düşünüldüğünde Kinect kamera imleci hareket ettirmekte ne kadar yeterlidir.

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

2. Uygulamadaki tecrübe düşünüldüğünde Kinect kamera nesne seçmede ne kadar yeterlidir.

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

3. ‘Uyarlanmış Bubble Cursor’ yöntemini(1. 2. ve 3. Seviyeler) belirtilen özelliklere göre nasıl puanlandırırınız.

a. Kullanım Kolaylığı

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

b. Hız

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

c. Kesinlik

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

4. ‘Uyarlanmış Depth Ray’ yöntemini(4. 5. ve 6. Seviyeler) belirtilen özelliklere göre nasıl puanlandırırınız.

a. Kullanım Kolaylığı

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

b. Hız

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

c. Kesinlik

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

5. ‘Uyarlanmış Hibrit Squad’ yöntemini(7. 8. ve 9. Seviyeler) belirtilen özelliklere göre nasıl puanlandırırsınız.

a. Kullanım Kolaylığı

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

b. Hız

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

c. Kesinlik

Yetersiz 1 2 3 4 5 6 7 8 9 10 Yeterli

6. Nesne az yoğun ortamlar(1. 4. ve 7. Seviyeler) düşünüldüğünde hangi yöntemi tercih edersiniz.

a) Uyarlanmış Bubble Cursor

b) Uyarlanmış Depth Ray

c) Uyarlanmış Hibrid Squad

7. Nesne yoğun ortamlar(2. 5. ve 8. Seviyeler) düşünüldüğünde hangi yöntemi tercih edersiniz.

a) Uyarlanmış Bubble Cursor

b) Uyarlanmış Depth Ray

c) Uyarlanmış Hibrid Squad

8. Nesne hareketli ortamlar(3. 6. ve 9. Seviyeler) düşünüldüğünde hangi yöntemi tercih edersiniz

a) Uyarlanmış Bubble Cursor

b) Uyarlanmış Depth Ray

c) Uyarlanmış Hibrid Squad

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Ömer Faruk Çangır

Doğum Yeri : Sakarya

Medeni Hali : Evli

E-posta : omerfarukcangir@gmail.com

Adresi : Turgut Özal Mah. Olimpiyatköy Sit. No:4E/13 Yenimahalle / Ankara

Eğitim

Lise : Akyazı Anadolu Lisesi

Lisans : Hacettepe Üniversitesi Bilgisayar Mühendisliği

Yüksek Lisans: Bilgisayar Animasyonu ve Oyun Teknolojileri

Yabancı Dil ve Düzeyi

- İngilizce : İyi

İş Deneyimi

- 3 Yıl

Deneyim Alanları

- Yazılım Mühendisliği, Oyun Teknolojileri

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

-

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

-