



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

HAREKET VE SES ALGILAMA İLE FATİH PROJESİ ETKİLEŞİMLİ TAHTALARI KULLANMA

Salih GÜNDÜZ

YÜKSEK LİSANS

Bilişim Teknolojileri Mühendisliği Anabilim Dalı

Ağustos-2017

KONYA

Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS

HAREKET VE SES ALGILAMA İLE FATİH PROJESİ ETKİLEŞİMLİ TAHTALARI KULLANMA

Salih GÜNDÜZ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilişim Teknolojileri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Hakan IŞIK

2017, 93 Sayfa

Jüri
Prof. Dr. Hakan IŞIK
Doç. Dr. Fatih BAŞÇİFTÇİ
Yrd. Doç. Dr. Uğur FİDAN

Günümüzde teknolojinin ilerlemesiyle eğitimde teknoloji kullanımı yaygınlaşmaya başlamıştır. Milli Eğitim Bakanlığı'da bu ihtiyacı karşılamaya yönelik Fatih Projesini başlatmış ve okullara etkileşimli tahtalar kurmuştur. Projede sınıflara dokunmatik ekranlı, Windows İşletim Sistemi ve internet bağlantısı olan etkileşimli tahtalar getirilmiştir. Fakat öğretmenler dokunmatik ekranı kullanmakta zorlanma, tahtaya bağlı kalma, sınıf içerisinde özgürce dolaşamama ve öğrenciye çoğunlukla arkasını dönme gibi sorunlar yaşamaktadırlar. Bu sorunları azaltmak, sınıf yönetimini kolaylaştırmak, öğrencilerin ders katılımını arttırmak ve engelli öğrencilerimizi de öğrenme sürecinin aktif bir parçası haline getirmek için bu çalışma yapılmıştır. Çalışmamızda hareket ve ses sensörleri kullanılarak etkileşimli tahtayı uzaktan herhangi bir kontrol aygıtına gerek duymadan yönetme işlevi gerçekleştirilmiştir. Gerçekleştirilen uygulama sınıf ortamında denenmiş ve öğretmen görüşleri kayıt altına alınmıştır. Öğretmen görüşlerinden yola çıkarak uygulama eğitim bilimleri açısından değerlendirilmiştir.

Anahtar Kelimeler: Eğitim, Fatih Projesi, Kinect, Teknoloji.

ABSTRACT

MS
USING FATIH PROJECT INTERACTIVE WHITEBOARDS BY MOTION AND SOUND SENSORS

Salih GÜNDÜZ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE INFORMATİON TECHNOLOGİES ENGINEERING

Advisor: Prof. Dr. Hakan IŞIK

2017, 93 Pages

Jury
Prof. Dr. Hakan IŞIK
Assoc. Prof. Dr. Fatih BAŞÇİFTÇİ
Asst. Prof. Dr. Uğur FİDAN

Nowadays, using technology in education spreads by the improvements of the technology. Ministry of Education started FATIH project to meet this need and established interactive boards to schools. By this project interactive boards with touch screen and high speed internet connection were set up in the classrooms. Despite all these developments, teachers may have problems like using touch screen, adherence to the board, turn back to students and not moving freely in the classroom. This work was done for reduce such problems, facilitate classroom management, increase the course attendance of students, making our disabled students an active part of the learning process. In our study, using interactive whiteboards is achieved by using motion and sound sensors without any control device . Develeoped application was tried in the classrooms and teachers opinions recorded. From the teacher's point of view, application was evaluated in terms of educational sciences.

Keywords: Education, Fatih Project, Kinect, Technology.

ÖNSÖZ

Beni yüksek lisans yapmaya teşvik eden, çalışmalarımda bana yol gösteren Saygıdeğer Danışmanım Prof. Dr. Hakan IŞIK'a, fikirleriyle ufkumu açan benden yardımlarını esirgemeyen Öğr.Gör. Dr. Mustafa Nevzat ÖRNEK'e, akademik çalışmaların nasıl yapılacağı konusunda bana yol gösteren Doç. Dr. Fatih BAŞÇİFTÇİ'ye, verdiği dersler ve yaptığımız birebir çalışmalarla tezimi yapabilecek seviyeye gelmemde büyük katkısı olan Yrd. Doç. Dr. Hasan Erdinç KOÇER'e ve aileme sevgi ve teşekkürlerimi sunarım.

Salih GÜNDÜZ
KONYA-2017

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

3. MATERYAL VE YÖNTEM **7**

3.1. Kinect V2 Sensör 7
3.1.1. Kinect sensörün iç kısmı 8
3.1.2. RGB kamera 9
3.1.3. Kızılötesi ışık verici ve kızılötesi derinlik algılama alıcısı 9
3.1.4. Eğim motoru 10
3.2. Kinect 2 Windows SDK 10
3.2. Fatih Projesi Etkileşimli Tahta 20
3.3. Yazılımın Geliştirilmesi 23
3.3.1. Resim görüntüleyici 25
3.3.2. PDF kitap görüntüleyici 26
3.3.3. Video oynatıcı 26
3.3.4. Sunu gösterici 31

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA **32**

4.1. Kurulum ve erişim 32
4.2. Öğretmen görüşleri 33
4.2.1. Örneklem seçimi 34
4.2.2. Verileri toplama araçları 36
4.2.3. Araştırma sonucu elde edilen bulgular 36

5. SONUÇLAR VE ÖNERİLER **46**

5.1 Sonuçlar 46
5.2 Öneriler 48

KAYNAKLAR **50**

EKLER **53**

ÖZGEÇMİŞ.....93

SİMGELER VE KISALTMALAR

Simgeler

m	:Metre
ms	:Mili Saniye
kb/s	:Kilobyte/Saniye
Ö	:Öğretmen

Kısaltmalar

MEB	:Milli Eğitim Bakanlığı
PDF	:Portable Document Format
RGB	:Red Green Blue-Kırmızı yeşil mavi
IR	:Infraded-Kızıl Ötesi
FPS	:Frame per second
CMOS	:Complementary Metal Oxide Semiconductor
SDK	:Software development kit

1. GİRİŞ

Günümüzde eğitimde teknoloji kullanımı önem kazanmıştır. Artık öğretmen merkezli eğitimin yerini, öğretmenin rehber olduğu, öğrencinin de sınıf içerisinde daha aktif ve katılımcı olduğu, teknolojik araçlarla eğitimin veriminin artırıldığı bir eğitim sistemi almaktadır. Milli Eğitim Bakanlığı Fatih Projesi de bu alanda yapılan önemli bir çalışmadır. Proje sayesinde Milli Eğitim Bakanlığı'na bağlı okullarda her sınıfa bir etkileşimli tahta kurulmuş, öğrencilere tabletler dağıtılmıştır. Birçok okulumuzda kurulum gerçekleşmiştir. Etkileşimli tahtalar dokunmatik ekrana sahiptir. Bu özellik birçok kolaylık sağlamaktadır fakat öğretmen tahtayı kullanırken tahtanın başında kalmakta ve çoğu zaman öğrenciye arkasını dönmek zorunda kalmaktadır. Bu durum öğretmenin sınıf yönetimi ve beden dili kullanımını kısıtlanmaktadır. Biz bu çalışmamızda Kinect Sensörün etkileşimli tahtalarda kullanılmasını sağlayarak, öğretmenlerin akıllı tahtayı etkin bir şekilde kullanmasıyla sınıf yönetimini kolaylıkla yapabilmesini sağlamayı ve ortopedik sorunlarını azaltmayı, öğrencilerin oturdukları yerden tahtayı kullanmasına imkan vererek ders içi katılımlarını artırmayı ve engelli öğrencilerimizin ders katılımını artırmayı amaçladık. Bu çalışma için Microsoft Kinect sensörden faydalandık. Sınıf içerisinde sık kullanılan PDF kitapları, resimleri, sunuları ve videoları gösterip uzaktan kontrol etmeyi planladık. Daha sonra uygulamayı sınıf içerisinde deneyip öğretmen izlenimlerini kayıt edip değerlendirmesini yaptık.

Eğitimde istendik davranışlar meydana getirebilmek için sınıf yönetimi büyük önem taşımaktadır. Kaliteli bir eğitimin verilebilmesi büyük ölçüde sınıf yönetimine bağlıdır (Paliç ve Keleş, 2011). İyi bir sınıf yönetimi öğrenme düzeyini artırırken sınıfta oluşabilecek olumsuzlukların da önüne geçmeyi sağlayabilecektir. İyi bir sınıf yönetimi ise beden dilini etkili bir şekilde kullanmayı ve sınıf içi iletişimi ve etkileşimi artırmayı zorunlu kılmaktadır. Verimli bir iletişim kurabilmek için sözcüklerin yanında beden dilini de etkili bir biçimde kullanmak gerekmektedir. Öğretmenin Fatih Projesi Etkileşimli Tahtaları kullanırken tahtanın başında beklemesi ve sınıfa arkasını dönmesi sınıf yönetimini olumsuz etkilemektedir. Öğretmen bu durumda beden dilini etkili bir şekilde kullanamamakta ve sınıf yönetimini kaybedilmektedir. Öğretmenin sınıf içerisinde ki konumu nerede duracağı yapacağı etkinlikte önem taşımaktadır. Ancak doğru zamanda doğru yerde durarak ve doğru etkileşimi sağlayarak etkinliğin faydası artırılabilir. Öğretmen sınıfın tamamına hakim olabilmek isterken tüm öğrencilerin de öğretmeni ve beden dilini rahatlıkla görmesi gerekir. Öğretmen sınıfın tümünü

görebilmek için tahtayı kullanırken bile arkasını dönmemeye çalışmalıdır (Şen, 2006). Eğitimde yeni değerlere göre vurgu öğrencidedir (Özden, 2005). Öğrenme ve öğretmede öğretmen rehber olmalı ve öğrencilerde sürece etkin olarak katılarak bilgilerden yeni bilgiler üretebilmelidir. Bizim çalışmamız kinect sensör sayesinde sınıf yönetimini kolaylaştırırken eğitimi daha eğlenceli hale getirip öğrenci katılımını da arttırabilecektir.

2. KAYNAK ARAŞTIRMASI

Literatürde, kinect sensörler, eğitimde teknoloji kullanımı, etkileşimli tahtalar, engelli öğrencilerin eğitimi, sınıf yönetimi, öğretmenlerde ortopedik problemler vb. konular üzerinde araştırmalar yapılmıştır. Bir oyun kontrol aygıtı olarak geliştirilen kinect sensörü daha sonraları robotik ve sanal gerçeklik gibi alanlarda da kullanılmaya başlanmasıyla parmak hareketleri algılama üzerine çalışmalarda yapıldı (Frati ve Prattichizzo, 2011). Yapılan çalışmada her parmağın konumu sensör tarafından okunup hareketler tanımlanmıştır. Yapılan uygulamanın deneme aşamasında işlemcinin %50'yi geçtiği nadiren görülmüştür. Alkış, telefonla konuşma, el sallama gibi bir çok hareketin tanımlanması farklı uygulamalar geliştirilmesine imkan sağlamıştır (Biswas ve Basu, 2011). Bu çalışmada derinlik görüntülerinin el jestlerini tanımada kullanılabileceği görülmüştür. Bu görüntülere ek olarak renkli görüntülerden de faydalanıldığında başarının artabileceği belirtilmiştir. Kang ve ark. (2011) kinect sensörün vücudun hangi eklem bölgelerinin tespit edilebildiğini çalışmalarında belirtmişlerdir. Bu şekilde bir fare işaretçisi algoritması geliştirmişlerdir. Sensörün eğitim alanında ki potansiyelleri hakkında da araştırmalar yapılmıştır (Hsu, 2011). Kinect sensörün insan iskeletini ve eklemleri izleyebilme yeteneği sağlık ve rehabilitasyon alanında da kullanılmalarını sağlamıştır (LaBelle, 2011). 2011 yılında yapılan kinect sensörün derinlik verilerinin doğruluğu üzerinde yapılan deneysel araştırmalarda sensörden uzaklaşıldıkça piksel yoğunluğunun azaldığı gözlenmiştir. Derinlik hesaplamalarında hesaplanan rastgele hatalar sensörden uzaklaştıkça artarken maksimum 4 cm ye kadar çıkmaktadır (Khoshelham, 2011). Jana (2012) yazdığı kitapta sensörün tüm fonksiyonlarından bahsetmiş örnekler vermiştir. Bu alandaki detaylı açıklamalar bulunmaktadır. Sensörün donanım özelliklerini detaylıca göstermiştir. Sitem gereksinimlerini belirtmiş ve bir uygulama baştan sona nasıl geliştirilir anlatmıştır. SDK indirilmesi kurulması ve sistemin hazır hale getirilmesi gösterilmiştir. Sensörlerin test edilmesi anlatılmıştır. SDK içerisinde ki bütün fonksiyonlar detaylıca değerlendirilmiştir. Derinlik ölçme uygulamaları geliştirilmiş denenmiştir. İnsan iskeleti tanıma, ses algılama, hareket algılama uygulamaları nasıl geliştirilir detaylıca anlatılmıştır. Andersen ve ark. (2012) kinect sensörünün gelişimi hakkında bilgi vermiştir. Derinlik algılama sensörü üzerinde araştırmalar ve derin bir değerlendirme yapmışlardır. 3 boyutlu nesne algılama ile ilgili karşılaşılan problemler üzerinde durmuşlardır. Çalışmalarında yansıtıcı nesnelerin

uzaklığının hesaplanamadığını ve çok güneş ışığı olan yerlerde sensörün verimli çalışamadığını belirtmişlerdir. Kinect v2 bütün ölçümlerde daha iyi sonuçlar vermiştir. Bu hata payı kabul edilebilecek seviyededir. Pedersoli ve ark. (2012) insan el jestlerini tanımlamışlardır. Kafa pozisyonları ve yüz ifadeleri tanımlama da kinect sensörü ile mümkün hale gelmiştir. Microsoft Avatar Kinect de bu teknolojiyi benimsemiştir. Microsoft Avatar Kinect ile insanların yüz ifadeleri avatarlarına aktarılmıştır (Zhang, 2012). Süzen ve Taşdelen (2013) sensörün iskelet algıma işlevi üzerine çalışmıştır. Bu özellikten faydalanarak bir piyano uygulaması geliştirmişlerdir. Bu çalışma sensörün müzik alanında da kullanıldığını göstermiştir. Ayrıca yapılan araştırmalarda kinect sensör ile yapılan 3 boyutlu yüz tanıma uygulamaları da başarılı olmuştur (Li ve ark., 2013). Bu çalışmada geliştirilen algoritma ile 5000'üzerinde görüntü işlenmiştir. Yüz görüntüleri farklı pozlarda ve farklı ifadelerle alınmıştır. Çalışmada kinect ile elde edilen 3 boyutlu görüntülerin çok gürültülü olduğu fakat yüz tanımada kullanılabildiği görülmüştür. Yapılan deneylerde RGB görüntülerde %96.7 başarıyla tanıma yapılırken 3 boyutlu gürültülü görüntülerde %88.7 başarı ile tanıma gerçekleştirilmiştir. 2014 yılında sensörün ikinci versiyonu piyasaya sürülmüştür. Bu yeni sensörde Time-of-Flight kamera kullanılmaya başlandı. Yeni sensörde çözünürlükler, mesafe, ses akışı ve kalitesi gibi özellikler arttırıldı. Yeni sensörde deneylerle yapılan ölçümlerde sapmalar 0.5-2.7 cm arasında ölçülmüştür (Lachat ve ark., 2015). Amon ve ark. (2014) donanım olarak kinect sensörleri incelemiş ve V1 ve V2 versiyonları arasında karşılaştırma yapmıştır. Yüz algılama uygulamalarında sensörleri karşılaştırmış ve kafa rotasyonundaki verimlerini incelemişlerdir. Kinect v2'nin işaretçi(pointer) doğruluğu değerlendirilmiş doğruluk oranının sensörün bakış açısına bağlı olduğu saptanmıştır (Fürntratt ve Neuschmied, 2014). Kinect v1 ve v2 derinlik görüntüleri karşılaştırıldığında Kinect v1 de ancak birkaç görüntü alındıktan sonra güvenilir görüntülere ulaşılabilirken v2 daha kaliteli görüntülere daha erken ulaşılabilmektedir. Kinect V1 de mesafe artarken doğruluk azalırken fakat v2 de doğruluğun daha sabit değerlere sahip olduğu görülmüştür.

Avcı ve Seferoğlu (2011) öğretmenlerde tükenmişliğin en önemli sebeplerinden birisinin teknoloji kullanımındaki eksiklikler olduğunu belirtmişlerdir. Bilgi toplumunda hedefin teknolojiyi etkili kullanan olumlu gelişmelere açık bireyler yetiştirmek olduğu belirtilmiştir. Ülkemizde teknolojinin eğitime girişinin 1970'lerden itibaren başladığını söylemişlerdir. Tekonoloji politikaları doğrultusunda ki planlara bakıldığı zaman çoğu kurum bilgi ve iletişim teknolojisine yönelik yatırım yapmasına

rağmen beklenen gelişme son bir kaç yıla kadar henüz gerçekleşmemiştir. Eğitimde "Fırsatları Artırma ve Teknolojiyi İyileştirme Hareketi" yani FATİH projesi bilişim teknolojilerinin eğitimde daha etkin olarak kullanılması için gerçekleştirilen bir projedir. Proje 2010 Yılının Kasım Ayı'nda başlamıştır (Akıncı ve ark., 2012). Vizyon 2023 hedefiyle başlayan proje teknolojiyi bilinçli kullanan ve refah seviyesi yüksek bir toplum amaçlamaktadır. Bu hedeflere ulaşmada en büyük sorumluluğun uygulayıcı olan öğretmenlere düştüğü belirtilmiştir. Pamuk ve ark. (2013) projenin amacı hakkında bilgi vermiştir. Proje sayesinde eğitim daha görselleştirilebilir hale getirilmiştir. Her sınıfta internet ve akıllı tahta olması hem okul yönetimlerinin hem de öğretmenlerin işlerini kolaylaştırmıştır. Keser ve Çetinkaya (2013) eğitimde teknoloji kullanımının etkisini çalışmışlardır. Altan ve Tüzün (2011) Fatih projesinde zengin öğrenme biçimlerinin yeri üzerine bir çalışma yapmıştır. Bilgisayar oyunlarının eğitimde kullanılabilirliğini incelemiştir. Quest Atlantis adlı eğitim amaçlı geliştirilmiş çevrimiçi bir rol oynama oyunu denenmiştir. Son olarak Fatih projesinde getirlmiş olan yeniliklerin sonuna kadar kullanılmadığı takdirde para ve zaman kaybına sebep olacağı belirtilmiştir. Yapılan araştırmalarla okul yöneticilerinin projeden memnun oldukları gözlenmiştir (Dursun ve ark., 2013). Öğretmenler proje sayesinde ders aktarımlarının hızlandığını, kazanımları daha rahat verdiklerini belirtirken göz temasının azaldığını ve sınıf yönetimin zorlaştığını belirmişlerdir (Kurt ve ark., 2013).

Paliç ve Keleş (2011) 42 öğretmen ile yaptıkları araştırmada sınıf yönetimine ilişkin sorunları analiz etmişlerdir. İlköğretim ve ortaöğretim kademelerinde çalışan öğretmenlerin etkili sınıf yönetimine ilişkin görüşleri üzerinde analizler yapılmıştır. Etkili sınıf yönetimi doğrultusunda karşılaşılan sorunlar incelendiğinde ilk sıralarda iletişim problemleri geldiği sunulmuştur. Derse karşı isteksizlik ve motivasyon düşüklüğü de sınıf yönetimini olumsuz etkilemiştir. Erdoğan ve ark. (2010) bilişim teknolojileri öğretmenlerinin sınıf yönetimine ilişkin karşılaştıları sorunları araştırmışlardır. Yaptıkları çalışmada sorunları, ilgi ve motivasyon eksikliği, kural ve rutin ihlali, altyapı eksikliği, zaman yönetimi ve sınıf içi iletişim ekskiliği olarak sıralamıştır. Çalışmaya göre yerleşim düzeni ve oturma planı iletişim güçlüğüne neden olmuştur. U tipi oturma düzeninin bu sorunu azaltacağını savunmuştur. Şen (2006) öğretmenin sınıf yönetiminde duruşunun nasıl olması gerektiğini belirtmiştir.Özden (2005) eğitimde öğrenme ve öğretme kuramlarında yenilikleri incelemiştir. Zeka üzerine yeni bir bakış açısı olan çoklu zeka kuramı üzerinde durmuştur. Eğitimde geleneksel değerler ve yeni değerleri karşılaştırmış ve dönüşümün gerekliliğine işaret

etmiştir. Bilgi toplumu kavramını anlatmış ve bilgi toplumu olmak için neler yapılması gerektiğinden bahsetmiştir.Albayrak (2014) etkileşimli tahtaların eğitime etkisini çalışmasında göstermiştir. Çalışmasındaki rapor incelendiğinde Fatih Projesi bir yenilik olmasına rağmen projenin sınıf oturma düzeninde ve yerleşiminde pek bir değişime neden olmadığı saptanmıştır. Yine raporlar incelendiğinde tahtaların dersi ilgi çekici hale getirdiği ve derse katılımı artırdığı gözlenmiştir. Projenin zaman yönetimine faydalarından da bahsedilmiştir. Ders öncesi planlama zamanları kısalmış, materyaller hazır bulunduğundan derse giriş daha kısa sürelere inmiştir. Sınıf ikliminin olumlu yönde etkilediği de çalışmada gösterilmiştir. Öğretmenler etkileşimli tahtayı kullanmaya olumlu yaklaşmışlardır. Etkileşimli tahtaların uygulamaları incelendiğinde ise tüm yazılımların etkileşimsiz olduğu ve projeksiyon ile yapılabilecek uygulamalarda kullanıldığı saptanmıştır. Bu yüzden yeni uygulamalar ve e-içerikler geliştirilmesi önerilmiştir.

Adigüzel ve ark. (2011) akıllı tahtaların engelli öğrencilere sağladığı katkıyı belirtmiştir. Ayrıca akıllı tahtalarda yapılan eğitimde bir gruba hitap etmenin daha kolay olduğu, grup etkinliklerinin daha rahat yapılabildiği belirtiliştir. Özel eğitime ihtiyacı olan öğrencilerin sözel iletişim kurmalarında sorunlar olduğu için etkileşimli tahtalar sayesinde onların dikkati daha çok çekilebilmektedir. Bunun önüne geçebilmek için teknolojiyi derse entegre ederek herkese hitap eden bir ders yapılması gerektiği görüşü savunulmuştur. Ayrıca bu çalışmada akıllı tahtaların engelli öğrenciler üzerinde ki etkisini ölçmek için aynı tür özelliklere sahip 3 birey üzerinde fonksiyonel kelime becerilerini artırma üzerine yapılan çalışmada, tahta üzerinde gösterilen 4 tane ürünün adlarını tahta üzerinde dokunma yoluyla göstermesi istenmiştir. Çalışmada gözlemleyerek ve etkileşime geçerek öğrenmenin önemine değinilmiştir.

3. MATERYAL VE YÖNTEM

Bu tez çalşmasında donanım olarak bir Kinect V2 Sensör ve MEB Fatih Projesi ile sınıflara getirilmiş olan bir etkileşimli tahta kullanılmıştır. Kinect for Windows veya Xbox One Kinect Sensör denilen sensörlerden birisi kullanılabilir. Yazılım Visual Studio ortamında C# dili kullanılarak geliştirilmiştir. Etkileşimli tahtalarda, uzaktan herhangi bir kontrol aygıtı olmadan sınıf içerisinde kullanılan dökümanlar açılıp yönetilmiştir. Uygulamanın etkisi, olumlu ve olumsuz yönleri öğretmen görüşleri yarı yapılandırılmış görüşme formları ile alınarak sonuçları rapor edilmiştir.

3.1. Kinect V2 Sensör

Sensör ilk olarak Microsoft tarafından Xbox 360 Oyun Konsolu için geliştirilmiştir. Çok başarılı olan bu sensör 10 milyondan fazla satılmıştır (Andersen ve ark., 2012). Sensör artık Xbox One için üretilmektedir ve bunlar Windows uyumludur. Bu sensörün amacı klavye ve mouse gibi herhangi bir kontrol aygıtı kullanmadan uzaktan el, kol hareketleriyle ve sesli komutlarla bilgisayarların kontrol edilmesini sağlamaktır. Tablo-1'deki donanım özelliklerine ilaveten Windows'un yayınlamış olduğu Kinect for Windows SDK 2.0 sayesinde birçok işlem gerçekleşmektedir. Renkli görüntü alma, derinlik görüntüleri üzerinde çalışma, insan iskeletini izleme ve eklem noktalarının konumunu ve hareketini görme, insan jest ve mimiklerini tanıma gibi işlemler gerçekleştirilebilmektedir.

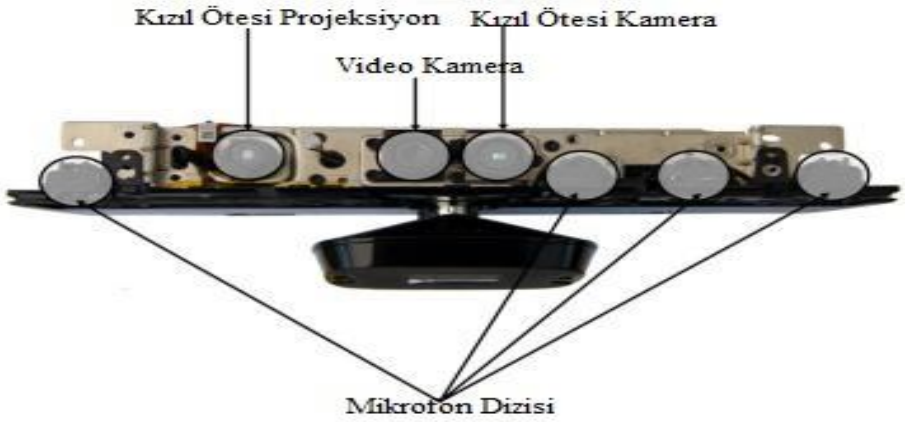
Daha çok oyun alanında kullanılan bu sensör bilgisayarlarda da çalışabilmektedir. Bu sayede Şekil 3.1.'de gösterildiği gibi eğitim, sağlık, alışveriş, eğlence ve spor gibi birçok alanda kullanılabilmektedir.



Şekil 3.1 Kinect Kullanım Alanları

3.1.1. Kinect sensörün iç kısmı

Haziran 2011'de Microsoft bir yazılım geliştirme kiti(SDK) yayınlayarak bilgisayarlarla görme alanında sensöre ilgiyi artırmıştır. Sensör 3 boyutlu görmeyi sağlamaktadır. Şekil 3.2.' de gösterildiği gibi cihazda 640*480 piksellik bir derinlik kamerası, 640*480 RGB kamera, kızıl ötesi ışık yayan bir projeksiyon, ve 4 kanaldan oluşan bir mikrofon dizisi bulunmaktadır. Bu sayede sensör tüm vücut hareketlerini ve sesleri tanımlayabilme yeteneği sağlamaktadır (Zhang, 2012).



Şekil 3.2 Açık Kinect Sensörü

Sensörde bulunan mikrofon dizisi kaliteli bir ses algılama sağlarken aynı zamanda sesin yönünü de tayin etmeyi mümkün kılar. 2014 yılında Kinect'in ikinci versiyonu da piyasaya sürülmüştür. Birinci versiyon ile ikinci versiyon arasında ki farklar Çizelge 3.1.'de karşılaştırılmıştır (Amon ve ark., 2014).

Çizelge 3.1 Kinect V1 –V2 Karşılaştırması

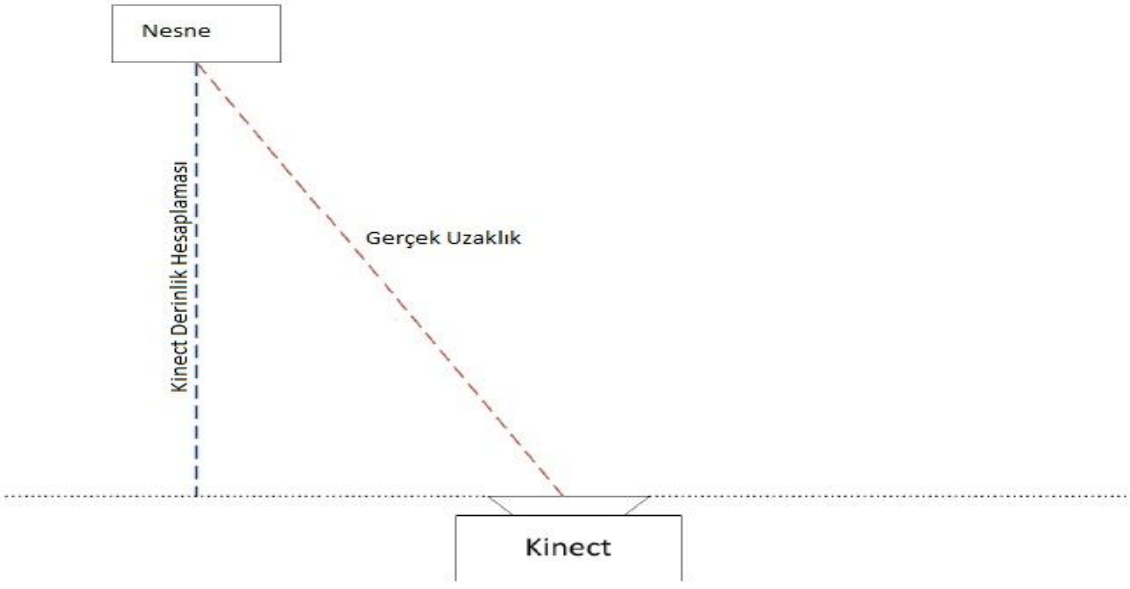
Bileşen	V1	V2
Teknik	Yapılandırılmış Işık	Uçuş Süresi
Derinlik Sensoru	1.8 to 3.5 m	1.3 to 3.5 m
IR Derinlik Görüntü	320*240	512*424
Renkli Görüntü	640*480	1920*1080
Kızıl Ötesi Görüntü	No IR	512*424
Yatay Görüş Açısı	57 Derece	70 Derece
Dikey Görüş Açısı	43 Derece	60 Derece
Minimum Gecikme	102 ms	20-60 ms
Ses Akışı	16 kHz, 16-bit	48 kHz,16-bit

3.1.2. RGB kamera

Bu kamera renkli görüntü ve video kaydı yapmak için kullanılır. Günümüzde kullanılan web kameralarının işlevini gerçekleştirir. Bu kameradan elde edilen görüntüler görüntü çerçevesi olarak bilgisayara aktarılır. Bu çerçeve (frame) akratırımı maksimum 30 FPS dir. 640*480 piksel çözünürlükte görüntüler elde edilebildiği gibi daha yüksek çözünürlükte görüntülerde elde edilebilir. Çözünürlük arttıkça FPS oranı düşer. 30 Hz ile çalışan RGBD kamera 8-bit kanalda 640*480 piksel renkli görüntüler alabilmektedir. İstenilirse FPS (frame per second) 10'a düşülürerek çözünürlüğü 1280*1024'e arttırılabilir (Cruz ve ark., 2012).

3.1.3. Kızılötesi ışık verici ve kızılötesi derinlik algılama alıcısı

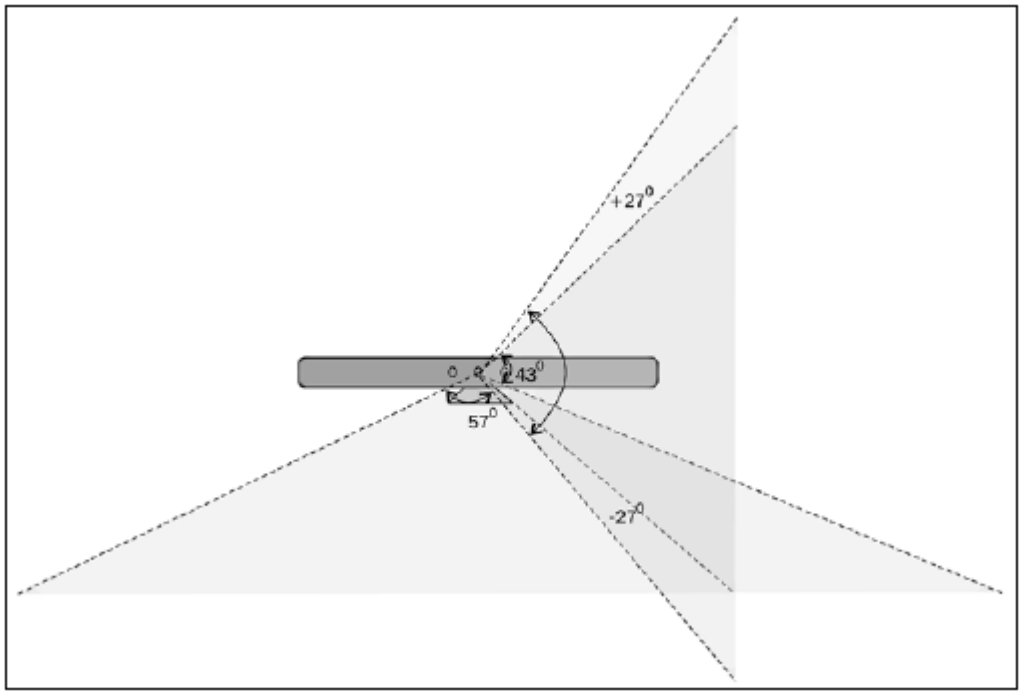
Cihazın klasik kameralara göre en büyük gücü derinlik ölçebilmesidir. Derinlik sensörü sayesinde 3 boyutlu nesnelerin x, y, z koordinatlarına erişişebilir. Cihazdan gönderilen kızıl ötesi sinyaller nesnelere çarpıp döndükten sonra derinlik bilgisi bilgisayarlara aktarılır. Nesnelerin uzaklıkları sensörün bulunduğu düzleme olan dik uzaklığı ile Şekil 3.3.'deki gibi hesaplanır (Andersen ve ark., 2012). Kızıl ötesi ışıklar insanlar tarafından görülemezken CMOS kameralar bu ışıklardan gelen veriyi okuyarak bilgisayara bilgi olarak gönderirler. Bu sayede derinlik ölçülmüş olur. Kameranın bir diğer güçlü yanı ise kızıl ötesi ışıkla çalıştığı için ortamda ki ışık düzeyinden etkilenmemesidir.



Şekil 3.3 Kinect Uzaklık Hesaplaması

3.1.4. Eğim motoru

Sensörün koyulacağı düzlem her zaman düz olmayabilir. Bazen eğimli alanlara da koyulabilir. Bu eğimden kaynaklanacak sorunları ortadan kaldırmak için cihaza bir eğim motoru yerleştirilmiştir. Cihaz eğimli olsa bile bu motor sayesinde açı düzeltilir ve insan iskeleti doğru bir şekilde tanımlanır. Cihaz Şekil 3.4.'deki gibi 27 derece yukarı ve 27 derece aşağı döndürülebilir.



Şekil 3.4 Döndürme Motoru Açıları

3.2. Kinect 2 Windows SDK

Ekim 2014'te yayınlanan kinect yazılım geliştirme aracı Kinect Sensörleri kullanarak hareket ve ses algılamayı desteklemektedir. Bu sürümle birlikte uygulamalar geliştirilip Windows Store'da yayınlanabilir. Bu araç sayesinde donanım kullanılıp özelliklerinden faydalanılabilir. İçersinde bulundurduğu fonksiyonlar sayesinde ses algılama, insan iskeletini tanıma gibi bir çok işlem kolaylıkla yapılabilmektedir. Kinect SDK kütüphanesi bize Şekil 3.5.'deki gibi kameralara, mikrofonlara ve eğitim motoruna doğrudan erişme imkanı verir.



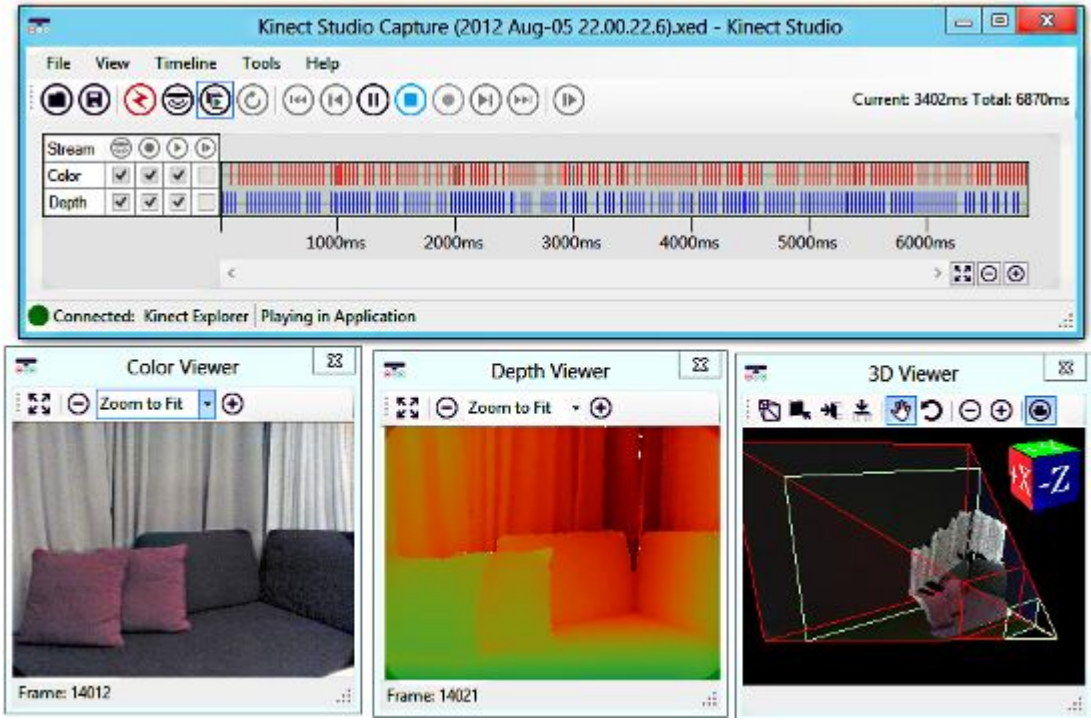
Şekil 3.5 Donanıma Erişme

Projeye kütüphaneyi dahil ederek donanıma eriştikten sonra sensörleri kullanmak için bir sensör değişkeni tanımlanır. C#'ta normal bir değişken tanımlamak gibidir.

Bu kütüphanede bulunan fonskiyonlar renkli görüntüler elde etmede kullanılabilir. Görüntüler bir akış şeklinde bilgisayara aktarılır. Bu akış görüntü çerçevelerinden oluşur. Bu akışa **ImageStream** denilmektedir. Kinect sensörlerde iki çeşit akış bulunmaktadır. Bunlardan birisi renkli görüntüler aktarırken diğeri derinlik bilgisi içeren görüntüler aktarır. Renkli görüntülerde pikseller kırmızı, yeşil ve mavi olmak üzere 3 renkten oluşur. 3 çeşit görüntü formatı Kinect tarafından desteklenir. Bunlar YUV, RGB ve Bayer'dir. Bu görüntüler yüz algılama ve bir çok uygulama da kullanılabilir. YUV formatı video kaydetmede kullanmak için elverişlidir. RGB formatı ise bitmap görüntüler oluşturmak için uygun bir formattır. Bayer filtresi uygulanmış görüntülerde pikseller yeşil tonlar ağırlıkta görüntülenir. İnsan gözü yeşile daha duyarlı olduğu için bu formattaki görüntülerde detaylar daha rahat görülür. RGB renk formatında her bir piksel 4 bilgi içerir. Red, green ve blue değerlerinin yanında birde alfa bilgisinin bulunduğu gözden kaçırılmamalıdır. Aksi takdirde görüntü işlerken döngülerde hatalar olabilir. 3 değer üzerinden işleme yapılırsa pikseller üzerinde ki işlemlerde kaymalar olur. Pikselde her bir renk değeri 0-255 arasında bir değer alabilir. 3 renkten alınan ağırlıktaki değerlerle piksel oluşur.

SDK sayesinde kızılötesi görüntüler elde edilebilir. Bunun için **Infrared Frame Reader** metodundan yararlanılır. Gelen kızıl ötesi görüntüler **infraredFrameData**

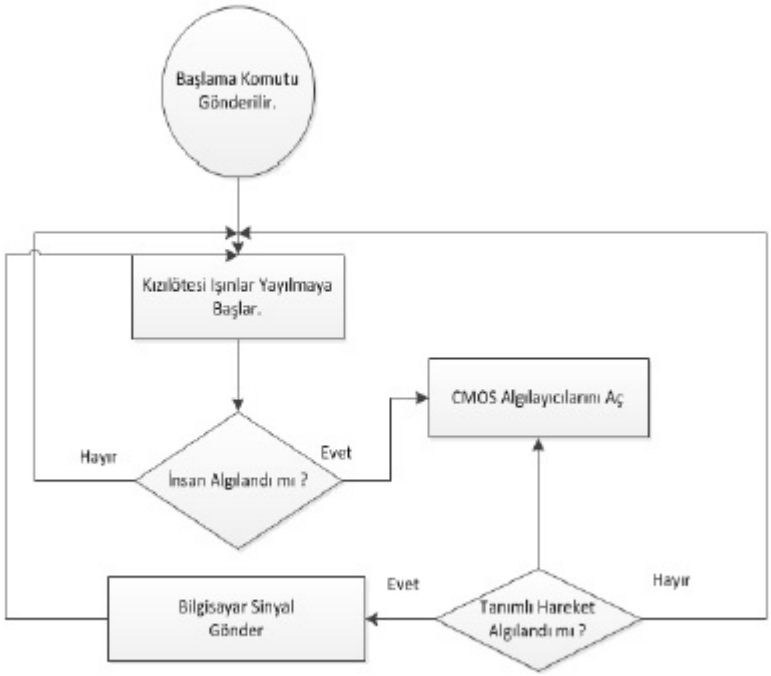
denilen yerde tutulur. Daha sonra bizim görebileceğimiz şekilde dönüştürülen pikseller **infraredPixels** değişkeni içerisinde tutulur. Bu pikselleri bir bitmap'e dönüştürülerek görülebilir. Bu işlemlerin tekrar tekrar yapılabilmesi için bir olay dinleyicisine ihtiyaç vardır. Bunu **FrameArrived** metodu ile sağlayabiliriz. SDK kurulurken yüklenen kinect studiodan yararlanarak Şekil 3.6.'daki gibi görüntüler elde elde edilebilir..



Şekil 3.6 Kinect Studio

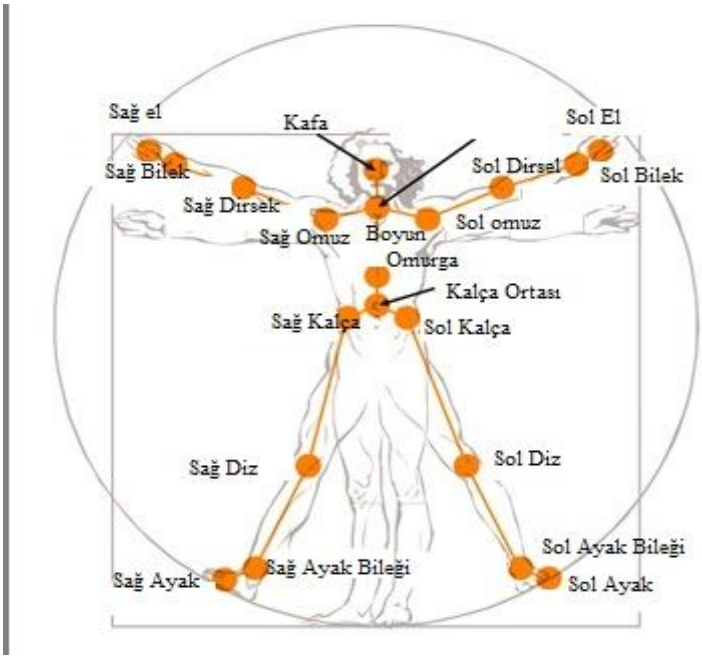
Kinect 16 bit derinlik ölçerken 16 bit veri gönderir. 13 bit tanımlanmış kişilerin milimetre cinsinden uzaklık bilgilerini tutarken, ilk 3 bit ise kişi indeksini saklar. Her pikselin uzaklığı ayrı ayrı hesaplanır.

Kinect yazılımı sayesinde insan iskeleti izlenebilmekte ve 20 tane eklemin hareketleri takip edilebilmekte ve konumları okunabilmektedir. İnsan hareketleri Şekil 3.7.'deki algoritmaya göre algılanmaktadır (Süzen ve Taşdelen, 2013). Aynı anda 6 adet insan iskeleti algınalabilmektedir. Fakat sadece iki kişinin iskelete konum detaylarını bilgi olarak döndürebilmektedir. Bu durum çok kişili uygulamalarda ve oyunlarda kolaylık sağlamaktadır.



Şekil 3.7 Kinect İnsan Hareketleri Algılama Algoritması

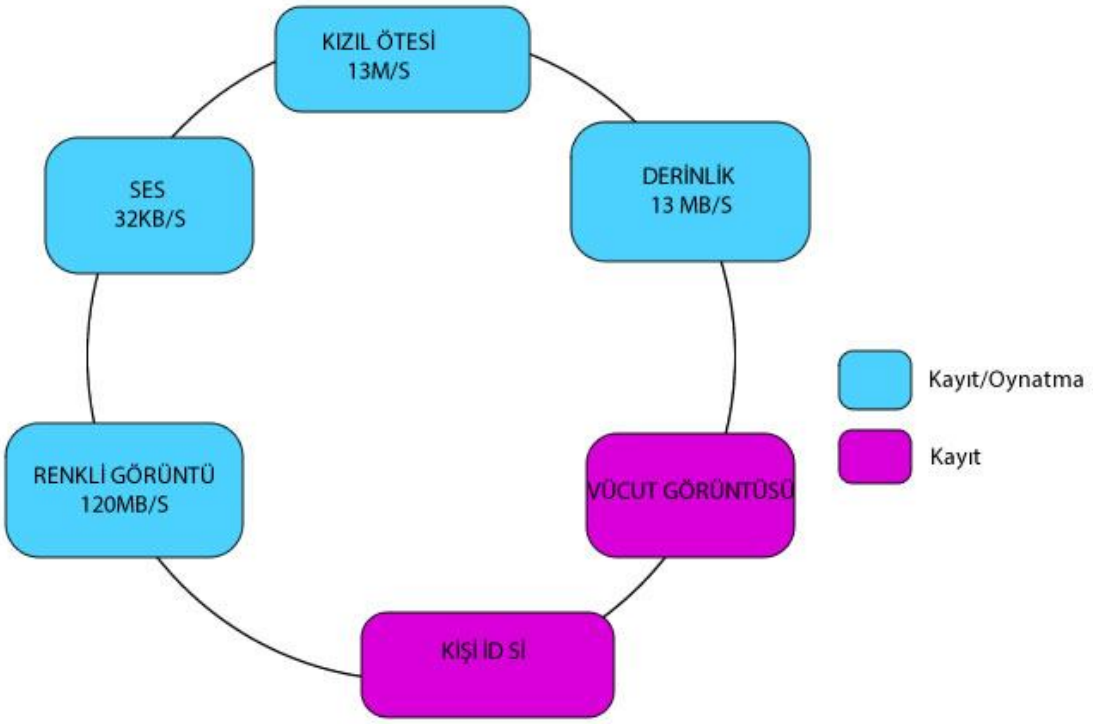
Bu sayede 20 eklem noktası algılanabilmektedir (bin Mohd Sidik ve ark., 2011). Bu bölgeler Şekil 3.8.'de gösterilmiştir. İskelet algılama modunu varsayılan yaparsak 20 eklem bölgesi algılanırken oturan insanlar için oturuş modunu seçildiğinde 10 eklem bölgesi tanımlanabilmektedir. Oturuş modunda merkez iskelet noktası iki omuzun ortası kabul edilir. Bu mod genellikle bir müzik aleti çalma gibi el işaretleriyle yönetilecek uygulamarda kullanılabilir.



Şekil 3.8 Kinect Tanımlı Eklem Bölgeleri

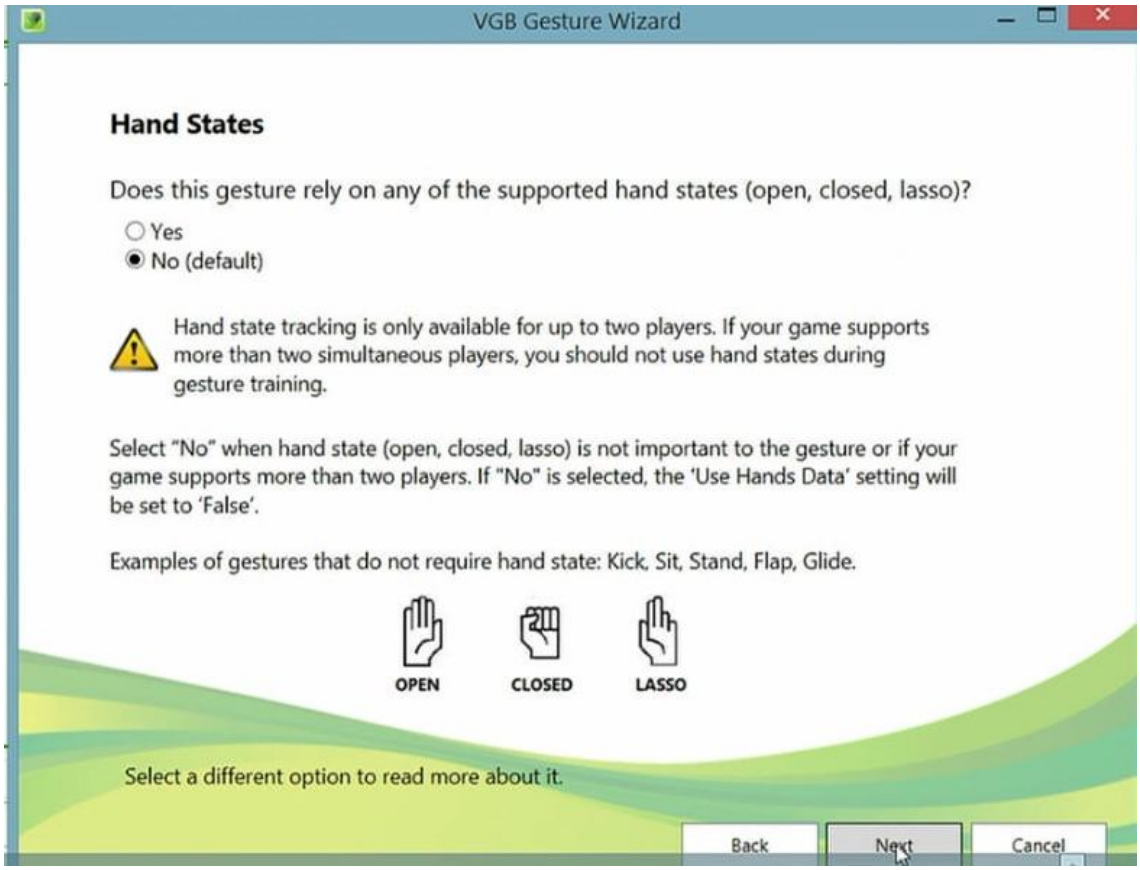
Daha sonra bu eklem bölgelerinin konumlarına göre tanımlı hareketler sensör tarafından algılanabilir veya görüntü işleme teknikleriyle yeni özel hareketler tanımlanıp sensörün işlevleri arttırılabilir. SDK kütüphanesinde bulunan Visual Gesture Builder aracı ile bu işlemi yapabiliriz. Kinect Studio sayesinde Şekil-3.6 daki gibi önce görüntüler kaydedilir. Daha sonra bu görüntüler Visual Gesture Builderda proje olarak açılır. Görüntüler üzerinde istenilen davranışlar işaretlenir. Diğer kısımlar ya negatif olarak işaretlenir ya da çıkartılır. Daha sonra makine öğrenmesi algoritmaları ile eğitim aşaması başlar. Eğitim sonucunda öğrenme gerçekleşir. Her harekete bir isim verilir ve eğitilmiş veri tabanı kaydedilir. Bu veri tabanı Visual Studio C# uygulamasında kullanılmak üzere kaydedilir.

İki çeşit hareket tanımlama yöntemi vardır. Bunlardan birisi sürekli hareketler diğeri ise ayrık hareketlerdir. Mesela elimizle bir daire çizmek sürekli bir hareketi temsil edebilir. Bir kolumuzla bir yönü işaret etmek ise ayrık bir hareket olarak tanımlanabilir. Ayrık harekette hareketimizin son hali önemlidir. İstenilen hareket yakalandığında olay tetiklenir. Video kayıt edilirken boyutuna ve hareketin net algılanabilir ortamda olmasına dikkat edilmelidir. Uygulama nasıl bir ortamda kullanılacaksa video kaydetme ve eğitim aşaması buna yakın bir ortamda yapılırsa daha yüksek oranda doğruluk payı olan uygulamalar geliştirilebilir. Kayıt ederken derinlik görüntüler, renkli görüntüler, kızıl ötesi görüntüler, iskelet görüntüsü, vücut görüntüsü gibi farklı tercihlerde bulunulabilinir. Bu tercihler kaydettiğimiz klibin boyutunu etkileyecektir. Hangi tercih yapılırsa yapılsın hareket algılamada eklem bölgelerinin konumlarından faydalanılacaktır. Visual gesture builder renkleri kullanmadığı için renkli görüntü ayarı kapatılırsa video klip boyutu küçültülmüş olur. Gerekli ayarlamalar yapıldıktan sonra kayıt yapılır ve dosya kaydedilir. Dosya xef uzantısı ile kaydolur. Kinect tarafından kaydedilebilen veriler Şekil 3.9.'da gösterilmiştir.



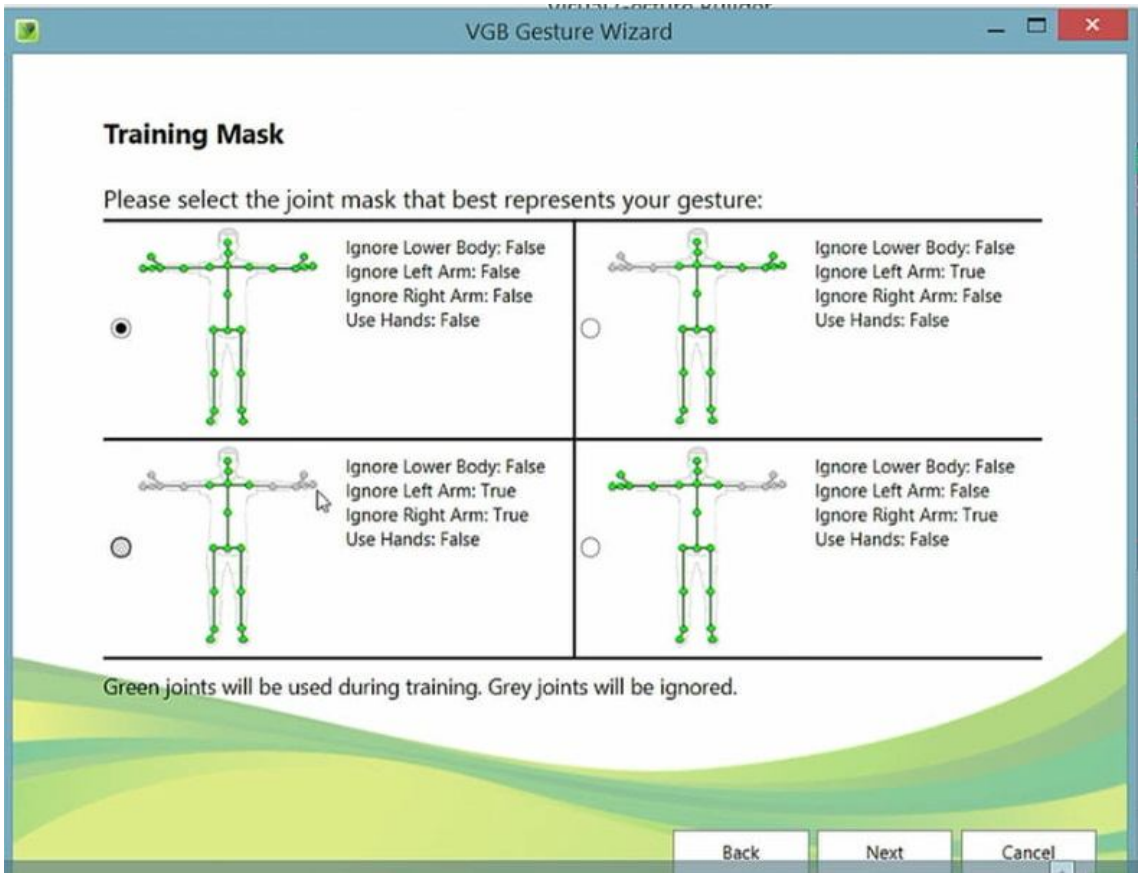
Şekil 3.9 Kinect Studio tarafından kaydedilebilen görüntüler

Kayıt yapıldıktan sonra yine SDK Browserda Visual Gesture Builder aracını açılır. Yeni bir solution dosyası oluşturulur ve veri tabanı dosyamız oluşur. Veri tabanı üzerinde sağ tuşa tıklayıp sihirbaz ile hareket oluştur tıklanır. Hareketlerin ismi verilir. Bu sihirbazda hangi hareketlerin tanımlanacağı ve eğitim aşamasının nasıl olacağı belirlenir. Hareketin hangi eklemlerden faydalanacağı da bu aşamada belirlenir. Mesela oturarak tanımlanan hareketlerde belden aşağısı eklemlerin çalışmamızda faydası yoktur. El durumu kullanılacak mı yoksa kullanılmayacak mı burada belirlenir. Kinect elin durumunu Şekil 3.10.'daki gibi açık, kapalı ve yarı açık olarak 3'e ayırmıştır.



Şekil 3.10 El durumları

El durumunu seçtikten sonra sihirbazda Şekil 3.11.'deki gibi hangi eğitim maskesi uygulanacağı sorulmaktadır. Yapay sinir ağlarında veriler ne kadar ayırt edici olursa eğitim aşamaması o kadar yüksek başarımla gerçekleşir. Bu yüzden bu aşamalar önemlidir. İhtiyaç duyulmayan veriler yapay sinir ağlarını hem zorlayabilir hem de başarı oranını düşürebilir. Bu yüzden örneğin sadece sağ el ile kullanılacak olan bir uygulamada sol eli göz ardı etmek başarı oranını arttırabilir ve yapay sinir ağını zorlamayabilir. Ayrıca bu sayede eğitim süresi de azalacaktır.

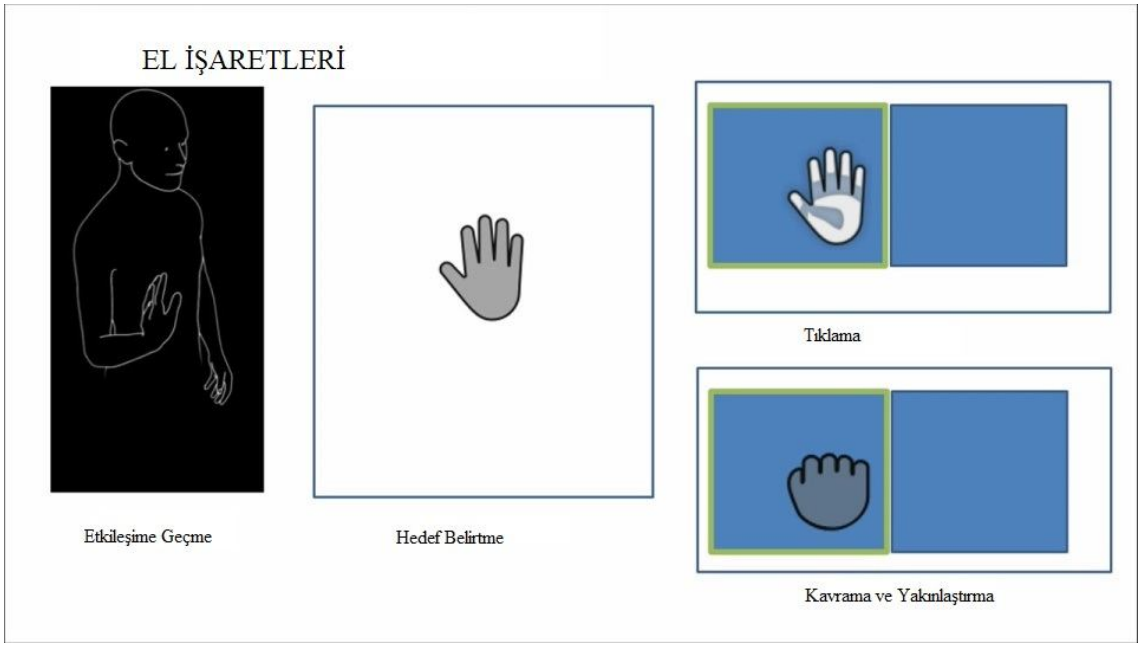


Şekil 3.11 Eğitim maskesi uygulama

Sihirbaz bitince hareket ismin üzerine sağ tuşa tıklanıp klip ekle denir. Daha öncesinde kaydedilen klip getirilir. Bu klip üzerinde kare kare işaretleme yapılır. İstenilen pozisyon oluştuğunda ki kare enter tuşu ile true olarak işaretlenir. Diğer durumlar ya false olarak işaretlenir ya da boş olarak bırakılır. Bu şekilde eğitime hazır hale getirilir. Eğitim işlemini başlatıp daha sonra deneme yaparak doğruluk oranını görülebilir. Sonuçlardan memnun kalınırsa veri tabanı Visual Studio projeye dahil edip bu hareketler komut haline getirilebilir.

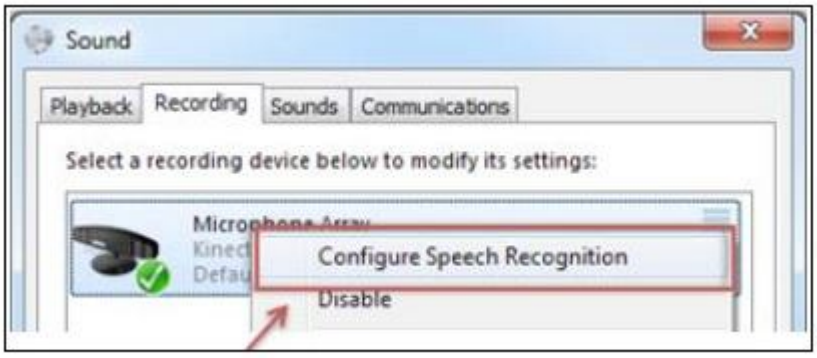
SDK kütüphanesinde bulunan **handpointer** sınıfı sayesinde insan eli bilgisayar faresinin yerini alabilmekte ve uygulamalar el ile yönetilebilmektedir (Kang ve ark., 2011). Bu sınıf sayesinde sensörle etkileşime geçilebilir. Elin aktif hale gelmesi için (Engagement) omuz yukarısına kaldırılarak bir süre beklenilmesi gerekmektedir. Bu hareket dışına çıkıldığı zaman cihaz tekrar serbest kalır. Bu durum sensör karşısında yapılan her hareketin bilgisayarı yönetmek olmadığı durumlarda onu serbest bırakmak için avantaj sağlar. Bu hareketle sisteme seni kullanacağım mesajı verilmiş olur. Elimizi tekrar indirdiğimizde ise şu anlık işimiz bitti mesajı vermekteyiz. Dolayısıyla parazit hareketlerin sistemi karıştırması engellenmiş olur. Aynı işlemi hem sağ hem de sol

elimizle yapabiliriz. Bu şekilde daha ergonomik bir kullanım sağlanmış olur. Yorulunca el değiştirebiliriz. Şekil 3.12.'de gösterildiği gibi handpointer'a ait grip(kavrama) metotu sayesinde kaydırma, yakınlaştırma ve uzaklaştırma işlevleri gerçekleştirilirken tıklama işlevide el ileriye hareket ettirilerek yapılır.



Şekil 3.12 El işaretçisi

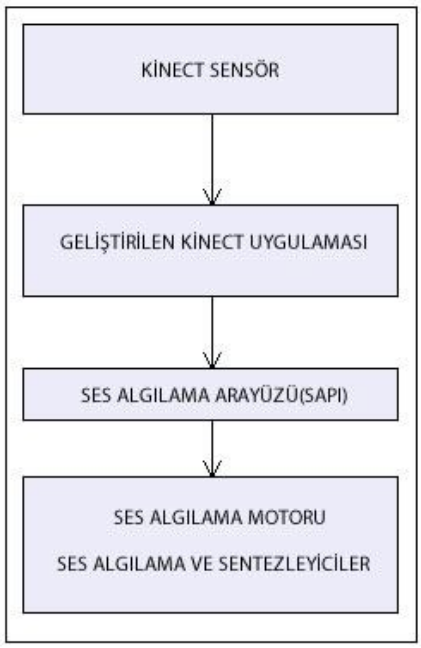
SDK kütüphanesi sayesinde insan sesi de komut olarak kullanılabilmektedir. Kinect senrörün mikrofon dizisinin kalitesinden faydalanarak ve Microsoft Speech Recognition kullanarak sesleri oluşturduğumuz komutlarla uygulamamızı yönetebiliriz. Bu özelliği kullanabilmek için Speech Recognition'ı etkin hale getirmek lazım ve kinect mikrofon dizisini Şekil 3.13.'deki gibi varsayılan mikrofon olarak ayarlamak gerekmektedir.



Şekil 3.13 Mikrofon dizisini varsayılan olarak ayarlamak

Uygulamada Speech Recognition Engine bir ara birim olarak seslere karşılık gelen sonuçları döndürür. Uygulama Şekil 3.14.'deki mantığa göre çalışmaktadır. Kelimelerin

birbirine benzeme durumunda da algoritma cümle içinde kelimenin gelişinden tahmin etme özelliğine sahiptir. Bu durum doğruluk oranını artırmaktadır. Normal bilgisayarlarda parazitlerin sesli komutları etkilediği görülürken Kinect Sensörlerin mikrofon dizisi sayesinde uzaktan bile sesler daha net algılanabilmektedir.



Şekil 3.14 Kinect sesli komut algılama

Sesleri komut olarak kullanabilmek için öncelikle tanıttığımız mikrofon dizisini aktif hale getirmemiz gerekir. Daha sonra ses veri akışını sağlarız. Ses Tanıma Arayüzü'nü projemizde kullanabilmek için projemizin referanslarına System.Speech kütüphanesini eklememiz gerekmektedir. Ekledikten sonra gramer oluştururuz. Sonra konuşma algılayıcısını başlatırız. Daha sonra gelen ses verilerini ses algılayıcısına göndeririz. Daha sonra bu komutların doğru olduğu anda işlem yaptırabilmek için bir olay dinleyici metot oluştururuz. Bu metot sayesinde söylenen komutlardan herhangi birisi yakalandığında istenilen işlemleri gerçekleştiririz.

Tüm bu özelliklerini incelediğimizde Kinect sensörlerin aşağıdaki işlemleri gerçekleştirebildiği görülmüştür:

- Eş zamanlı video kaydı yapma.
- İnsan vücudunu yakalama ve iskelet izleme sistemiyle hareketleri tanıyabilme.
- Nesnelerin sensöre olan uzaklığını hesaplama.
- Bir derinlik haritası çıkarabilme.

- Nesneleri 3 boyutlu görebilme.
- Karanlıkta görme.
- Ses tanıma ve yönünü tayin etme.

3.2. Fatih Projesi Etkileşimli Tahta

Bilgi iletişim teknolojilerinde ki hızlı gelişmeler eğitim alanında görülmeye başlamış ve Fatih Projesi ile teknoloji sınıflarımıza da girmiştir. Milli Eğitim Bakanlığı'na bağlı okullarda her sınıfa Şekil 3.15.'deki gibi bir etkileşimli tahta kurulmuş, öğrencilere tabletler dağıtılmıştır. Tahta büyük dokunmatik ekrana sahip bir bilgisayar kısmı birde diğer yarısında normal sınıf içi kara tahta olacak şekilde tasarlanmıştır. Montajları sağlam bir şekilde yapılıp uzun yıllar kullanılabilecek bir şekilde tasarlanmıştır. Projede getirilen sadece tahtalar olmayıp, oluşturulan ağ yapısı da sınıf içinde teknoloji kulanımını arttırmıştır. Çekilen fiberoptik hatlarla internet hızı arttırılmıştır. Okulllara kurulan yerel ağ sayesinde sağlam bir altyapı oluşmuş bütün cihazlar birbirine bağlı hale getirilmiştir. Proje sayesinde sınıfta kullanılan projeksiyon cihazlar ve bilgisayarlara sınıflarda ihtiyaç kalmamış. Hem çirkin görüntülerin önüne geçilmiş hem de sınıf içerisinde bulunan güvenlik riski oluşturan kablolar ve prizler ortadan kaldırılmıştır. Okulların elektrik tesisatları gözden geçirilip daha güvenli ve verimli hale getirilmiş ve elektronik cihazların elektrik kaynaklı arızalanmalarının önüne geçilmiştir. Öğretmenler bu donanım sayesinde yanında bilgisayar, taşınabilir sabit disk gibi donanımlar taşımak zorunda kalmamışlardır. Öğretmenler materyallerini öğretim yılı öncesinde tahtalara yükleyebilip yıl içerisinde döküman taşıma zorluğundan kurtulmuştur. Bu durum derse hazırlık süresini azaltmıştır. Ayrıca tahtalarda internet olması ve EBA(Eğitim Bilişim Ağı) gibi sistemlere bağlanabilmesi geniş kaynak erşimini getirmiştir. Donanım olarak proje iyi yerlere gelmiş artık bu donanımların daha etkili bir şekilde kullanılabilmesi için daha fazla materyale ve yazlımlara ihtiyaç vardır.



Şekil 3.15 Fatih Projesi Etkileşimli Tahta

Her alanda olduğu gibi sınıf ortamında da teknolojiden faydalanma ihtiyacı artmıştır. Zamanı etkili kullanma ve eğitimi görselleştirme açısından proje önemli bir yer tutmaktadır. Yapılan araştırmalar akıllı tahtaların eğitim-öğretim faaliyetlerine katkıda bulunduğunu ortaya çıkarmaktadır (Keser ve Çetinkaya, 2013). Etkileşimli tahta ile yapılan öğrenci merkezli görsel etkinliklerin sınıf yönetimine olumlu katkıları olduğu saptanmıştır (Albayrak, 2014). Fatih Projesi'nin temel amacı eğitimde fırsat eşitliğini sağlamak ve okullarda ki teknolojiyi geliştirmektir (Pamuk ve ark., 2013). Bizde çalışmamızda tahtaların kullanım alanlarını genişletmeyi amaçladık. Fatih Projesi Etkileşimli Tahtalar donanımsal olarak incelendiğinde Çizelge 3.2.'deki donanımlara sahip oldukları görünmektedir.

Çizelge 3.2 Fatih Projesi Etkileşimli Tahtaların Teknik Özellikleri

Teknik Özellikler	
İşlemci	Intel® Core™ i3 3120M 2.5GHZ 3MB Cache DDR3/L/-RS 1333/1600 AMD Quad-Core A10 -Series 5750M, 4MB Cache, 2.5GHz, DDR3-1866MHz Max.
Yonga Seti	Intel® HM76 Express Yonga Seti AMD A10 yonga seti
Bellek/Grafik Denetleyicisi	SATA III (6GB/s) port (1 Adet) DDR3 RAM slotu x 2 (16GB) Uzaktan Yönetim desteği Intel® HD Graphics 4000 (512MB) AMD Radeon HD 8650G (512MB) DirectX 10, OpenGL ES 1.x ve 2.X desteği Donanım Temelli Ölçekleme (1080p dahil) Çalışma frekansı ; 600Mhz
Bellek	H.263, H.264, MPEG-4, HEVC uygulamaları çalıştırabilme. 1080p HD ve 30fps (frame per second) desteği DDR3 So-Dimm Single Channel 8 GB- DDR3 / 1333Mhz bellek modülü
Veri Depolama	256GB-SSD Sabit disk – 6 Gbps Okuma Hızı 220MB/s , Yazma hızı 110MB/s SSD ömrü 800 bin saat I/O miktarı 50.000 IOPS

Çizelge 3.2 incelendiğinde Fatih Projesi Etkileşimli Tahtalarının donanımsal olarak yeterli bir teknolojiye sahip oldukları söylenebilir. Kinect sensörünün çalışacağı bilgisayarda olması gereken minimum özellikler Çizelge 3.3.'de gösterilmiştir.

Çizelge 3.3 Minimum Sistem Gereksinimleri

Minimum Sistem Gereksinimleri
64-bit (x64) İşlemci
Çift veya daha fazla çekirdekli işlemci
USB 3.0 bağlantı noktası
4 GB RAM bellek
Windows 8, 8.1 veya 10 işletim sistemi

Görüldüğü gibi etkileşimli tahtalar bu ihtiyaçları karşılamaktadır. Tekirdağ Şarköy Mesleki ve Teknik Eğitim Merkezinde kurulmuş olan bir etkileşimli tahtada kinect sensör denenmiş ve başarılı bir şekilde çalışmıştır. Bunun üzerine yapılan çalışmamızın mümkün olduğu görülmüştür.

3.3. Yazılımın Geliştirilmesi

Projemizde Windows 8.1 işletim sistemi üzerinde çalışılmıştır. Sistem gereksinimlerini inceledikten ve kurulum tamamladıktan sonra kinect adaptör üzerinde ki ışık sarı yanıyorsa artık çalışılmaya hazır demektir. Işık turuncu yanıyorsa bir problem var demektir. Kinect Studio sayesinde sorunlar görüntülenebilir. Kinect sensör kendisini kullananan bir program yoksa kapalı durmaktadır. Programdan talep gelince açılmaktadır. Uygulamamızı Visual Studio platformunda C# dilinde geliştirdik. SDK sayesinde gerekli kütüphaneleri uygulamamıza dahil edip kütüphanelerin fonksiyonlarından faydalanarak uygulamalar geliştirebilmekteyiz. Uygulamamızı WPF (Windows Presentation Foundation) olarak geliştirdik. WPF Windows formlara göre daha yeni bir sunum çerçevesidir (Sells ve Griffiths, 2005). WPF XAML sayfaları sayesinde ara yüzlerini geliştirmemizi kolaylaştırır. Kinect SDK içerisinde WPF Control Basics adlı örnek uygulamayı incelediğimizde sınıf içerisinde kullanacağımız uygulamanın bu formatta geliştirilebileceğini düşündük. Sınıf içerisinde kullanılan materyalleri incelediğimizde en çok resimler, grafikler, küreler, modeller, PDF kitaplar, sunular ve görsel eğitim disklerinin kullanıldığını görmekteyiz (Demiralp, 2007). Bizde

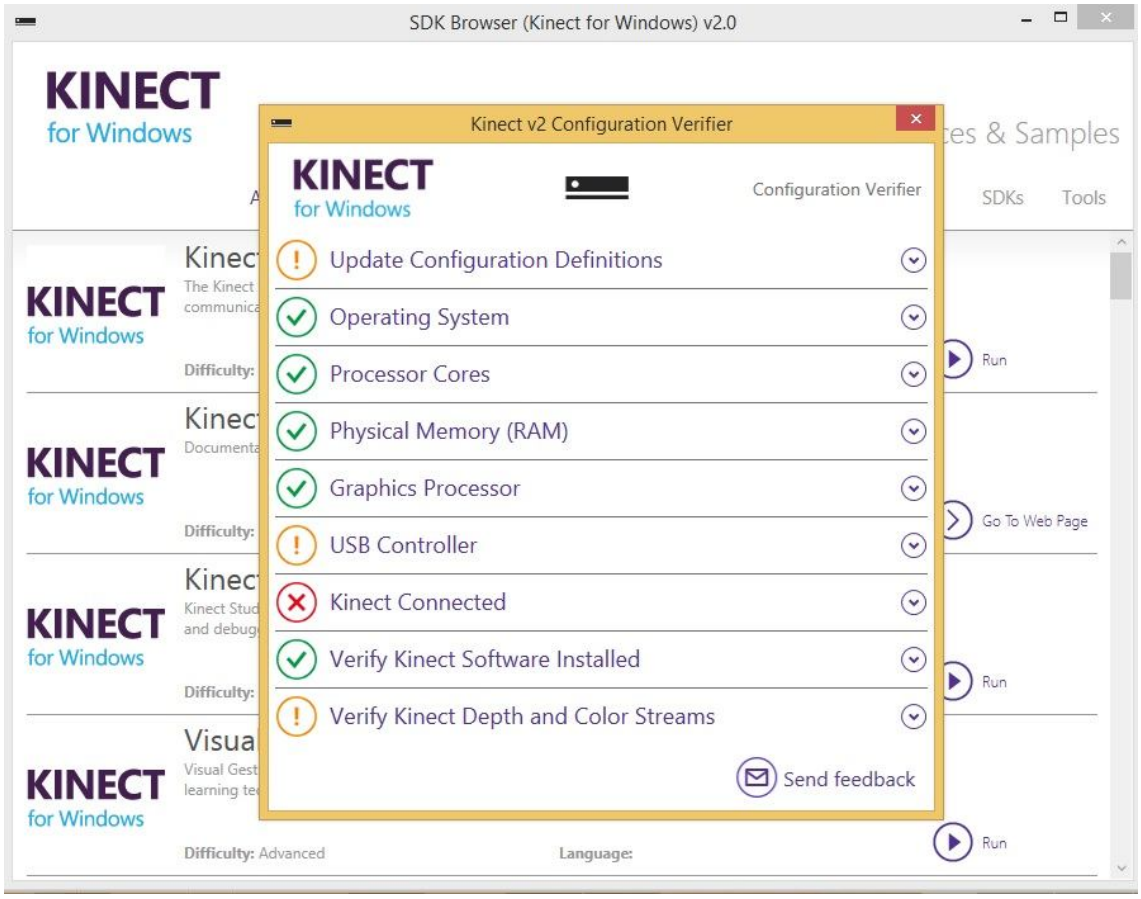
bu duruma dayanarak uygulamamızı 4 temel işlev üzerine geliştirmeyi amaçladık. Ana giriş ekranı Şekil 3.16.'da gösterilen uygulama şu 4 temel işlevden oluşmuştur:

1. Resim Görüntüleyici.
2. PDF Kitap Görüntüleyici.
3. Video Görüntüleyici.
4. Sunu Gösterici.



Şekil 3.16 Uygulamanın Ana Ekranı

Kinect Sensörün bilgisayarda çalışabilmesi için Kinect SDK 2.0 kurulması gerekmektedir. SDK kurulurken sensörün sürücüleri de bilgisayara kurulur. SDK kurulunca programlarda SDK browser arayüzüne ulaşabiliriz. Bu browser sayesinde cihazımızın doğru kurulup kurulmadığını Şekil 3.17.'deki gibi kontrol edebilir ve bazı örnek uygulamalar çalıştırabiliriz.

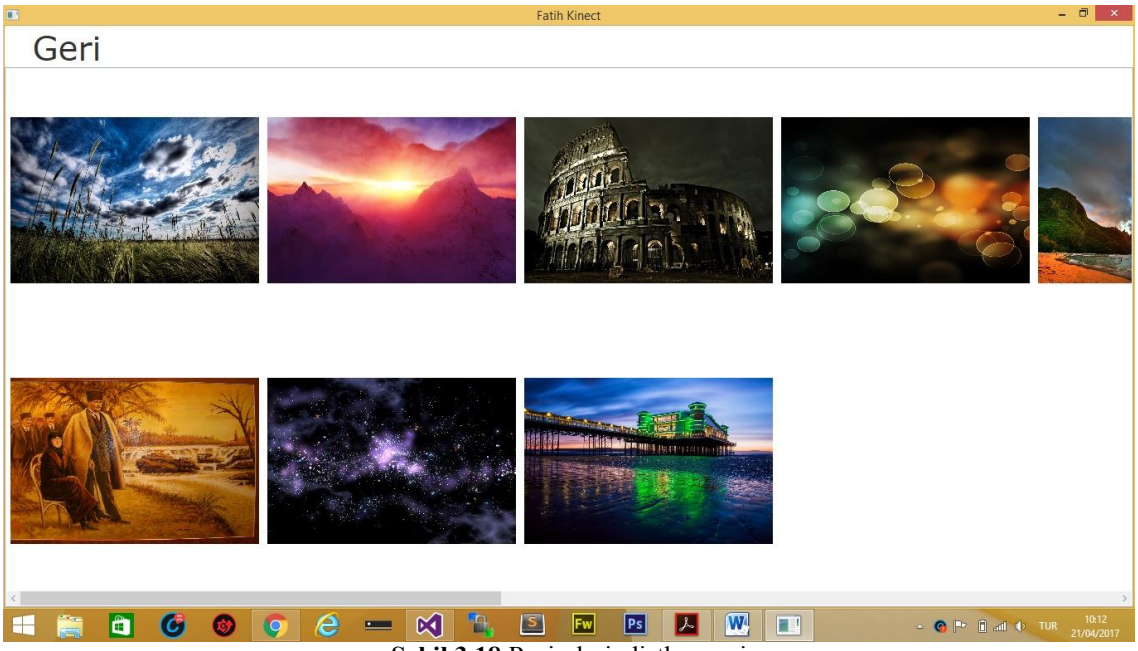


Şekil 3.17 SDK Browser

Kinect SDK fonksiyonları sayesinde insan eli otomatik olarak tanımlanabilmektedir (Pedersoli ve ark., 2012). Geliştirilen uygulamalarda XAML dosyalarında ki kullanıcı kontrolleri KinectRegion denilen bölgeler içerisine konulduğunda kinect sensörün HandPointer sınıfı otomatik olarak çağırılmaktadır. Bu sayede HandPointer sınıfının metotları bu alanlarda kullanılabilmektedir.

3.3.1. Resim görüntüleyici

İlk olarak WPF de Windows Resim Görüntüleyiciye benzer bir uygulama geliştirilip kinect sayesinde el ile kontrolü sağlanmıştır. Resimler sınıf içerisinde en çok kullanılan materyallerden bir tanesidir. Kullanıcılar resim galerileri arasında gezebilecek ve resimleri uzaklaştırıp yakınlaştırabilecektir. Bu uygulama için Şekil 3.18.'deki gibi bir XAML sayfası oluşturulmuştur. Önce bilgisayarda kayıtlı resimler bir liste şeklinde görüntülenmiştir. Kinect kütüphanesi projemize dahil edilmiştir. Bu listedeki resimleri Kinect sensör sayesinde yönetebilmek için XAML arayüzü KinectRegion dediğimiz Kinect tarafından yönetilebilir alan olarak tanımlanmıştır.



Şekil 3.18 Resimlerin listlenmesi

3.3.2. PDF kitap görüntüleyici

İkinci aşamada MEB'in yayınlamış olduğu PDF kitapları gösteren bir işlev geliştirilmiştir. Yine bu işlevde kitap içerisinde gezilebilecek ve yakınlaştırma uzaklaştırma yapılabilmiştir. Mesleki eğitimde modül denilen bu pdf kitaplar bütün kazanımları içermektedir. Ders içerisinde çok kullanılan bir materyaldir. Yine resim görüntüleyicide ki gibi Kinect etkinleştirilmiş ve yakınlaştırma uzaklaştırma özellikleri aktif hale getirilmiştir.

3.3.3. Video oynatıcı

Üçüncü aşamada bir video oynatıcı geliştirilmiştir. Bu sayede görsel eğitimler izlenirken öğretmen sınıf içerisinde dolaşabilecek istediği anda hiçbir kontrole gerek duymadan eliyle videoyu durdurup oynatabilecektir. Günümüzde görsel eğitim materyalleri genelde video olarak hazırlanmaktadır. Bu görseller öğretmenlere büyük kolaylıklar sağlamaktadır. Video oynatırken hem sınıf içerisinde dolaşabilmek hem de sınıf içerisinde her konumdan videoyu kontrol etmek amaçlı bu işlev geliştirilmiştir. Video oynatma sayfamızda çoklu kontrol sağlanmıştır. Video oynatıcı hem el ile kontrol edilirken ses ile kontrolde eklenmiştir. Videolar önce listelenmiş istenilen video tıklandığında sayfa açılmıştır. Sayfaya el ile kontrol edilebilen Şekil 3.19.'daki gibi bir arayüz hazırlanmıştır.



Şekil 3.19 Video oynatıcı

Uygulamamızı ses kontrolü eklerken Windows'un konuşma tanıma motorunu kullandık. Bunu kullanabilmek için Türkçe desteği olmadığından işletim sistemi dilimizi İngilizceye çevirdik. Daha sonra bilgisayarımızda konuşma tanımayı aktif hale getirdik. Aktifleştirdikten sonra konuşma tanımada kullanmak üzere donanım olarak Kinect mikrofon dizisini aktif cihaz olarak seçtik. Bir sınama yapıp cihazın çalıştığını gördük. Daha sonra projemize System.Speech kütüphanesini ekledik. Bu kütüphaneyi ekledikten sonra Sensörden gelen sesleri konuşma tanıma motoruna gönderebildik. Bunun için bir gramer oluşturduk. Gramerimize bize gerekli olan komutları ekledik. Video oynatma da oynatıp durdurmak, ileri ve geri sarma yapmak, sesi açıp kapamak için kullanılmak üzere Şekil 3.20.'deki gibi İngilizce komut dizisini girdik.

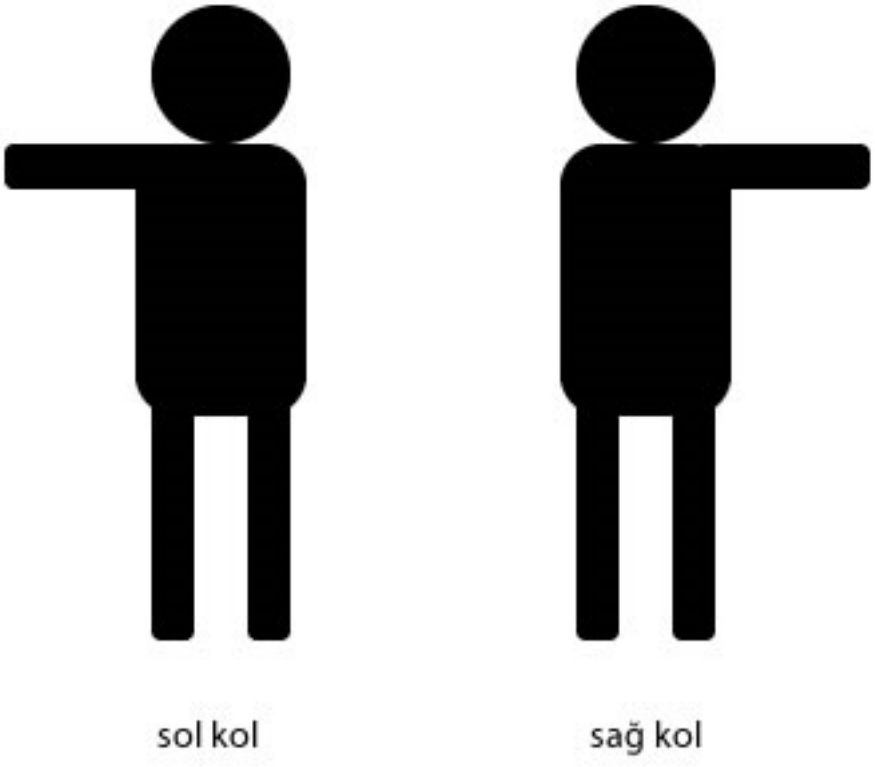


Şekil 3.20 Sesli komutlar

Komut dizisi oluşturulduktan sonra bu komutlardan herhangi birinin söylendiğini takip etmek için bir olay dinleyicisi ekledik olay dinleyicisi içinde gelen komuta göre olayları tetikledik.

Video oynatıcımıza birde handpointer kontrolü ekledik. Şekil 3.20.‘ deki gibi butonları bu el kontrolü ile kontrol edebiliyoruz. Sınıf içerisinde gürültülü ortamlarda ses kontrolünü sağlayamadığımız anlarda da yine istersek ile videomuzu durdurup oynatabiliyoruz.

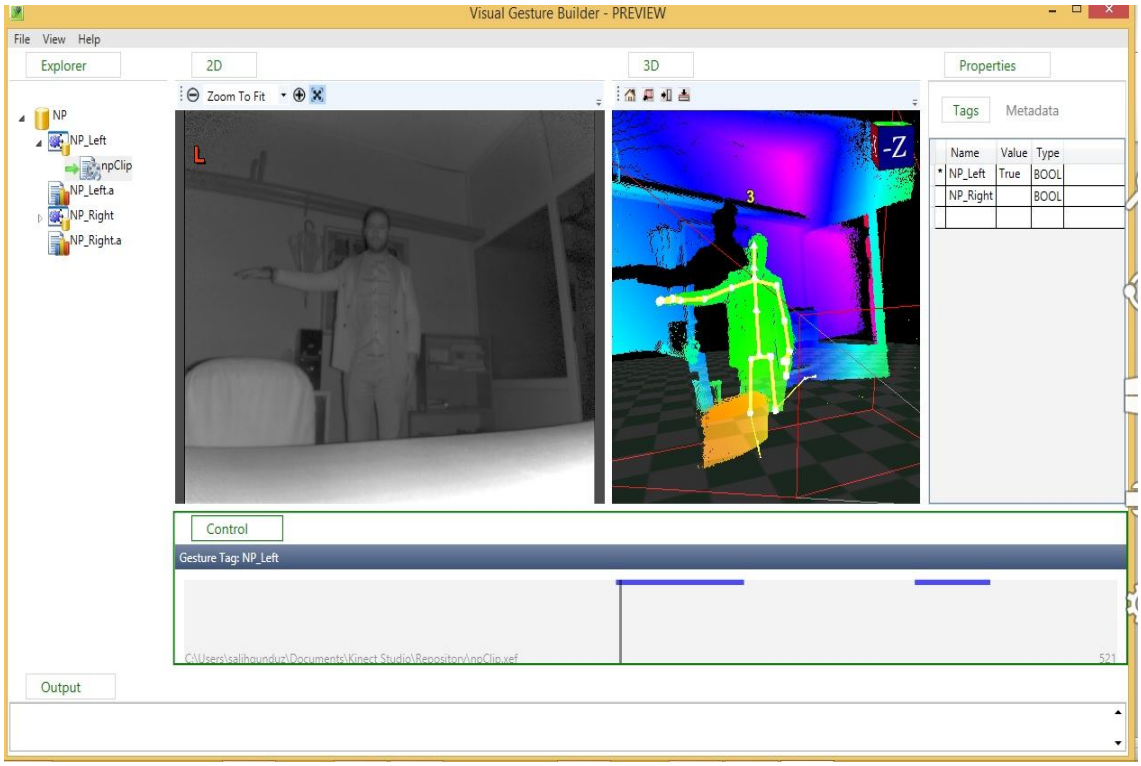
Kinect’in bir diğer avantajı ise SDK içerisinde bulunan Gesture Builder aracıdır. Bu sayede uygulamamıza yeni özgün hareketler tanımlayıp aynı ses kontrolünde olduğu gibi bu hareketleri dinleyip yakalandığı anda olayları tetikleyebiliriz. Bunun için Kinect SDK Browserdan yararlandık. Önce tanımlayacağımız hareketlere karar verdik. Şekil 3.21.‘deki gibi sağ ve sol kolun vücudumuzla 90 derecelik açı yaptığı iki hareket tanımlamaya karar verdik.



Şekil 3.21 Tanımlanan komutlar

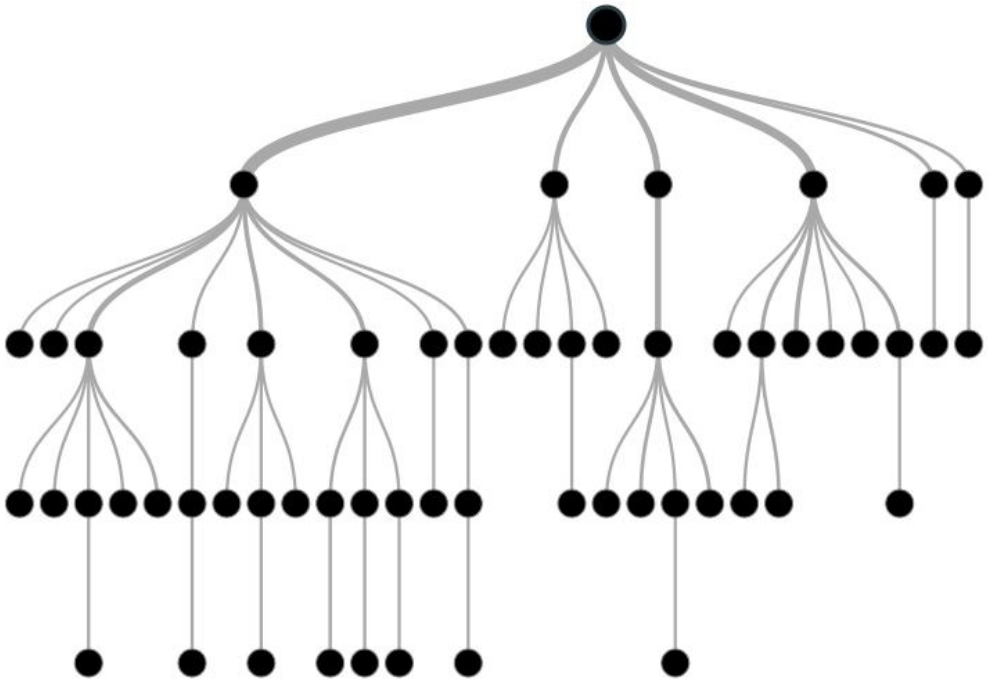
Öncelikle eğitim aşamasına geçmeden, içinde yapacağımız hareketlerinde olduğu bir video klip çektik. Videomuzu Kinect Studio yardımı ile kaydettik. Kayıt yaparken tanımlanan hareketleri belirgin bir şekilde 2 şer defa yaptık. Bu işlemi yaptıktan sonra Visual Gesture Builder’ı açtık ve sihirbazı çalıştırdık. Sihirbazda projemize isim verdik

ve proje içerisinde 2 hareket tanımlayacağımızı belirttik. Bu hareketleri NP_Left ve NP_Right olarak isimlendirdik. Bu işlemden sonra hareket işaretleme aşamasına geçtik. Her kareyi tek tek istediğimiz hareket oluştuğunda enter tuşu ile Şekil 3.22.'deki gibi true olarak işaretledik. İstemediğimiz kısımlara işaret koymadık. False olarak ta işaretlenebilir fakat bu eğitim süresini uzatabilir.



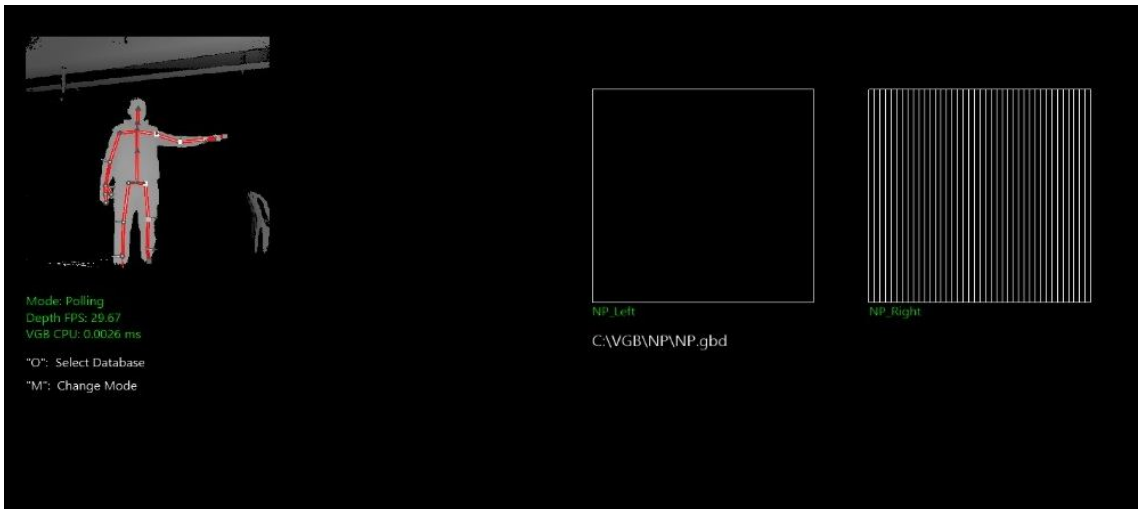
Şekil 3.22 Hareketlerin işaretlenmesi

Visual Gesture Builder eğitim aşamasında Şekil 3.23.'deki gibi makine öğrenmesi yöntemlerini kullanmaktadır. Bu yöntemlerden biri rastgele karar ormanlarıdır.



Şekil 3.23 Rastgele Karar Ormanı

Eğitim aşaması tamamlandıktan sonra "gbd" uzantılı veri tabanı dosyası projemizde kullanılmak üzere kaydedilir. Veri Tabanını ve doğruluk oranını test etmek için Şekil 3.24.'deki gibi Visual Gesture Builder Viewer kullanabiliriz. Hareketlerimizi denediğimiz zaman %99 luk oran ile hareketlerimizi tanımıştır. Daha sonra projemize bu veri tabanı eklenir. Veri tabanına erişim ve hareketleri kullanma için GestureDedector.cs sınıfı yazılmıştır. Bu sınıf veri tabanına bağlanmayı sağlar ve veri tabanında ki hareketleri okur. Projemizde Kinect sensör etkinleştirilir ve veri akışı başlar. Bu veri akışı içerisinde istenilen hareketler yakalanınca olay dinleyici istediğimiz komutları çalıştırmamıza imkan sağlar. Bu şekilde hareketler komut olarak kullanılmış olur. Sensör ortam ışığından etkilenmediği için karanlık ortamlarda bile bu komutlar çalışabilmektedir.



Şekil 3.24 Hareketin doğruluk oranı

3.3.4. Sunu gösterici

Dördüncü aşamada ise bir sunu gösterici geliştirilmiştir. Uygulama önceden hazırlanmış sunuları kendi göstereceği formata otomatik olarak dönüştürecek ve uzaktan kontrolü sağlanacaktır. Bu sayede öğretmen sunumunu sınıf içerisinde dolaşarak yapabilirken öğrenciler ve engelli öğrencilerde hazırladıkları sunumlarını yerlerinden yapma imkanı bulabileceklerdir. Tüm bu uygulamalar geliştirildikten sonra sınıf içerisinde uygulanıp yeterli sayıda öğrenci üzerinde denenmiştir. Sınıf yönetimi ve öğrencilerin ders katılımları takip edilip öğretmen görüşleri alınmıştır.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Kurulum ve erişim

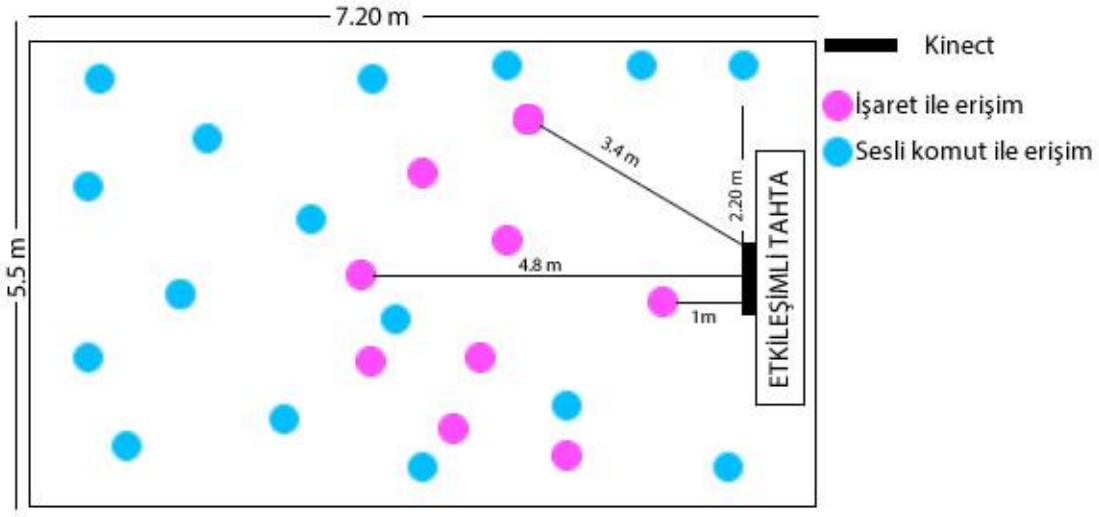
Kinect Sensörün Türkiye satışı olmadığı için sipariş yurt dışından geldiğinden ürünü elde etmek bir hafta kadar sürmüştür. Sensör Amerikan tarzı fişe sahip olduğundan bir dönüştürücüye gerek olduğu görülmüştür. Kurulum ve ilk denemelerden sonra kamera ve ses kayıt kalitesinin oldukça yüksek olduğu görülmüştür. İlk olarak yapılan uygulama, tahtanın donanım özelliklerini incelemek ve sınıf içerisinde sensörün kullanılabilirliğini test etmek amacıyla Şarköy Mesleki ve Tekknik Eğitim Merkezi Bilişim Teknolojileri Sınıfında denenmiştir. Windows 8 işletim sistemi yüklü olan etkileşimli tahtaya sensörün sürücüleri yüklenirken herhangi bir sorunla karşılaşılmamıştır. Tahtalarda USB 3.0 desteği olduğu için bağlantıda sorun yaşanmamıştır. Kinect 2 Windows SDK yüklendikten sonra Kinect Configuration Verifier uygulaması açılarak kontrol edilmiş ve tüm bileşenlerin eksiksiz yüklendiği görülmüştür. Sensörün kızılötesi ışık sayesinde aydınlık ve karanlık ortamlarda çalıştığı görülmüştür. Kinect Studio uygulaması ile ekran görüntüsü alınarak kameranın görüş açısının tüm sınıfı kapladığı Şekil 4.1.'deki gibi görülmüştür.



Şekil 4.1 Kameranın sınıfı görüş açısı

Sınıf ortamında denenen sensöre sınıf içerisinde erişilebilen yerler Şekil 4.2.'de gösterilmiştir. Yapılan ölçümler cihazın standart sınıflarda rahatlıkla kullanılabileceğini

göstermiştir. Sensör oturarak ve ayakta kontrol edilebilmiştir. Tüm bu denemelerden sonra geliştirilen uygulama 20 farklı öğretmen tarafından 20 derste denenmiştir.



Şekil 4.2 Sınıf içi kinect erişim noktaları

Ölçümler kinectin sınıf içerisinde rahatlıkla kullanılabileceğini göstermiştir. İşaret ile dik uzaklık olarak 4.8 metre kadar uzaklıktan erişim sağlanabilmiştir. Kenar açılardan erişim 3.4 metreye kadar düşmüştür. Sesli erişim sınıfın her yerinden sağlanabilmiştir. Fakat video oynatma gibi sesli uygulamalar ve sınıf gürültüsü ses algılamayı zorlaştıran etkenler olmuştur. Ses algılama kabiliyetinin normal mikrofonlara göre daha başarılı olduğu görülmüştür.

4.2. Öğretmen görüşleri

Bu çalışmada öğretmen görüşleri alınırken görüşme yöntemi kullanılmıştır. Yapılan uygulama her ne kadar teknoloji ile ilgili olsa da yazılımın sınıf içerisinde kullanılması sosyal bir yapı ortaya çıkarmaktadır. Öğretmen ve öğrenci duyguları, çalışmanın sınıf yapısına etkisi ve sonuçların incelenmesi sosyal bir inceleme gerektirmektedir. Eğitim bilimciler sosyal olayları incelemede önemli derecede görüşme yöntemini kullanmışlardır (Türnüklü, 2000). Bizim dışımızda ki insanların çalışma hakkında görüşlerini almak hem çalışmayı değerlendirmemizi sağlayabilecek hem de daha sonra yapılacak olan çalışmalara da katkıda bulunabilecektir. İnsanların tutum ve davranışlarını nicel olarak ölçmek tam anlamıyla mümkün olmadığı için yarı yapılandırılmış görüşme tekniği kullanılmıştır. Yarı yapılandırılmış görüşme tekniği yapılandırılmış görüşme tekniğine göre daha esnektir. Bu yöntemle, öğretmenler ile

yapılacak olan görüşme için sorular önceden hazırlanmıştır. Yarı yapılandırılmış görüşme tekniğinde protokol önceden hazırlandığı için daha sistematik ve işlenebilir veriler elde edilebilmektedir. Esnekliği sayesinde ihtiyaç halinde görüşmeci alt sorularla görüşmenin akışını belirleyebilmektedir. Çalışmamızda 17 öğretmen üzerinde uygulamak üzere 5 soruluk bir yarı yapılandırılmış görüşme formu hazırlanmıştır. Uygulama bir sınıfa kurulmuş ortam hazırlandıktan sonra öğretmenlerin bu uygulamayı deneyerek ders anlatmaları sağlanmıştır. Ders sonunda öğretmenler farklı bir odaya alınarak görüşme gerçekleştirilmiş görüşme kayıt altına alınmıştır.

4.2.1. Örneklem seçimi

Örneklem olarak Şarköy Mesleki Teknik Eğitim Merkezinin Öğretmenleri seçilmiştir. Örneklem seçiminde farklı yaşlarda ve farklı alanlardan öğretmenler seçmeye dikkat edilmiştir. Okul idarecileri de çalışmaya dahil edilmiştir. Seçilen öğretmenlerin yaş, cinsiyet, meslekteki yılı ve ünvanı gibi bilgileri alınmış kişisel bilgileri çalışmada paylaşılmamıştır. Öğretmenlere kodlama ile isimler verilmiştir. Katılımcıların bilgileri Çizelge 4.1.'deki gibidir.

Çizelge 4.1 Öğretmenlerin bilgileri

KOD	*Cinsiyet*	*Alan*	*Ünvan*	*Yaş*	*Meslekteki Yılı*
Ö1	Bayan	Çocuk Gelişimi	Öğretmen	29	5
Ö2	Erkek	Bilişim Teknolojileri	Öğretmen	26	4
Ö3	Erkek	Bilişim Teknolojileri	Müdür Yardımcısı	34	8
Ö4	Bayan	İngilizce	Öğretmen	42	20
Ö5	Erkek	Motorlu Araçlar Tek.	Okul Müdürü	43	23
Ö6	Bayan	İngilizce	Öğretmen	40	15
Ö7	Bayan	Rehberlik	Öğretmen	26	2
Ö8	Bayan	Muhasebe	Müdür Yardımcısı	44	23
Ö9	Erkek	Matematik	Öğretmen	34	7
Ö10	Erkek	Bilişim Teknolojileri	Öğretmen	32	2
Ö11	Erkek	Felsefe	Öğretmen	48	23
Ö12	Bayan	Türk Dili ve Edebiyatı	Öğretmen	27	2
Ö13	Bayan	İngilizce	Öğretmen	34	13
Ö14	Bayan	Matematik	Öğretmen	26	4
Ö15	Erkek	Matematik	Öğretmen	32	5
Ö16	Erkek	Coğrafya	Öğretmen	49	27
Ö17	Erkek	Bilişim Teknolojileri	Öğretmen	31	3

Çizelge 4.1.'i incelediğimizde farklı alanlardan farklı tecrübelerde öğretmenlerin bulunduğu görülmektedir. Farklı yaş kategorilerinden öğretmenler seçerek tecrübeli öğretmenler ile genç öğretmenlerin uygulamaya yaklaşımları görülmek istenmiştir. Branş olarak sayısal ve sözel derslerde projeden yararlanma durumu da incelenmek istenmiştir.

4.2.2. Verileri toplama araçları

Araştırmanın amacına yönelik kaynak taraması yapılarak ve uzman görüşleri alınarak görüşme formu oluşturulmuştur. Form sayesinde nitel veriler elde edilmiştir. Nitel veriler analiz edilerek ulaşılan bulgular rapor edilmiştir.

4.2.3. Araştırma sonucu elde edilen bulgular

Her bir soruya tüm öğretmenlerin verdikleri cevaplar bir araya toplanmış ve cevapların şu şekilde olduğu görülmüştür:

Soru 1: Sınıfınızda kullandığınız uygulamanın kullanılabilirliği hakkında ne düşünüyorsunuz?

Ö1: *"Kullanılabilir olduğunu ve derslerde verimli olacağını düşünüyorum."*

Ö2: *"Sınıf içerisinde kullanılabilirliği çok etkin ve özellikle özel öğrencilerin derse teşvikini ve ilgi açısından gelişimini sağlayabilir."*

Ö3: *"Sınıf içerisinde rahatlıkla uygulanabilir olduğunu düşünüyorum. Öğretmenin tahtaya sürekli gitmesine gerek olmadan tahtayı yönetebiliyor."*

Ö4: *"Öğretmen ve öğrenciler açısından kullanmaya uygundur. Herhangi bir problemle karşılaşmadım."*

Ö5: *"Sınıf içerisinde kullanılabileceğini düşünüyorum. Faydalı olabileceğini düşünüyorum."*

Ö6: *"Sınıf içerisinde rahatlıkla kullanılabilecek ve öğretmenin işini kolaylaştırabilecek bir uygulama."*

Ö7: *"Sınıfımda kesinlikle kullanabileceğim bir uygulama. İnteraktif eğitimlerde de kullanılabilir."*

Ö8: *"Sistemin kullanımına uygun sınıflarda rahatlıkla kullanılabilir bir uygulama."*

Ö9: *"Sınıf içerisinde uygulayıcı pratikleştikten sonra sınıf içerisinde rahatlıkla kullanılabilir. Hızlı ilerlemeyi ve zaman kazancı sağlar."*

Ö10: *"Özellikle teorik derslerin uygulanabilirliği açısından önemlidir. Teorik dersleri eğlenceli hale getireceğini düşünüyorum."*

Ö11: *"Kullanılabilirliği çok güzel. Eğitim öğretim aracı olarak kullanılabilecek niteliktedir. Derste akıllı tahta kullanımında öğretmenlere yardımcı olabilecek bir projedir."*

Ö12: *"Kalabalık olmayan sınıflarda kullanılabilir olduğunu düşünüyorum."*

Ö13: *"Derslerde işe yarayabileceğini düşünüyorum. Tahtaya bağımlılığı azaltır."*

Ö14: *"Derslerde kullanılabilecek bir uygulama olduğunu gördüm. Eğlenceli buldum."*

Ö15: *"Öğrencilerin dikkatini çeken ve merak uyandıran bir uygulama olmuş."*

Ö16: *"Pratik, zaman kazandırıcı ve özellikle dersi bölmeden akış içerisinde öğretmenin gösterim ya da grafiğe müdehalesi kolaylaşmıştır. Faydalı bulmaktayım."*

Ö17: *"Hassasiyeti yeterli seviyede olursa daha etkin kullanılabilir. Oturduğun yerde Mouse kullanmadan el hareketleriyle kolay bir şekilde tahtayı yönetebiliyoruz. Ortopedik açıdan hareketli olunduğu için daha sağlıklıdır."*

Bildirilen görüşlerin frekans ve yüzdeleri çizelge 4.2.'de sunulmuştur.

Çizelge 4.2 Uygulamanın kullanılabilirlik açısından değerlendirilmesi

Kodlar	***Öğretmenler***	***f***	***%***
Kullanılabilir	Ö1, Ö2, Ö3, Ö4, Ö5, Ö6, Ö7, Ö9, Ö10, Ö11, Ö13, Ö14, Ö15, Ö16	14	82,3
Kullanılamaz	Uygun sınıf gerekli. Ö8, Ö12, Ö17 Kalabalık olmayan sınıflarda kullanılabilir. Hassasiyeti artırılırsa kullanılabilir.	3	17,6

Çizelge 4.2. incelendiğinde tüm öğretmenlerin uygulamanın sınıf içerisinde kullanılabileceğini belirttiği görülmüştür. Yavuz ve Coşkun (2008) yaptıkları çalışmada eğitimde teknolojinin yerinde ve zamanında kullanılması durumunda verimli olacağı sonucuna varmışlardır. Cevaplar içerisinde olumsuz görüşler olmasa da Ö8, Ö9, Ö12, Ö17 belirli şartlar sağlanırsa uygulamanın kullanılabileceğini belirtmişlerdir. Çalışmaya uygun sınıflar olmaması, sensörün algılama mesafesi ve hassasiyeti gibi eksiklikler giderilirse mantık olarak kullanılabilir olduğunu söylemişlerdir.

Soru 2: Uygulamanın sınıf yönetimine katkısı hakkında ne düşünüyorsunuz?

Ö1: *"Sınıf içerisinde gezildiği için sınıf hakimiyeti daha kolay sağlanabilir. Öğrencilere arkamızı daha az döneriz . Sınıf yönetimini kolaylaştıracağını düşünüyorum."*

Ö2: *"Sınıf yöneticisi açısından birden duyuyla hitap ettiği için sınıf kontrolü daha etkindir."*

Ö3: *"Sınıf içerisinde serbest dolaşabildiği için öğrencilere daha rahat hakim olur."*

Ö4: *"Öğrencilere sırtımızı dönmediğimiz için sınıf hakimiyeti artıyor. Sınıf içerisinde 5 metreye kadar kullanılabiliyor. Buda sınıf hakimiyetini kolaylaştırır."*

Ö5: *"Tahta başına bağlı kalmadan yazarak değil de sınıf içerisinde gezerek ders anlatılması sınıf hakimiyetini kolaylaştıracaktır."*

Ö6: *Öğrencilere yakın olduğumuz için aralarında dolaşabildiğimiz için sınıf hakimyeti artıyor. Vücut dilini daha etkili kullanabiliyoruz."*

Ö7: *"Sınıf içerisinde dolaşılabileceği için sınıf hakimiyeti kolaylaşır ve sınıfın konuya odaklanmasına katkısı olur. Öğrenci üzerinde daha rahat hakimiyet sağlanır. Özellikle geniş sınıflarda daha kullanışlı olur."*

Ö8: *"Öğrencileri olumlu etkileyeceğini ve derse çekeceğini düşünüyorum."*

Ö9: *"Matematik derslerinde tahtadan uzaklaşma imkanım olacak. Öğrencilerin katılımı ile derse ilgi artacak ve sınıf disiplini daha kolay sağlanacak."*

Ö10: *"Ders anlatma esnasında sınıf içinde özgürce dolaşma imkanı verdiğinden sınıf hakimiyetini arttıracağına inanıyorum."*

Ö11: *"Sınıf yönetimine olumlu etkisi olur. Masaya bağlı kalmadan sınıf içerisinde serbest dolaşmayı sağlar. Bu da sınıf yönetimini kolaylaştırır."*

Ö12: *"Zaman tasarrufu sağladığı için dersin akışına olumlu yönde katkıları olabilir."*

Ö13: *"Tahta yanında durduğumuzda ve sınıfa arkamızı döndüğümüzde sınıfa hakimiyet azalıyor. Bu çalışma sınıf içerisinde hareket ederken tahtayı kullanabiliriz."*

Ö14: *"Tahtaya bağımlı kalmayacağımız için sınıf hakimiyeti kolaylaşıyor. Zamanı verimli kullanmaya katkısı olur."*

Ö15: *"Öğretmen vücut dilini tahtaya dönmeden daha rahat kullanabileceği için sınıf yönetimi kolaylaşacaktır."*

Ö16: *"Öğretmen sınıfı ve tahtayı geniş açıdan alabileceği için sınıf hakimiyeti yüzde yüz artacaktır. Sesli komut ile pratik kullanım birleşince öğrenciyi soru cevap ile konunun içerisinde tutacaktır. Uygulamada sınıf tahta ile öğretmen arasında mevcutta ise öğretmen tahta ile sınıf arasında kalıyor. Bu durum hakimiyeti zayıflatıyordu. Bu eksiklik giderilmiş."*

Ö17: *"Sınıf içerisinde gezildiği sınıf hakimiyeti daha kolay sağlanabilir. Öğrencilere arkamızı daha az döneriz . Öğretmenin bilgisayara oturma bağımlılığını ortadan kaldırır."*

Verilen cevaplar özetlenmiştir. Cevapların frekans ve yüzdeleri Çizelge 4.3.'de sunulmuştur.

Çizelge 4.3 Uygulamanın sınıf yönetimine katkısı

Cevaplar	*Öğretmenler*	*f*	*%*
Sınıf içerisinde rahat dolaşma imkanı sağlar.	Ö1, Ö2, Ö3, Ö4, Ö5, Ö6, Ö7, Ö9, Ö10, Ö11, Ö13, Ö14, Ö16, Ö17	14	82,3
Birden fazla duyu organına hitap etmemizi sağlar.	Ö2	1	5,8
Vücut dilini etkili bir şekilde kullanmamıza olanak tanır.	Ö1, Ö4, Ö6, Ö15	4	23,5
Zaman tasarrufu sağlar.	Ö12	1	5,8

Çizelge 4.3 incelendiğinde öğretmenler, uygulamanın sınıf yönetimine faydalı olacağını söylemişlerdir. Uygulamanın sınıf yönetimine en önemli etkisinin sınıf içerisinde özgürce, tahtaya bağlı kalmadan dolaşabilmek olduğu görülmüştür. Bu rahatlık diğer bütün sınıf yönetimi bileşenlerini etkilediği için en çok bu cevap verilmiştir. Atıcı (2001) etkili sınıf yönetimi sağlanırsa öğrenci katılımının da artacağı görüşünü belirtmiştir.

Soru 3:Uygulamanın öğrenci katılımını ne şekilde etkileyeceğini düşünüyorsunuz?

Ö1: *"Olumlu yönde etkileyeceğini düşünüyorum. Ders başarısını arttırır."*

Ö2: *"Teknolojinin yeni ve etkin kullanımı öğrenci açısından da ilgi ve başarıyı arttırır. Yeni eğitim modelinde aktif öğrenci beklendiği için uygulama öğrencileri sürece aktif olarak dahil edebilir."*

Ö3: *"Öğrenciyi daha aktif yapacaktır. Çünkü öğrenci sürekli tahta ile etkileşim halinde olacaktır."*

Ö4: *"Öğrenci aktif olduğu için ders katılımı artacaktır. Bu durum notlarına yansıyacaktır. Başarıyı yükselteceğini düşünüyorum."*

Ö5: *"Öğrencilerde interaktif olarak sürece dahil olacağından ve eğlenceli olacağından katılımı arttıracaktır."*

Ö6: "*Özellikle İngilizce dersinde öğrenciler tahtaya kalkmakta zorlanıyorlar. Tahtaya kalkma fobisi olan öğrencilerimiz olabiliyor. Bu noktada oturduğu yerden derse katılması öğrenciyi rahat hissettireceğinden olumlu motivasyon sağlayacaktır."*

Ö7: *"Öğrenciyi hem zihinsel hem de fiziksel olarak derse dahil ettiğimiz için daha aktif kılar. Dikkat düzeyini arttıracağı için öğrenme kolaylaşır.."*

Ö8: *"Uygulama görsel zenginliğe sahip olduğundan öğrenci katılımını arttıracaktır."*

Ö9: *"Öğrenci katılımını artıracağını düşünüyorum. Uygularken de öğrenci ilgisi artmıştır."*

Ö10: *"Ders daha eğlenceli şekilde işleneceğinden ve derse girerken işlem basamaklarından dikkat çekme maddesine en uygun uygulamalardan bir tanesi olduğunu düşünüyorum."*

Ö11: *"Öğrenci katılımını ve derse motivasyonu olumlu yönde etkileyeceğini ve ders verimini arttıracağını düşünüyorum."*

Ö12: *"Öğrencilerin merak duygusunu arttıracağını düşünüyorum. Bu da ders katılımını arttırabilir."*

Ö13: *"Derse katılımı arttırabilir. Sınıf yönetimine bağlı olarak öğrenci motivasyonu da artar. Öğrenciler görevlendirilip derse katılımları arttırılabilir."*

Ö14: *"Tahtaya kalkma çekincesi bulunan öğrencilere kolaylık sağlayacaktır. Motivasyonlarını artıracaktır."*

Ö15: *"Dikkati artırdığından öğrenci katılımını arttıracaktır."*

Ö16: *"Öğrenci sanki hep tahtaya kalkmış gibi etkileşim halinde olacaktır. Öğretmen ile tahta arasında kaldığında öğretmenin göz hapsinde olup dikkatini ders dışına çıkaramayacaktır."*

Ö17: *"Öncelikle öğrencilerin ilgisini çeken bir uygulama olduğu için derse dikkati artıracaktır. Bu durum ders katılımını arttıracaktır ve derslerden elde ettiğimiz verimi üst seviyeye çıkaracaktır."*

Bu soru ile ilgili öğretmen cevapları derlenmiş ve Çizelge 4.4.'de frekans ve yüzdeleri verilmiştir.

Çizelge 4.4 Uygulamanın öğrenci katılımına etkisi

Cevaplar	***Öğretmenler***	***f***	***%***
Öğrencilerin derse aktif katılımını kolaylaştırır.	Ö2, Ö3, Ö4, Ö5, Ö16	5	29,4
Birden fazla duyu organına hitap ettiğimiz için katılım artar.	Ö7, Ö8	2	11,7
Tahtaya kalkma korkusu olanlara kolaylık sağladığından katılımı arttırır.	Ö6, Ö14, Ö16	3	17,6
Eğlenceli olduğu için katılımı arttırır.	Ö10	1	5,8
Derse ilgiyi, dikkati ve motivasyonu artırdığı için katılım artar.	Ö2, Ö7, Ö11, Ö12, Ö13, Ö15, Ö17	7	41

Verilen cevaplar incelendiğinde uygulamanın öğrencilerin aktif katılımlarını sağladığı ve motivasyonunu artırdığı görüşleri ön plana çıkmıştır. Uygulamada birden fazla duyu organına hitap edildiği için zengin bir öğrenme saylayabileceği belirtilmiştir. Sarıtepeci ve Yıldız (2014) genel sınıf içi yüz yüze öğrenme faaliyetleri ile çevrimiçi öğrenmenin güçlü ve olumlu yönlerinin birşeltirilmesi olarak tanımladığı harmanlanmış öğrenme ortamlarının önemi hakkında çalışmalar yapmış, bu ortamların öğrenci motivasyonunu arttırdığını belirtmiştir. Bizim geliştirdiğimiz uygulama da bu tarz ortamların hazırlanmasına zemin hazırlamaktadır.

Soru 4: Uygulamanın engelli öğrencilere katkısı hakkında ne düşünüyorsunuz?

Ö1: *"Özellikle bedensel engelli öğrencilere katkısı olacağını düşünüyorum."*

Ö2: *"İşitme engelli öğrenciler için elle kontrol ve görsel zenginlik kolaylık sağlar. Fiziksel engelliler için uzaktan sesle kontrol kolaylık sağlayabilir."*

Ö3: *"Günümüz teknolojisiyle herhangi bir katkı sağlayabileceğini düşünmüyorum. Fiziksel engellilere oturduğu tahtaya ulaşma imkanı sağlar."*

Ö4: *"Engelli öğrencilerin engelleri kaldırılacak ve derse katılımı sağlayabilecek bir uygulama. Özellikle sesli komutla kollarında problem olan öğrencilerde uygulamaya dahil olabilir."*

Ö5: *"Sesli komutla uygulamayı yönetme görme engelli öğrenciler için faydalı olabilir. İşitme engelli öğrencilerde işaret dili ve görsel materyalleri kullanarak derse katılım sağlayabilir. Destek eğitim odalarından bu sistem kurulabilir."*

Ö6: "*Görme engelli öğrencilere sesli komutla faydalı olabilir."*

Ö7: *"Görme engelli öğrencilere sesli komut kolaylık sağlar. Fiziksel engelli öğrencilerde yerinde otururak derse aktif katılımda bulunabilirler. Zihinsel engellilere yönelik içerik eklenerek katılımları sağlanabilir.."*

Ö8: *"Engelli öğrencilerin özgüvenini arttıracağını düşünüyorum."*

Ö9: *"Engelli öğrencilere çok katkısı olabilir. Görsel engelli öğrencilerde ses komutu çok büyük bir avantaj olur. Fiziksel engelli olanlar için ses komutu ve hareketli komut derse katılımlarını arttırır."*

Ö10: *"Yürüme zorluğu çeken öğrenciler açısından oldukça etkili ve faydalı bir uygulamadır."*

Ö11: *"Özellikle engelli öğrenciler bu uygulamayla dersi içi performanslarının artacağı ve derslere aktif bir şekilde katılımlarının sağlanması yönünde çok katkılar sağlayacağı düşüncesindeyim."*

Ö12: *"Engelli öğrencilerin ders katılımına kolaylık sağlayabilir."*

Ö13: *"Tahtaya kalkma zorunluluğu azaldığı için kolaylık sağlar."*

Ö14: *"Engelli öğrencilere özel uygulamalar geliştirilmesi lazım."*

Ö15: *"Uygulama bu haliyle engelli öğrencilere katkısının az olacağını düşünüyorum. İçerikler öğrencilere göre düzenlenirse faydalı olabilir."*

Ö16: *"Özellikle ortopedik engelli öğrenciler yerinden dersin içerisine girebilecek sunumlarını gerçekleştirebilecektir. Görme engelliler için sesli komut uygulaması sağır ve dilsizler için işaret diline uygun bir projedir."*

Ö17: *"Sesli komutla uygulamayı yönetme görme engelli öğrenciler için faydalı olabilir. İşitme engelli öğrencilerde işaret dili ve görsel materyalleri kullanarak derse katılım sağlayabilir."*

Engelli öğrencilere katkı hakkında verilen cevaplar irdelendiğinde Çizelge 4.5.'deki veriler elde edilmiştir.

Çizelge 4.5 Uygulamanın engelli öğrencilere katkısı

	Cevaplar	*Öğretmenler*	*f*	*%*
Olumlu Düşünceler	Görme engelli öğrencilere katkısı olacağını düşünüyorum.	Ö5, Ö6, Ö7, Ö16, Ö17	5	29,4
	Bedensel engelli öğrencilere katkısı olacağını düşünüyorum.	Ö1, Ö2, Ö3, Ö4, Ö7, Ö10, Ö11, Ö13, Ö16	9	52,9
	İşitme engelli öğrencilere katkısı olacağını düşünüyorum.	Ö2, Ö5, Ö17	3	17,6
	Engelli öğrencilerin öz güvenini arttırır ve ders katılımını arttırır.	Ö1, Ö2, Ö4, Ö5, Ö6, Ö7, Ö8, Ö9, Ö10, Ö11, Ö12, Ö13, Ö16, Ö17	14	82,3
Olumsuz Düşünceler	Bir katkısı olacağını düşünmüyorum.	Ö3	1	5,8
	Mevcut içeriklerle katkısı olacağını düşünmüyorum. Özel uygulamalar geliştirilmesi lazım.	Ö14, Ö15	2	11,6

Uzun ve ark. (2013) zihinsel engellilere yönelik yaptıkları çalışmalarda bu teknolojinin engelli öğrencilerde kullanabileceği fikrini desteklemişlerdir. Öğretmenlerimize yönelttilen sorulardan aldığımız görüşlere göre tüm engelli öğrencilere yönelik çalışmalar yapılabilir. Kinect teknolojisinin bu alana yönelik çalışmalara uygun olduğu görülmüştür.

Soru 5 Uygulamanın eksik yönleri olduğunu düşünüyor musunuz? Var ise bu eksiklikler nelerdir?

Ö1: *"Sensör etkileşiminde zorlandım. İçerikler geliştirilebilir."*

Ö2: *"Türkçe dil desteği olmaması bir eksiklik olabilir."*

Ö3: *"Yeni bir uygulama olduğu için eksiklikleri olduğunu düşünüyorum. Öğretmenler tarafından kullanıldıkça öğretmenlerin fikirlerine göre yeni içerikler oluşturulabilir."*

Ö4: *"5 metreden daha uzakta çalışabilse daha faydalı olur. Bu şekilde toplantılarda konferans salonlarında veya anfilerde de kullanılabilir. Office programlarının dosyaları açılabilir. Ayrıca sadece bu uygulama değilde bilgisayarın komple yönetimi sağlanabilirse daha iyi olur."*

Ö5: *"Gürültülü ortamda sesli komutla yönetmek zor olabilir. Öğrenciler süreci hareketleriyle olumsuz yönde etkileyebilir. Bu açıdan önlem alınması gerekir."*

Ö6: "*Uygulamayı rahat kullanmak için biraz adaptasyon gerekiyor. Parazitler kullanımı zorlaştırabiliyor. Öğrenciler art niyetli hareketlerle ders işlenişini bozabiliyor. Öğretmen üzerine bir işaret koyularak diğer kişiler öğretmen izin vermeden kullanamaması lazım."*

Ö7: *"Sesli komut İngilizce olması zorluk çıkarabilmektedir. Etkileşim mesafesi daha yüksek olursa daha büyük sınıflarda kullanılabilir."*

Ö8: *"Etkileşim mesafesi ve uygulama ara yüzü geliştirilebilir. İçerikler zenginleştirilebilir."*

Ö9: *"Hareketlerin hassasiyetinin düşük olduğunu gördüm. 3 boyutlu nesnelerin görüntülenmesi ders içinde daha faydalı olabilir. Gösterilen materyaller üzerine direk çizim yapılması daha iyi olur. Sadece yapılan uygulama değil de masa üstünü de yönetmek daha güzel olur. Sesli komutların Türkçeleştirilmesi daha kolay olur. "*

Ö10: *"Uygulama içerisinde yapılacak işlemler sınırlıdır. Bu bakımdan yapılacak uygulamaların geliştirilmesi önem arz etmektedir."*

Ö11: *"Öğretmenlerin kendi branşları ile ilgili materyal yükleyebilecekleri bir arayüz oluşturulabilirse daha güzel ve verimli olur."*

Ö12: *"İçerik genişletilebilir. Türkçe desteği eklenebilir."*

Ö13: *"Materyaller eklenebilir. İngilizceye yönelik interaktif uygulamalar da olması gerekir. Söylenenler tahta ya yazılması kolaylık sağlar. Telafuz çalışmaları eklenebilir. Konuşma egzersizleri yapılabilir. Grup çalışmalarına ağırlık verilmeli. Ekrandaki materyaller üzerinde yazma işlemi yapılması kolaylık sağlar."*

Ö14: *"Algılamada zorluklar yaşandı. Mesafesi arttırılabilir. Ses algılaması gürültüden etkilenebilmektedir."*

Ö15: *"İlk kullanımda değişik gelebiliyor. Öğretmenlere kısa bir eğitim verilmesi gerekiyor. "*

Ö16: *"Özellikle ortopedik engelli öğrenciler yerinden dersin içerisine girebilecek sunumlarını gerçekleştirebilecektir. Görme engelliler için sesli komut uygulaması sağır ve dilsizler için işaret diline uygun bir projedir."*

Ö17: *"En büyük eksiklik öğretmenlerin hazırbulunuşluluğudur. Uygulamanın formasyon eğitimi içerisinde Öğretmen adaylarına verilmesi etkin kullanımı kolaylaştırıp gelişmesini sağlayacaktır."*

Çalışmanın eksik yönleri hakkında öğretmenlerin verdikleri cevaplar da belirli başlıklar altında tasnif edilerek cevapların Çizelge 4.6'daki gibi frekans ve yüzdeleri verilmiştir.

Çizelge 4.6 Uygulamanın eksik yönleri hakkında görüşler

Cevaplar	***Öğretmenler***	***f***	***%***
Sensör hassasiyeti ve etkileşimi yetersiz.	Ö1, Ö4, Ö8, Ö9, Ö14	5	29,4
Türkçe dil desteği olması gerekir.	Ö2, Ö7, Ö9, Ö12	4	23,5
Uygulama içeriği geliştirilmeli.	Ö1, Ö3, Ö4, Ö8, Ö9, Ö10, Ö11, Ö13	8	47
Öğretmenlerin konu ile ilgili eğitimi gereklidir.	Ö3, Ö6, Ö15, Ö17	4	23,5
Öğrencilerin suistimali ve parazitlere yönelik önlemler alınmalı.	Ö5, Ö6, Ö14	3	17,6

Tabloda elde edilen veriler uygulamaların eksiklikleri hakkında bilgiler vermektedir. En yüksek oranda verilen cevap %47 ile içeriğin geliştirilmesi yönünde olmuştur. En düşük olarak %17,6 ile öğrencilerin süreci suistimal etmesi olmuştur.(Tenekeci ve Gumuscu, 2016) okuma yazma faaliyetlerinde kinect uygulaması gerçekleştirilmiştir. Bizim projemizde ortaöğretim kurumlarında kullanılan materyaller kinect ile gösterilip yönetilmiştir.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Kinect sensör ülkemizde satışta olmasa da online alışveriş web siteleri sayesinde artık her ürüne kolaylıkla erişmek mümkün olmaktadır. Elde edilen sensör Milli Eğitim Bakanlığı'na ait Şarköy Mesleki ve Teknik Eğitim Merkezinde denenmiştir. Sensör öncelikle yeterlilik anlamında test edilmiş ve sınıf içerisinde sensöre erişimin hem sesli hemde işaret ile mümkün olduğu görülmüştür. Sesli erişim normal mikrofonlara göre daha uzak mesafelerden sağlanmıştır. Sınıf içerisinde bir çok bölgeden el kol hareketleriyle sensörle etkileşime geçilebilmiştir. Ayrıca kızılötesi sayesinde ortamda ki ışık az olsa bile sistem çalışabilmektedir. Daha sonra geliştirilen uygulama sınıf içerisinde denenmiş ve uygulamanın büyük ölçekte amacına ulaştığı öğretmen görüşlerinde görülmüştür. Beş soruya da verilen cevaplar detaylıca incelenmiştir.

Birinci soru sensörün kullanılabilirliği hakkında olup verilen cevaplar büyük ölçüde sensörün sınıf içerisinde eğitim amaçlı kullanılabileceğini göstermiştir. Örneklem seçimi yapılırken farklı yaşta, farklı cinsiyette, farklı alanda öğretmenler seçilmeye dikkat edilmiştir. Verilen cevaplarda tecrübeli ve genç öğretmenlerin uygulamaya olumlu baktığı görülmüştür. Sözel ve sayısal alanlarda ki öğretmenlerinde % 100'e yakını projeye olumlu bakmıştır. Bu durum uygulamanın her derste her yaşta öğretmen tarafından kolaylıkla kullanılabileceğini göstermiştir. Öğretmenler tarafından direk olarak kullanılamaz diyen çıkmamıştır. Fakat belirli şartlar altında kullanılabileceğini belirtenler olmuştur. Bu şartlardan birisi uygun sınıfların olmaması olarak belirtilmiştir. Bu cevap incelendiğinde uygulamanın test edildiği sınıf 7.2 metre uzunluğunda bir sınıftır. Sensör dikey olarak 4.8 metreye kadar çalışabilmiştir. Fakat öğretmen açısından bakıldığında sınıfın en ucunda kullanılamasa bile sınıf ortasından rahatlıkla kullanılabiliyor olması büyük bir kolaylık sağlamaktadır. Sensör hassasiyeti ile ilgili eksiklik giderilirse kullanılabileceği belirtilmiştir. Bu cevap incelendiğinde sensörü ilk defa kullananlar "engagement" denilen sistemle etkileşime geçme mantığını tam olarak kavrayamadığından zorluklar yaşamıştır. Bir kaç kullanımdan sonra uygulama rahatlıkla kullanılabilmektedir. Hassasiyet sorunlarını gidermek için tıklanacak olan düğmeler büyük boyutlu tasarlanmıştır. Sensörün kullanımı aynı anda 6 kişiye kadar desteklemekte olduğu için sınıf ortamında öğrenciler aynı anda etkileşime

geçemeyebilirler. Sonuç olarak birinci soru incelendiğinde sensörün derste kullanılabileceği görüşü bildirilmiştir.

İkinci sorumuz uygulamanın sınıf yönetimine ne gibi etkilerini olacağını görmek için sorulmuştur. Öğretmenlerin hepsi böyle bir uygulamanın sınıf yönetimini kolaylaştıracağını söylemişlerdir. Sınıf yönetimin zorlaşmasında en büyük etkenlerden birisi de öğretmenlerin tahtaya bağlı kalması ve sınıfa arkasını dönmek zorunda kalmasıdır. Öğretmenlerin %82si sınıf içerisinde rahat dolaşmanın sınıf yönetimini kolaylaştıracağını söylemişlerdir. Sınıf içerisinde gezebilen öğretmen vücut dilini daha iyi kullanabilecek, öğrencilerle birebir etkileşimde olacağı için sınıf içi disiplini daha rahat sağlayabilecektir. Sınıfa arkasını dönmesine gerek kalmaması ise sürekli öğrenci davranışlarını takip edebilme imkanı sağlayacaktır. Etkili sınıf yönetimi öğrenci katılımını ve ders başarısını da arttırabilecektir.

Üçüncü soruda öğrenci katılımını ne şekilde etkileyeceği sorusu yöneltilmiş ve öğrenci katılımını olumlu yönde etkileyeceği görüşü bildirilmiştir. Uygulama öğrencilerin ilgi ve motivasyonunu artıracağı söylenmiştir. Uygulama fiziksel aktivitelerde içerdiği için öğrencilerin aktif katılımını mümkün kılmaktadır. Eğitimde ne kadar çok duyu organına hitap edilirse öğrenme o kadar yüksek oranda gerçekleşir. Ayrıca birden fazla duyu organına hitap ettiği için eğitimi ezbercilikten kurtarıp uygulayıcı bir hale getirebiliriz. Bu durum kalıcı öğrenmeyi sağlayabilecektir. Yeni öğretim modellerinde öğretmenin rehber olması ve öğrencinin sürece aktif olarak katılması istenmektedir. Uygulama bu sürece katkıda bulunabilecektir.

Dördüncü soruda engelli öğrencimize bu uygulamanın neler katabileceğini sorduk. Verilen cevaplardan uygulamanın bedensel, zihinsel, görme ve işitme engelli öğrencilere katkıları olabileceği söylenmiştir. Günümüzde engelli öğrencilerin eğitim süreçlerine aktif olarak katılması istenmektedir. Son yıllarda engelli öğrencilere kaynaştırma eğitimi verilmesi istenmektedir. Kaynaştırma eğitimi özel gereksinimli öğrencilerin gerektiğinde ek destek eğitimler ve etkinlikler verilerek normal sınıf içerisinde eğitim görmesinin sağlanmasıdır. Sınıfta giderilemeyen eksiklikler için oluşturulan kaynak odalar kullanılmaktadır(Batu, 2000). Kaynak oda engelli öğrenciye göre dizayn edilir ve sınıfta verilemeyen kazanımlar burada öğrenciye özel olarak verilir. Bu açıdan okullarımızda erişilebilirlik çalışmaları yapılmış ve halen devam eden çalışmalar bulunmaktadır. Uygulamanın kaynak odalarda kullanılabileceği görülmüştür. Engelli öğrencilerinde örgün eğitimde olması çok önemlidir. Bu uygulamada engelli öğrencilerin sınıfta derse aktif katılımlarını sağlamaya yönelik bir uygulamadır. Örneğin

ayağa kalkamayan bir öğrenci oturarak el hareketleriyle hazırladığı bir sunuyu sınıf içerisinde anlatabilir. Kendini sürecin bir parçası olarak hissedebilir. Görme engelli öğrenciler de sesli komutlarla tahtayı yönetebilirler. Aktif katılım engelli öğrencilerin özgüvenlerini arttırabilecektir. Tablo incelendiğinde verilen olumsuz görüşlerde görülmektedir. Uygulamanın engelli öğrencilere daha çok katkısı olması için özel içerikler hazırlanması gerektiği görüşü bildirilmiştir.

Beşinci soruda ise yapılacak olan çalışmlara fikir olması amacıyla uygulamanın eksiklikleri sorulmuştur. En çok verilen cevap %47 ile içeriğin geliştirilmesi olmuştur. Kullanıcılar uygulamaya istedikleri içerikleri yükleyebilmektedirler. Burada kullanıcılar uygulama arayüzünün geliştirilmesi hakkında isteklerde bulunmuşlardır. Matematik öğretmenleri uygulamanın 3 boyutlu nesneleri gösterme gibi bir bölümünün olabileceğini belirtmişlerdir. Bir diğer eksiklik olarak sensör hassasiyeti görülmüştür. Burada öğretmenlerin ilk defa kullanıyor olması da sensör algılamasının kötü olduğu düşüncesine sebep olmuştur. Türkçe dil desteği olmaması da bir eksiklik oalrak görülmüştür. Bu durum Microsoft Speech Recognition uygulamasının Türkçe destek vermemesinden kaynaklanmıştır. Bu yüzden işletim sisteminin dili İngilizce'ye çevrilmek zorunda kalınmıştır. Öğretmenlerin konu ile ilgili hazır bulunuşluğu da eksiklik olarak görülmüştür. Fakat iki veya üç defa kullanan öğretmenler uygulamaya rahatlıkla alışabilmişlerdir.

5.2 Öneriler

Günümüzde kaynakların ve zamanın verimli kullanılması için eğitimde teknoloji kullanımı önemli hale gelmiştir. Fatih projesi ile okullarımıza etkileşimli tahtalar gelmiş fakat tam anlamıyla kullanmak için daha fazla döküman ve uygulamaya ihtiyaç vardır. Bu noktada öğretmenler de bu süreci geliştirmenin bir parçası olabilir. Geliştirilen uygulama da resim gösterme, kitap gösterme, video oynatıcı ve sunu gösterici olarak geliştirilmiştir. Öğretmenlerden alınan görüşlerde 3 boyutlu nesneleri gösterme ve döndürme işlevi eklenebilir. Örneğin matematik dersinde 3 boyutlu cisimler gösterilip kinect sensör sayesinde döndürülebilir. Bir el hareketiyle kesiti alınıp oluşan şekiller ve izdüşümleri öğrenciye gösterilebilir. Fen Bilimleri derslerinde organların 3 boyutlu modelleri oluşturulup yine kinect ile bu organlar döndürülebilir. Özel eğitim gereken öğrencilere engeline yönelik arayüzler geliştirilebilir. Coğrafya ve Tarih derslerinde kullanılmak üzere bir Google Earth arayüzü eklenip yeni hareketler tanımlayarak harita üzerinde gezinti yapılabilir. Bazı öğretmenler harici bir uygulama yerine tüm tahta yönetiminin sensör tarafından yapılabilmesinin daha faydalı olabileceğini

belirtmişlerdir. Söylenenleri yazma gibi bir işlevde uygulamaya faydalı olabilir. Türkçe dili desteği olmaması da kullanıcıları zorlyayabilmektedir. Bu noktada bir yapay sinir ağı modeli sayesinde Türkçe komutlardan elde edilen özellik vektörleri eğitim aşamasına sokularak bir öğrenme gerçekleştirilebilir. Daha sonra bu komutları tanıma ile Türkçe desteğide sağlanmış olur. Bu uygulama Visual Gesture Builder ile benzer bir şekilde yapılabilir. Uygulamada kişilere kimlik verilebilir. Örneğin uygulama açılınca öğretmeni tanır ve onun izinlerini açar. Öğrecilerin ise sadece öğretmenin onay vermesi durumunda etkileşime geçmesi sağlanabilir. İleride yapılacak olan çalışmalarada özellikle özel gereksinime ihtiyaç duyan öğrencilere yönelik etkinlikler hazırlanabilir.

KAYNAKLAR

Adigüzel, T., Gürbulak, N. ve Sariçayir, H., 2011, Akıllı Tahtalar Ve Öğretim Uygulamaları/Smart Boards And Their Instructional Uses, *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 8 (15).

Akıncı, A., Kurtoğlu, M. ve Seferoğlu, S. S., 2012, Bir teknoloji politikası olarak Fatih Projesinin başarılı olması için yapılması gerekenler: Bir durum analizi çalışması, *Akademik Bilişim Konferansı*, 1-3.

Albayrak, E., 2014, Fatih projesi kapsamında ki okullarda bilişim teknolojileri kullanımının sınıf yönetimi açısından değerlendirilmesi, *Akdeniz Üniversitesi*, Antalya, 6-24.

Altan, T. ve Tüzün, H., 2011, Teknoloji-zengin bireysel öğrenme ortamlarının FATİH projesindeki yeri, *Akademik Bilişim*, 11.

Amon, C., Fuhrmann, F. ve Graf, F., 2014, Evaluation of the spatial resolution accuracy of the face tracking system for kinect for windows v1 and v2, *Proceedings of the 6th Congress of the Alps Adria Acoustics Association*, 16-17.

Andersen, M. R., Jensen, T., Lisouski, P., Mortensen, A. K., Hansen, M. K., Gregersen, T. ve Ahrendt, P., 2012, Kinect depth sensor evaluation for computer vision applications, *Electrical and Computer Engineering Technical Report ECE-TR-6*.

Atıcı, M., 2001, Yüksek ve düşük yetkinlik düzeyine sahip öğretmenlerin sınıf yönetimi stratejileri, *Kuram ve Uygulamada Egitim Yönetimi Dergisi*, 7 (4), 483-499.

Avcı, Ü. ve Seferoğlu, S. S., 2011, Bilgi toplumunda öğretmenin tükenmişliği: Teknoloji kullanımı ve tükenmişliği önlemeye yönelik alınabilecek önlemler, *Akdeniz Eğitim Araştırmaları Dergisi*, 9, 13-26.

Batu, E. S., 2000, Kaynaştırma, destek hizmetler ve kaynaştırmaya hazırlık etkinlikleri, *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Özel Eğitim Dergisi*, 2 (04).

bin Mohd Sidik, M. K., bin Sunar, M. S., bin Ismail, I. ve bin Mokhtar, M. K., 2011, A study on natural interaction for human body motion using depth image data, *Digital Media and Digital Content Management (DMDCM), 2011 Workshop on*, 97-102.

Biswas, K. K. ve Basu, S. K., 2011, Gesture recognition using microsoft kinect®, *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, 100-103.

Cruz, L., Lucio, D. ve Velho, L., 2012, Kinect and rgbd images: Challenges and applications, *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on*, 36-49.

Demiralp, N., 2007, Coğrafya eğitiminde materyaller ve 2005 coğrafya dersi öğretim programi, *Kastamonu Eğitim Dergisi*, 15 (1), 373-384.

Dursun, Ö. Ö., Kuzu, A., Kurt, A. A., Güllüpınar, F. ve Gültekin, M., 2013, Okul yöneticilerinin FATİH Projesinin pilot uygulama sürecine ilişkin görüşleri, *Trakya Üniversitesi Eğitim Fakültesi Dergisi*, 3 (1).

Erdoğan, M., Kurşun, E., Tan Şişman, G., Saltan, F., Gök, A. ve Yıldız, İ., 2010, Sınıf yönetimi ve sınıf içi disiplin problemleri, nedenleri ve çözüm önerileri üzerine nitel bir araştırma: Bilişim teknolojileri dersi örneği, *Kuramdan Uygulamaya Eğitim Bilimleri–Educational Sciences: Theory & Practice*, 10 (2), 853-891.

Frati, V. ve Prattichizzo, D., 2011, Using Kinect for hand tracking and rendering in wearable haptics, *World Haptics Conference (WHC), 2011 IEEE*, 317-321.

Fürntratt, H. ve Neuschmied, H., 2014, Evaluating pointing accuracy on Kinect V2 sensor, *International Conference on Multimedia and Human-Computer Interaction (MHCI)*, 124.121-124.125.

Hsu, H.-m. J., 2011, The potential of Kinect in education, *International Journal of Information and Education Technology*, 1 (5), 365.

Jana, A., 2012, Kinect for windows SDK programming guide, Packt Publishing Ltd, p.

Kang, J.-w., Seo, D.-j. ve Jung, D.-s., 2011, A study on the control method of 3-dimensional space application using kinect system, *International Journal of Computer Science and Network Security*, 11 (9), 55-59.

Keser, H. ve Çetinkaya, L., 2013, Öğretmen ve Öğrencilerin Etkileşimli Tahta Kullanımına Yönelik Yaşamış Oldukları Sorunlar ve Çözüm Önerileri, *Electronic Turkish Studies*, 8 (6).

Khoshelham, K., 2011, Accuracy analysis of kinect depth data, *ISPRS workshop laser scanning*, W12.

Kurt, A. A., Abdullah, K., Dursun, Ö. Ö., GÜLLEPINAR, F. ve Gültekin, M., 2013, FATİH projesinin pilot uygulama sürecinin değerlendirilmesi: Öğretmen görüşleri, *Journal of Instructional Technologies & Teacher Education*, 2 (1).

LaBelle, K., 2011, Evaluation of Kinect joint tracking for clinical and in-home stroke rehabilitation tools, *Undergraduate Thesis, University of Notre Dame*.

Lachat, E., Macher, H., Mittet, M., Landes, T. ve Grussenmeyer, P., 2015, First experiences with Kinect v2 sensor for close range 3D modelling, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40 (5), 93.

Li, B. Y., Mian, A. S., Liu, W. ve Krishna, A., 2013, Using kinect for face recognition under varying poses, expressions, illumination and disguise, *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, 186-192.

Özden, Y., 2005, Eğitimde yeni değerler, *Ankara: Pegem A Yayıncılık*.

Paliç, G. ve Keleş, E., 2011, Sınıf yönetimine ilişkin öğretmen görüşleri, *Kuram ve Uygulamada Eğitim Yönetimi*, 2 (2), 199-220.

Pamuk, S., Çakır, R., Ergun, M., Yılmaz, H. B. ve Ayas, C., 2013, Öğretmen ve öğrenci bakış açısıyla tablet PC ve etkileşimli tahta kullanımı: FATİH Projesi değerlendirmesi.

Pedersoli, F., Adami, N., Benini, S. ve Leonardi, R., 2012, XKin-: eXtendable hand pose and gesture recognition library for kinect, *Proceedings of the 20th ACM international conference on Multimedia*, 1465-1468.

Sarıtepeci, M. ve Yıldız, H., 2014, Harmanlanmış öğrenme ortamlarının öğrencilerin derse katılım ve derse karşı motivasyonları üzerine etkisinin incelenmesi, *Ahi Evran Üniversitesi Kırşehir Eğitim Fakültesi Dergisi*, 15 (1).

Sells, C. ve Griffiths, I., 2005, Programming Windows presentation foundation, " O'Reilly Media, Inc.", p.

Süzen, A. A. ve Taşdelen, K., 2013, Microsoft Kinect ile örnek bir uygulama: Piyano, *Akademik Bilişim Konferansı, Antalya*, 23-25.

Şen, S., 2006, Sınıf İçi İletişimde Beden Dili, *GAZİ ÜNİVERSİTESİ*, Ankara.

Tenekeci, M. E. ve Gumuscu, A., 2016, Okuma Yazma Eğitiminde Kinect Kullanımı, XVI. Akademik Bilişim Konferansı'nda sunulan bildiri. Mersin Üniversitesi, Antalya, Türkiye.

Türnüklü, A., 2000, Eğitimbilim araştırmalarında etkin olarak kullanılabilecek nitel biraraştırma tekniği: Görüşme, *Kuram ve Uygulamada Egitim Yönetimi Dergisi*, 6 (4), 543-559.

Uzun, F. D., Gülen, Ş. B., Uzun, C., Çakır, H., Çağıltay, K., Karasu, N. ve Akıllı, G., 2013, Beden Hareketleriyle Kontrol Edi len Kinect Teknolojisinin Zihinsel Engelli Öğrenciler Açısından Eğitimde Kullanım Potansiyeli. 7, *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, 6-8.

Yavuz, S. ve Coşkun, E. A., 2008, Sınıf öğretmenliği öğrencilerinin eğitimde teknoloji kullanımına ilişkin tutum ve düşünceleri, *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 34 (34).

Zhang, Z. Y., 2012, Microsoft Kinect Sensor and Its Effect, *Ieee Multimedia*, 19 (2), 4-10.

EKLER

EK-1 Öğretmenlere uygulanan görüşme formu.

ÖĞRETMEN GÖRÜŞME FORMU

BRANŞ :
CİNSİYET :
YAŞI :
MESLEKTEKİ YILI:

Değerli meslektaşım,

Sizlerin de bildiği gibi FATİH projesi ile okullara etkileşimli tahtalar kurulmuş ve internet ağı alt yapısı çekilmiştir. Gelişen teknolojiyle okullarımızda da teknoloji kullanımı artmıştır. Etkileşimli tahtaların kullanımıyla döküman ve uygulama ihtiyaçları ortaya çıkmıştır. Alt yapı iyi yerlere gelmiş fakat halen kullanımda zorluklar yaşanabilmektedir. Bu eksikliği gidermede biz öğretmenlere de önemli görevler düşmektedir. Bu amaçla geliştirilen yazılımımızı sınıfınızda denediğiniz için teşekkür ederiz. Bu yazılım hakkında ki görüşleriniz çalışmamızın değerlendirmesi açısından önemlidir. Bu yüzden sorularımıza verdiğiniz objektif cevaplar için şimdiden teşekkür ederiz.

Soru 1) Sınıfınızda kullandığınız uygulamanın kullanılabilirliği hakkında ne düşünüyorsunuz?(uygulanabilirlik açısından)
Cevap:

Soru 2) Uygulamanın sınıf yönetimine katkısı hakkında ne düşünüyorsunuz?
Cevap:

Soru 3) Uygulamanın öğrenci katılımını ne şekilde etkileyeceğini düşünüyorsunuz?
Cevap:

Soru 4) Uygulamanın engelli öğrencilere katkısı hakkında ne düşünüyorsunuz?
Cevap:

Soru 5) Uygulamanın eksik yönleri olduğunu düşünüyor musunuz? Var ise bu eksiklikler nelerdir?
Cevap:

Verdiğiniz cevaplar için teşekkür ederiz.

EK-2 Uygulamanın C# ve XAML kodları.

```
//------------------------------------------------------------------------
---
// <copyright file="GestureDetector.cs" company="Microsoft">
//     Copyright (c) Microsoft Corporation.  All rights reserved.
// </copyright>
//------------------------------------------------------------------------
---

namespace WPF_ders_1
{
    using System;
    using System.Collections.Generic;
    using Microsoft.Kinect;
    using Microsoft.Kinect.VisualGestureBuilder;
    using System.Windows;
using System.IO;
    using System.Windows.Media.Imaging;
    using System.Windows.Media.Animation;
    using System.Windows.Controls;


    /// <summary>
    /// Gesture Detector sınıfı  for VisualGestureBuilderFrame olaylarını
dinlemek için yazılmıştır.
    /// and updates the associated GestureResultView object with the latest
results for the gesture
    /// </summary>
    public class GestureDetector : IDisposable
    {
        //public RoutedEventHandler GestureRecognized { get; set; }

        //important lab 13
        /// <summary> Path to the gesture database that was trained with VGB
</summary>
        private readonly string gestureDatabase =
@"C:\Users\salihgunduz\Documents\visual studio 2013\Projects\WPF ders 1\WPF
ders 1\gestureDatabase\NP.gbd";

        //important lab 13
        /// <summary> Name of the discrete gesture in the database that we
want to track </summary>
        private readonly string leftHand = "NP_Left";
        private readonly string rightHand = "NP_Right";
        private int kontrol = 0;
        /// <summary> Gesture frame source which should be tied to a body
tracking ID </summary>
        private VisualGestureBuilderFrameSource vgbFrameSource = null;

        /// <summary> Gesture frame reader which will handle gesture events
coming from the sensor </summary>
        private VisualGestureBuilderFrameReader vgbFrameReader = null;

        /// <summary>
        /// Initializes a new instance of the GestureDetector class along
with the gesture frame source and reader
        /// </summary>
```

```
        /// <param name="kinectSensor">Active sensor to initialize the
VisualGestureBuilderFrameSource object with</param>
        /// <param name="gestureResultView">GestureResultView object to store
gesture results of a single body to</param>
        ///  int selected = 0;
        List<String> image_names = new List<String>();

        public GestureDetector(KinectSensor kinectSensor, GestureResultView
gestureResultView)
        {
            if (kinectSensor == null)
            {
                throw new ArgumentNullException("kinectSensor");
            }

            if (gestureResultView == null)
            {
                throw new ArgumentNullException("gestureResultView");
            }

            this.GestureResultView = gestureResultView;

            // create the vgb source. The associated body tracking ID will be
set when a valid body frame arrives from the sensor.
            this.vgbFrameSource = new
VisualGestureBuilderFrameSource(kinectSensor, 0);
            this.vgbFrameSource.TrackingIdLost += this.Source_TrackingIdLost;

            // open the reader for the vgb frames
            this.vgbFrameReader = this.vgbFrameSource.OpenReader();
            if (this.vgbFrameReader != null)
            {
                this.vgbFrameReader.IsPaused = true;
                this.vgbFrameReader.FrameArrived +=
this.Reader_GestureFrameArrived;
            }

            // load the gesture from the gesture database
            using (VisualGestureBuilderDatabase database = new
VisualGestureBuilderDatabase(this.gestureDatabase))
            {
                // we could load all available gestures in the database with
a call to vgbFrameSource.AddGestures(database.AvailableGestures),
                // but for this program, we only want to track one discrete
gesture from the database, so we'll load it by name
                //foreach (Gesture gesture in database.AvailableGestures)
                //{
                //    if (gesture.Name.Equals(this.seatedGestureName))
                //    {
                //        this.vgbFrameSource.AddGesture(gesture);
                //    }
                //}

                // we could load all available gestures in the database with
a call to vgbFrameSource.AddGestures(database.AvailableGestures),
                // but for this program, we only want to track one discrete
gesture from the database, so we'll load it by name
                foreach (Gesture gesture in database.AvailableGestures)
                {
```

```
                    this.vgbFrameSource.AddGesture(gesture);

                }
            }
        }

        /// <summary> Gets the GestureResultView object which stores the
detector results for display in the UI </summary>
        public GestureResultView GestureResultView { get; private set; }

        /// <summary>
        /// Gets or sets the body tracking ID associated with the current
detector
        /// The tracking ID can change whenever a body comes in/out of scope
        /// </summary>
        public ulong TrackingId
        {
            get
            {
                return this.vgbFrameSource.TrackingId;
            }

            set
            {
                if (this.vgbFrameSource.TrackingId != value)
                {
                    this.vgbFrameSource.TrackingId = value;
                }
            }
        }

        /// <summary>
        /// Gets or sets a value indicating whether or not the detector is
currently paused
        /// If the body tracking ID associated with the detector is not
valid, then the detector should be paused
        /// </summary>
        public bool IsPaused
        {
            get
            {
                return this.vgbFrameReader.IsPaused;
            }

            set
            {
                if (this.vgbFrameReader.IsPaused != value)
                {
                    this.vgbFrameReader.IsPaused = value;
                }
            }
        }

        public bool GestureRecognized
        {
            get
            {
                return this.GestureRecognized;
```

```
            }

            set
            {
                if (this.GestureRecognized != value)
                {
                    this.GestureRecognized = value;
                }
            }
        }

        /// <summary>
        /// Disposes all unmanaged resources for the class
        /// </summary>
        public void Dispose()
        {
            this.Dispose(true);
            GC.SuppressFinalize(this);
        }

        /// <summary>
        /// Disposes the VisualGestureBuilderFrameSource and
VisualGestureBuilderFrameReader objects
        /// </summary>
        /// <param name="disposing">True if Dispose was called directly,
false if the GC handles the disposing</param>
        protected virtual void Dispose(bool disposing)
        {
            if (disposing)
            {
                if (this.vgbFrameReader != null)
                {
                    this.vgbFrameReader.FrameArrived -=
this.Reader_GestureFrameArrived;
                    this.vgbFrameReader.Dispose();
                    this.vgbFrameReader = null;
                }

                if (this.vgbFrameSource != null)
                {
                    this.vgbFrameSource.TrackingIdLost -=
this.Source_TrackingIdLost;
                    this.vgbFrameSource.Dispose();
                    this.vgbFrameSource = null;
                }
            }
        }

        /// <summary>
        /// Handles gesture detection results arriving from the sensor for
the associated body tracking Id
        /// </summary>
        /// <param name="sender">object sending the event</param>
        /// <param name="e">event arguments</param>
        private void Reader_GestureFrameArrived(object sender,
VisualGestureBuilderFrameArrivedEventArgs e)
        {
            VisualGestureBuilderFrameReference frameReference =
e.FrameReference;
```

```
            using (VisualGestureBuilderFrame frame =
frameReference.AcquireFrame())
            {
                if (frame != null)
                {
                    // get the discrete gesture results which arrived with
the latest frame
                    IReadOnlyDictionary<Gesture, DiscreteGestureResult>
discreteResults = frame.DiscreteGestureResults;

                    if (discreteResults != null)
                    {
                        // we only have one gesture in this source object,
but you can get multiple gestures
                        foreach (Gesture gesture in
this.vgbFrameSource.Gestures)
                        {
                            //if (gesture.Name.Equals(this.seatedGestureName)
&& gesture.GestureType == GestureType.Discrete)
                            //{
                            //    DiscreteGestureResult result = null;
                            //    discreteResults.TryGetValue(gesture, out
result);

                            //    if (result != null)
                            //    {
                            //        // update the GestureResultView object
with new gesture result values
                            //
this.GestureResultView.UpdateGestureResult(true, result.Detected,
result.Confidence);
                            //    }
                            //}

                            if (gesture.Name.Equals(this.rightHand) &&
gesture.GestureType == GestureType.Discrete)
                            {
                                DiscreteGestureResult result = null;
                                discreteResults.TryGetValue(gesture, out
result);

                                if (result.Detected==true)
                                {
                                    // update the GestureResultView object
with new gesture result values


this.GestureResultView.UpdateGestureResult(true, result.Detected,
result.Confidence);



                                }
                            }
                            if (gesture.Name.Equals(this.leftHand) &&
gesture.GestureType == GestureType.Discrete)
                            {
                                DiscreteGestureResult result = null;
```

```
                                discreteResults.TryGetValue(gesture, out
result);

                                if (result.Detected == true)
                                {
                                    // update the GestureResultView object
with new gesture result values


this.GestureResultView.UpdateGestureResult(true, result.Detected,
result.Confidence);



                                }
                            }
                        }
                    }
                }
            }
        }

        private void showNextPrevious(string p)
        {
            DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\img\");
            FileInfo[] images = folder.GetFiles("*.jpg");
            foreach (FileInfo img in images)
            {
                this.image_names.Add("img/" + img.Name);
            }
        }


        /// <summary>
        /// Handles the TrackingIdLost event for the
VisualGestureBuilderSource object
        /// </summary>
        /// <param name="sender">object sending the event</param>
        /// <param name="e">event arguments</param>
        private void Source_TrackingIdLost(object sender,
TrackingIdLostEventArgs e)
        {
            // update the GestureResultView object to show the 'Not Tracked'
image in the UI
            this.GestureResultView.UpdateGestureResult(false, false, 0.0f);
        }

    }
}


//------------------------------------------------------------------------
---
// <copyright file="GestureResultView.cs" company="Microsoft">
//     Copyright (c) Microsoft Corporation.  All rights reserved.
// </copyright>
```

```
//------------------------------------------------------------------------------
---

namespace WPF_ders_1
{
    using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Runtime.CompilerServices;
    using System.Windows;

    /// <summary>
    /// Stores discrete gesture results for the GestureDetector.
    /// Properties are stored/updated for display in the UI.
    /// </summary>
    public sealed class GestureResultView : INotifyPropertyChanged
    {
        /// <summary> The body index (0-5) associated with the current
gesture detector </summary>
        private int bodyIndex = 0;
        public List<String> image_names = new List<String>();
        /// <summary> Current confidence value reported by the discrete
gesture </summary>
        private float confidence = 0.0f;
        private int hareket_kontrol = 0;
        /// <summary> True, if the discrete gesture is currently being
detected </summary>
        private bool detected = false;
        private string img="img/uk-england-beach-bay-sea-pier-lights-evening-
sunset-sky-clouds-architecture-ocean-high-resolution.jpg";
        /// <summary> True, if the body is currently being tracked </summary>
        private bool isTracked = false;
        private DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\img\");

        /// <summary>
        /// Initializes a new instance of the GestureResultView class and
sets initial property values
        /// </summary>
        /// <param name="bodyIndex">Body Index associated with the current
gesture detector</param>
        /// <param name="isTracked">True, if the body is currently
tracked</param>
        /// <param name="detected">True, if the gesture is currently detected
for the associated body</param>
        /// <param name="confidence">Confidence value for detection of the
'Seated' gesture</param>
        public GestureResultView(int bodyIndex, bool isTracked, bool
detected, float confidence)
        {
            this.BodyIndex = bodyIndex;
            this.IsTracked = isTracked;
            this.Detected = detected;
            this.Confidence = confidence;
        }

        /// <summary>
```

```
        /// INotifyPropertyChangedPropertyChanged event to allow window
controls to bind to changeable data
        /// </summary>
        public event PropertyChangedEventHandler PropertyChanged;

        /// <summary>
        /// Gets the body index associated with the current gesture detector
result
        /// </summary>
        public int BodyIndex
        {
            get
            {
                return this.bodyIndex;
            }

            private set
            {
                if (this.bodyIndex != value)
                {
                    this.bodyIndex = value;
                    this.NotifyPropertyChanged();
                }
            }
        }

        /// <summary>
        /// Gets a value indicating whether or not the body associated with
the gesture detector is currently being tracked
        /// </summary>
        public bool IsTracked
        {
            get
            {
                return this.isTracked;
            }

            private set
            {
                if (this.IsTracked != value)
                {
                    this.isTracked = value;
                    this.NotifyPropertyChanged();
                }
            }
        }

        /// <summary>
        /// Gets a value indicating whether or not the discrete gesture has
been detected
        /// </summary>
        public bool Detected
        {
            get
            {
                return this.detected;
            }

            private set
```

```
            {
                if (this.detected != value)
                {
                    this.detected = value;
                    this.NotifyPropertyChanged();
                }
            }
        }

        /// <summary>
        /// Gets a float value which indicates the detector's confidence that
the gesture is occurring for the associated body
        /// </summary>
        public float Confidence
        {
            get
            {
                return this.confidence;
            }

            private set
            {
                if (this.confidence != value)
                {
                    this.confidence = value;
                    this.NotifyPropertyChanged();
                }
            }
        }

        /// <summary>
        /// Updates the values associated with the discrete gesture detection
result
        /// </summary>
        /// <param name="isBodyTrackingIdValid">True, if the body associated
with the GestureResultView object is still being tracked</param>
        /// <param name="isGestureDetected">True, if the discrete gesture is
currently detected for the associated body</param>
        /// <param name="detectionConfidence">Confidence value for detection
of the discrete gesture</param>
        public void UpdateGestureResult(bool isBodyTrackingIdValid, bool
isGestureDetected, float detectionConfidence)
        {
            this.IsTracked = isBodyTrackingIdValid;
            this.Confidence = 0.0f;

            if (!this.IsTracked)
            {
                this.Detected = false;

            }
            else
            {
                this.Detected = isGestureDetected;

                if (this.Detected)
                {
                    //if (hareket_kontrol==0)
                    //{
```

```
                    //    System.Windows.MessageBoxResult result =
MessageBox.Show(Convert.ToString(hareket_kontrol), "Confirmation",
MessageBoxButton.YesNo, MessageBoxImage.Question);

                    //    hareket_kontrol ++;

                    //}


                }
            }
        }

        /// <summary>
        /// Notifies UI that a property has changed
        /// </summary>
        /// <param name="propertyName">Name of property that has
changed</param>
        private void NotifyPropertyChanged([CallerMemberName] string
propertyName = "")
        {
            if (this.PropertyChanged != null)
            {
                this.PropertyChanged(this, new
PropertyChangedEventArgs(propertyName));
            }
        }
    }
}
```

```
<Window x:Class="WPF_ders_1.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Fatih Kinect"
        xmlns:k="http://schemas.microsoft.com/kinect/2014"
HorizontalAlignment="Center" VerticalAlignment="Top"
WindowStyle="ThreeDBorderWindow" WindowStartupLocation="CenterScreen"
WindowState="Maximized">


    <k:KinectRegion x:Name="kinectRegion">

        <Grid>

            <Frame Name="myFrame" Content="myFrame"
NavigationUIVisibility="Hidden" Grid.ColumnSpan="2"/>

        </Grid>


    </k:KinectRegion>



</Window>
```

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            PageMenu pm = new PageMenu();
            myFrame.NavigationService.Navigate(pm);


        }









    }
}

<Page x:Class="WPF_ders_1.Page1"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"

       Title="Page1"
      xmlns:k="http://schemas.microsoft.com/kinect/2014"
      >
    <Page.Resources>

        <Style  TargetType="Button">
            <Setter Property="Foreground" Value="White"></Setter>
            <Setter Property="Width" Value="250"></Setter>
            <Setter Property="Height" Value="250"></Setter>
            <Setter Property="BorderThickness" Value="0"></Setter>

        </Style>
        <Style x:Name="tBlock"  TargetType="TextBlock">
```

```
            <Setter Property="FontSize" Value="30"></Setter>
            <Setter Property="Width" Value="250"></Setter>
            <Setter Property="Height" Value="250"></Setter>
            <Setter Property="Padding" Value="50"></Setter>

        </Style>
    </Page.Resources>


    <Grid >

        <Grid.RowDefinitions>
            <RowDefinition Height="*"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>



            <ScrollViewer Grid.Row="0"
                          HorizontalScrollBarVisibility="Auto"
VerticalScrollBarVisibility="Auto"
                          k:KinectRegion.IsHorizontalRailEnabled="true"
k:KinectRegion.IsVerticalRailEnabled="true"
                          k:KinectRegion.ZoomMode="Enabled">
            <Image x:Name="image_view_1"  Source="img/amazing-dusk-
background-1920x1200.jpg" Stretch="Fill" Width="1366" Height="700"
x:FieldModifier="public">
            </Image>
        </ScrollViewer>

            <Button x:Name="previous_button" Click="previous_button_Click"
HorizontalAlignment="Left" >
                <Button.Background>
                    <ImageBrush ImageSource="img/button_previous.png"/>
                </Button.Background>
            </Button>
            <Button x:Name="next_button" Click="next_button_Click"
HorizontalAlignment="Right" >
                <Button.Background>
                    <ImageBrush ImageSource="img/button_next.png"/>
                </Button.Background>
            </Button>



        <Button Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="200" Click="Button_Click" Height="150"
Canvas.Left="10" Canvas.Top="20" Background="Transparent" FontSize="40"
FontFamily="Verdana" Opacity="0.8" />
        <Image Name="image" HorizontalAlignment="Center"
VerticalAlignment="Top" Width="200"  Height="150" Canvas.Left="10"
Canvas.Top="20" Opacity="0.8" />
    </Grid>

</Page>

using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Media.Animation;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Kinect;
using Microsoft.Kinect.Toolkit;
using System.IO;
using System.ComponentModel;


namespace WPF_ders_1
{


    /// <summary>
    /// Interaction logic for Page1.xaml
    /// </summary>
    public partial class Page1 : Page
    {
       //string[] imgs = new string[] { "img/amazing-dusk-background-
1920x1200.jpg", "img/amazing-high-resolution-images-1920x1080.jpg",
"img/amazing-sunset-photos-1920x1080.jpg" };
       public int selected ;
        List<String> image_names = new List<String>();
        private KinectSensor kinectSensor = null;

        private List<GestureDetector> gestureDetectorList = null;
        private Body[] bodies = null;
        private Boolean hareket_kontrol =false;
        private BodyFrameReader bodyFrameReader = null;

        public Page1(int val):this()
        {
            InitializeComponent();
            selected = val;
            this.kinectSensor = KinectSensor.GetDefault();
            kinectSensor.Open();
            this.bodyFrameReader =
this.kinectSensor.BodyFrameSource.OpenReader();
            this.bodyFrameReader.FrameArrived+=bodyFrameReader_FrameArrived;
            this.gestureDetectorList = new List<GestureDetector>();
            int maxBodies = this.kinectSensor.BodyFrameSource.BodyCount;
            for (int i = 0; i < maxBodies; ++i)
            {
                GestureResultView result = new GestureResultView(i, false,
false, 0.0f);
                GestureDetector detector = new
GestureDetector(this.kinectSensor, result);
                this.gestureDetectorList.Add(detector);
```

```
            }
            DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\img\");
            FileInfo[] images = folder.GetFiles("*.jpg");
            foreach (FileInfo img in images)
            {
                this.image_names.Add("img/"+img.Name);
            }
            showImage(image_names[selected], selected);




        }

        private void bodyFrameReader_FrameArrived(object sender,
BodyFrameArrivedEventArgs e)
        {
            bool dataReceived = false;

            using (BodyFrame bodyFrame = e.FrameReference.AcquireFrame())
            {
                if (bodyFrame != null)
                {
                    if (this.bodies == null)
                    {

                        // creates an array of 6 bodies, which is the max
number of bodies that Kinect can track simultaneously
                        this.bodies = new Body[bodyFrame.BodyCount];
                    }

                    // The first time GetAndRefreshBodyData is called, Kinect
will allocate each Body in the array.
                    // As long as those body objects are not disposed and not
set to null in the array,
                    // those body objects will be re-used.
                    bodyFrame.GetAndRefreshBodyData(this.bodies);

                    dataReceived = true;
                }
            }

            if (dataReceived)
            {
                // visualize the new body data


                // we may have lost/acquired bodies, so update the
corresponding gesture detectors
                if (this.bodies != null)
                {



                    // loop through all bodies to see if any of the gesture
detectors need to be updated
```

```
                    int maxBodies =
this.kinectSensor.BodyFrameSource.BodyCount;
                    for (int i = 0; i < maxBodies; ++i)
                    {
                        Body body = this.bodies[i];
                        ulong trackingId = body.TrackingId;


                        // if the current body TrackingId changed, update the
corresponding gesture detector with the new value
                        if (trackingId !=
this.gestureDetectorList[i].TrackingId)
                        {
                            this.gestureDetectorList[i].TrackingId =
trackingId;

                            // if the current body is tracked, unpause its
detector to get VisualGestureBuilderFrameArrived events
                            // if the current body is not tracked, pause its
detector so we don't waste resources trying to get invalid gesture results
                            this.gestureDetectorList[i].IsPaused = trackingId
== 0;
                        }
                    }
                }
            }
        }

        public event PropertyChangedEventHandler PropertyChanged;

        public Page1()
        {
            // TODO: Complete member initialization
        }



        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu p2 = new PageMenu();
            this.NavigationService.Navigate(p2);

        }

        private void next_button_Click(object sender, RoutedEventArgs e)
        {
            if (selected == 0)
            {
                selected = image_names.Count - 1;
                showImage(image_names[selected], selected);
            }
            else
            {
                selected = selected - 1; showImage(image_names[selected],
selected);
            }
        }

        public void previous_button_Click(object sender, RoutedEventArgs e)
```

```
        {
            if (selected == image_names.Count - 1)
            {
                selected = 0;
                showImage(image_names[selected], selected);
            }
            else
            {
                selected = selected + 1; showImage(image_names[selected],
selected);
            }
        }
        public void showImage(string img, int i)
        {
            BitmapImage b1 = new BitmapImage();
            b1.BeginInit();
            b1.UriSource = new Uri(img, UriKind.Relative);
            b1.EndInit();
            TimeSpan ts = TimeSpan.FromSeconds(0.5);
            var fadeInAnimation = new DoubleAnimation(1d,ts);

            if (image_view_1.Source != null)
            {
                var fadeOutAnimation = new DoubleAnimation(0d, ts);

                fadeOutAnimation.Completed += (o, e) =>
                {
                    image_view_1.Source = b1;
                    image_view_1.BeginAnimation(Image.OpacityProperty,
fadeInAnimation);
                };

                image_view_1.BeginAnimation(Image.OpacityProperty,
fadeOutAnimation);
            }
            else
            {
                image_view_1.Opacity = 0d;
                image_view_1.Source = b1;
                image_view_1.BeginAnimation(Image.OpacityProperty,
fadeInAnimation);
            }



        }
        private void RegisterGesture(BodyFrame bodyFrame)
        {
            bool dataReceived = false;
            Body[] bodies = null;

            if (bodyFrame != null)
            {
                if (bodies == null)
                {
                    // Creates an array of 6 bodies, which is the max
                    // number of bodies the Kinect can track simultaneously
                    bodies = new Body[bodyFrame.BodyCount];
                }
```

```
                // The first time GetAndRefreshBodyData is called,
                // allocate each Body in the array.
                // As long as those body objects are not disposed and
                // not set to null in the array,
                // those body objects will be re-used.
                bodyFrame.GetAndRefreshBodyData(bodies);
                dataReceived = true;
            }

            if (dataReceived)
            {
                // We may have lost/acquired bodies,
                // so update the corresponding gesture detectors
                if (bodies != null)
                {
                    // Loop through all bodies to see if any
                    // of the gesture detectors need to be updated
                    for (int i = 0; i < bodyFrame.BodyCount; ++i)
                    {
                        Body body = bodies[i];
                        ulong trackingId = body.TrackingId;

                        // If the current body TrackingId changed,
                        // update the corresponding gesture detector with
                        // the new value
                        if (trackingId !=
                          this.gestureDetectorList[i].TrackingId)
                        {
                            this.gestureDetectorList[i].TrackingId =
trackingId;

                            // If the current body is tracked, unpause its
                            // detector to get
                            // VisualGestureBuilderFrameArrived events
                            // If the current body is NOT tracked, pause its
                            // detector so we don't waste resources trying to
get
                            // invalid gesture results
                            this.gestureDetectorList[i].IsPaused =
                                trackingId == 0;
                        }
                    }
                }
            }
        }
        public void showMesage(string ms)
        {
            System.Windows.MessageBoxResult result = MessageBox.Show(ms,
"Confirmation", MessageBoxButton.YesNo, MessageBoxImage.Question);
            if (result == System.Windows.MessageBoxResult.Yes)
            {
                Application.Current.Shutdown();
            }
        }
    }

}
```

```
<Page x:Class="WPF_ders_1.Page2"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      xmlns:k="http://schemas.microsoft.com/kinect/2014"
       Title="Page2">




    <Grid>
        <MediaElement Name="MediaElement1"   LoadedBehavior="Manual"
MediaOpened="MediaElement1_MediaOpened" Width="1366" Height="720"  />



            <Slider Name="slider_timer" Margin="0,0,89,0"
HorizontalAlignment="Center" Width="1200"
ValueChanged="slider_timer_ValueChanged"  />
            <StackPanel Orientation="Horizontal" HorizontalAlignment="Center"
VerticalAlignment="Bottom" >
                <Button Name="button_open"    Width="130" Height="130"
Click="Button_Click_3" Opacity="50" Margin="10" BorderBrush="{x:Null}">
                    <Button.Background>
                        <ImageBrush ImageSource="img/dosya-icon.png"/>
                    </Button.Background>
                </Button>
                <Button Name="button_play"  Width="150" Height="150"
Click="Button_Click_1" Margin="10" BorderBrush="{x:Null}">
                    <Button.Background>
                        <ImageBrush ImageSource="img/play-icon.png"/>
                    </Button.Background>
                </Button>
                <Button Name="button_pause"   Width="150" Height="150"
Click="Button_Click_2" Opacity="50" Margin="10" BorderBrush="{x:Null}">
                    <Button.Background>
                        <ImageBrush ImageSource="img/pause-icon.png"/>
                    </Button.Background>
                </Button>
                <Slider x:Name="slider_volume" HorizontalAlignment="Right"
Width="143" ValueChanged="slider_volume_ValueChanged" Margin="10"
Height="118"   />
            </StackPanel>

        <Button Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="200" Click="Button_Click" Height="150"
Canvas.Left="0" Canvas.Top="0" Background="Transparent" FontSize="40"
FontFamily="Verdana" Opacity="0.8" BorderBrush="{x:Null}" />
    </Grid>


</Page>

using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Forms;
using System.Speech.Recognition;
using Microsoft.Kinect;
namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for Page2.xaml
    /// </summary>
    public partial class Page2 : Page
    {
        SpeechRecognitionEngine recEngine = new SpeechRecognitionEngine();
        private KinectSensor kinectSensor = null;

        private List<GestureDetector> gestureDetectorList = null;
        private Body[] bodies = null;
        private Boolean hareket_kontrol = false;
        private Boolean bereket_kontrol = false;
        private BodyFrameReader bodyFrameReader = null;
        string video_name = "";
        public Page2(string val):this()
        {
            InitializeComponent();
            video_name = val;
            this.MediaElement1.Source = new
Uri(@"C:\Users\salihgunduz\Documents\Visual Studio 2013\Projects\WPF ders
1\WPF ders 1\bin\x64\Debug\video\" + video_name);
            System.Windows.Threading.DispatcherTimer dispatcherTimer = new
System.Windows.Threading.DispatcherTimer();
            dispatcherTimer.Tick += new EventHandler(timer_Tick);
            dispatcherTimer.Interval = new TimeSpan(0, 0, 1);
            dispatcherTimer.Start();
            MediaElement1.Play();
            Choices komutlar=new Choices();
            komutlar.Add(new string[] { "play", "pause", "forward","volume
up","volume down","previous","mute" });
            GrammarBuilder gBuilder = new GrammarBuilder();
            gBuilder.Append(komutlar);
            Grammar grammar = new Grammar(gBuilder);

            recEngine.LoadGrammarAsync(grammar);
            recEngine.SetInputToDefaultAudioDevice();
            recEngine.RecognizeAsync(RecognizeMode.Multiple);
            recEngine.SpeechRecognized += recEngine_SpeechRecognized;
            this.kinectSensor = KinectSensor.GetDefault();
            kinectSensor.Open();
            this.bodyFrameReader =
this.kinectSensor.BodyFrameSource.OpenReader();
```

```
            this.bodyFrameReader.FrameArrived +=
bodyFrameReader_FrameArrived;
            this.gestureDetectorList = new List<GestureDetector>();
            int maxBodies = this.kinectSensor.BodyFrameSource.BodyCount;
            for (int i = 0; i < maxBodies; ++i)
            {
                GestureResultView result = new GestureResultView(i, false,
false, 0.0f);
                GestureDetector detector = new
GestureDetector(this.kinectSensor, result);
                this.gestureDetectorList.Add(detector);


            }
        }

        private void bodyFrameReader_FrameArrived(object sender,
BodyFrameArrivedEventArgs e)
        {
            bool dataReceived = false;

            using (BodyFrame bodyFrame = e.FrameReference.AcquireFrame())
            {

                if (bodyFrame != null)
                {
                    if (this.bodies == null)
                    {
                        // creates an array of 6 bodies, which is the max
number of bodies that Kinect can track simultaneously
                        this.bodies = new Body[bodyFrame.BodyCount];
                    }

                    // The first time GetAndRefreshBodyData is called, Kinect
will allocate each Body in the array.
                    // As long as those body objects are not disposed and not
set to null in the array,
                    // those body objects will be re-used.
                    bodyFrame.GetAndRefreshBodyData(this.bodies);
                    dataReceived = true;
                }
            }

            if (dataReceived)
            {
                // visualize the new body data


                // we may have lost/acquired bodies, so update the
corresponding gesture detectors
                if (this.bodies != null)
                {



                    // loop through all bodies to see if any of the gesture
detectors need to be updated
```

```
                    int maxBodies =
this.kinectSensor.BodyFrameSource.BodyCount;
                    for (int i = 0; i < maxBodies; ++i)
                    {
                        Body body = this.bodies[i];
                        ulong trackingId = body.TrackingId;
                        GestureDetector detector = new
GestureDetector(this.kinectSensor, gestureDetectorList[i].GestureResultView);

                        //if
(gestureDetectorList[i].GestureResultView.Detected == true)
                        //{
                        //    if (hareket_kontrol == false)
                        //    {
                        //        TimeSpan ts = TimeSpan.FromSeconds(100);
                        //        MediaElement1.Position =
MediaElement1.Position+ts;

                        //        hareket_kontrol = true;

                        //    }

                        //}
                        //else hareket_kontrol = false;

                        // if the current body TrackingId changed, update the
corresponding gesture detector with the new value
                        if (trackingId !=
this.gestureDetectorList[i].TrackingId)
                        {
                            this.gestureDetectorList[i].TrackingId =
trackingId;

                            // if the current body is tracked, unpause its
detector to get VisualGestureBuilderFrameArrived events
                            // if the current body is not tracked, pause its
detector so we don't waste resources trying to get invalid gesture results
                            this.gestureDetectorList[i].IsPaused = trackingId
== 0;
                        }
                    }
                }
            }
        }

        private void recEngine_SpeechRecognized(object sender,
SpeechRecognizedEventArgs e)
        {
            switch (e.Result.Text)
            {
                case "play":
                    MediaElement1.Play();
                    break;
                case "pause":
                    MediaElement1.Pause();
                    break;
                case "forward":
                    TimeSpan ts = TimeSpan.FromSeconds(50);
                    MediaElement1
```

```
.Position = MediaElement1.Position+ts;
                    break;
                case "previous":
                    TimeSpan ts1 = TimeSpan.FromSeconds(50);
                    MediaElement1.Position = MediaElement1.Position - ts1;
                    break;
                case "volume up":
                    MediaElement1.Volume = MediaElement1.Volume+5;
                    break;
                case "volume down":
                    MediaElement1.Volume = MediaElement1.Volume - 5;
                    break;
                case "mute":
                    MediaElement1.Volume = 0;
                    break;
            }
        }

        public Page2()
        {
            // TODO: Complete member initialization
        }

        private void timer_Tick(object sender, EventArgs e)
        {
            slider_timer.Value = MediaElement1.Position.TotalSeconds;
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu p1 = new PageMenu();
            this.NavigationService.Navigate(p1);

        }

        private void Button_Click_1(object sender, RoutedEventArgs e)
        {



            MediaElement1.Play();


        }

        private void Button_Click_2(object sender, RoutedEventArgs e)
        {
            MediaElement1.Pause();
        }

        private void slider_timer_ValueChanged(object sender,
RoutedPropertyChangedEventArgs<double> e)
        {
            TimeSpan ts = TimeSpan.FromSeconds(e.NewValue);
            MediaElement1.Position = ts;
        }

        private void slider_volume_ValueChanged(object sender,
RoutedPropertyChangedEventArgs<double> e)
```

```
        {
            MediaElement1.Volume = slider_volume.Value;
        }

        private void MediaElement1_MediaOpened(object sender, RoutedEventArgs
e)
        {
            if (MediaElement1.NaturalDuration.HasTimeSpan)
            {
                TimeSpan ts =
TimeSpan.FromMilliseconds(MediaElement1.NaturalDuration.TimeSpan.TotalMillise
conds);
                slider_timer.Maximum = ts.TotalSeconds;
            }
        }

        private void Button_Click_3(object sender, RoutedEventArgs e)
        {
           /* OpenFileDialog fileDialog = new OpenFileDialog();
             fileDialog.Filter = "Media files
(*.mp3;*.mpg;*.mpeg)|*.mp3;*.mpg;*.mpeg|All files (*.*)|*.*";
                        fileDialog.ShowDialog();
                             //  this.MediaElement1.Source = new
Uri(fileDialog.FileName);
            */
            PageVideoAra p1 = new PageVideoAra();
            this.NavigationService.Navigate(p1);

        }

        public void showMesage(string ms)
        {
            System.Windows.MessageBoxResult result =
System.Windows.MessageBox.Show(ms, "Confirmation", MessageBoxButton.YesNo,
MessageBoxImage.Question);
            if (result == System.Windows.MessageBoxResult.Yes)
            {
                System.Windows.Application.Current.Shutdown();
            }
        }


    }
}
```

```
<Page x:Class="WPF_ders_1.PageImageAra"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="300"
      xmlns:k="http://schemas.microsoft.com/kinect/2014"
       Title="PageImageAra">
    <Page.Resources>
        <DataTemplate x:Key="ItemTemplate">
            <StackPanel Orientation="Vertical">
                <Image Width="300" Height="200" Stretch="Fill"
Source="{Binding}"/>
```

```
            </StackPanel>
        </DataTemplate>
    </Page.Resources>

        <Grid x:Name="movie_grid">
            <Grid.RowDefinitions>
                <RowDefinition Height="Auto"/>
                <RowDefinition />
            </Grid.RowDefinitions>

            <Button Grid.Row="0" Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="150" Height="50" Background="Transparent"
FontSize="40" FontFamily="Verdana" Opacity="0.8" BorderBrush="{x:Null}"
Click="Button_Click" />
            <ListView Grid.Row="1"
SelectionChanged="Thumbnails_SelectionChanged"
                      k:KinectRegion.IsHorizontalRailEnabled="true"
k:KinectRegion.IsVerticalRailEnabled="true"
                      k:KinectRegion.ZoomMode="Enabled"
              Name="Thumbnails"
              ItemTemplate="{StaticResource ItemTemplate}"
              ItemsSource="{Binding Path = movie_posters_list}">
                <ListView.ItemsPanel>
                    <ItemsPanelTemplate>
                        <UniformGrid Columns="10" />
                    </ItemsPanelTemplate>
                </ListView.ItemsPanel>

            </ListView>




        </Grid>



</Page>
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WPF_ders_1
{
    /// <summary>
```

```
    /// Interaction logic for PageImageAra.xaml
    /// </summary>
    public partial class PageImageAra : Page
    {
        public PageImageAra()
        {
            InitializeComponent();
            Loaded+=PageImageAra_Loaded;

        }

        List<String> image_names = new List<String>();

        private void PageImageAra_Loaded(object sender, RoutedEventArgs e)
        {
            DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\img\");
            FileInfo[] images = folder.GetFiles("*.jpg");
            foreach (FileInfo img in images)
            {

                Thumbnails.Items.Add(img.FullName);
                this.image_names.Add(img.FullName);

            }
        }

        private void Thumbnails_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {
            int selected_id = Thumbnails.SelectedIndex;
            Page1 pMenu = new Page1(selected_id);
            NavigationService.Navigate(pMenu);
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu pMenu = new PageMenu();
            this.NavigationService.Navigate(pMenu);
        }
    }
}

<Page x:Class="WPF_ders_1.PageMenu"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"

       Title="PageMenu"
      xmlns:k="http://schemas.microsoft.com/kinect/2014"   >

    <Page.Resources>

        <Style  TargetType="Button">
```

```
            <Setter Property="Background" Value="Transparent"></Setter>
            <Setter Property="Foreground" Value="White"></Setter>
            <Setter Property="BorderBrush" Value="White"></Setter>
            <Setter Property="BorderThickness" Value="1"></Setter>
            <Setter Property="Width" Value="250"></Setter>
            <Setter Property="Height" Value="250"></Setter>

            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Border x:Name="focusRect" BorderThickness="6"
Margin="-6">
                            <ContentPresenter />
                            <Border.BorderBrush>
                                <SolidColorBrush x:Name="focusRectBrush"
Color="Tomato" />
                            </Border.BorderBrush>
                            <VisualStateManager.VisualStateGroups>
                                <VisualStateGroup x:Name="CommonStates">
                                    <VisualState x:Name="Normal"/>
                                    <VisualState x:Name="MouseOver">
                                        <Storyboard>
                                            <ColorAnimation Duration="0"
Storyboard.TargetName="focusRectBrush" Storyboard.TargetProperty="Color"
To="White"/>
                                        </Storyboard>
                                    </VisualState>
                                    <VisualState x:Name="Pressed">
                                        <Storyboard>
                                            <ColorAnimation Duration="0"
Storyboard.TargetName="focusRectBrush" Storyboard.TargetProperty="Color"
To="Gray"/>
                                        </Storyboard>
                                    </VisualState>
                                </VisualStateGroup>
                            </VisualStateManager.VisualStateGroups>
                        </Border>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>
        <Style x:Name="tBlock"  TargetType="TextBlock">
            <Setter Property="FontSize" Value="30"></Setter>
            <Setter Property="Width" Value="250"></Setter>
            <Setter Property="Height" Value="250"></Setter>
            <Setter Property="Padding" Value="50"></Setter>

        </Style>
    </Page.Resources>
    <Grid>
        <Grid.Background>
            <ImageBrush Stretch="Fill" ImageSource="img/Manzara-Resmi19.jpg"
AlignmentY="Top" AlignmentX="Center"/>
        </Grid.Background>

        <Grid>
            <StackPanel Orientation="Horizontal"
HorizontalAlignment="Center">
```

```
                <Button Name="button_harita"  Click="button_harita_Click"
Margin="10">
                    <TextBlock Text="Resimler" HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                </Button>
                <Button Name="button_modul"   Click="button_modul_Click"
Margin="10">
                    <TextBlock Text="Modüller" HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                </Button>
                <Button Name="button_video"  Click="button_video_Click"
Margin="10">
                    <TextBlock Text="Görsel Eğitim"
HorizontalAlignment="Center" VerticalAlignment="Center"/>
                </Button>
                <Button Name="button_sunu"   Click="button_sunu_Click"
Margin="10">
                    <TextBlock Text="Sunular" HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                </Button>

            </StackPanel>

        </Grid>

    </Grid>

</Page>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for PageMenu.xaml
    /// </summary>
    public partial class PageMenu : Page
    {
        public PageMenu()
        {
            InitializeComponent();
        }

        private void button_harita_Click(object sender, RoutedEventArgs e)
        {
            PageImageAra pMenu = new PageImageAra();
```

```
            this.NavigationService.Navigate(pMenu);
        }

        private void button_modul_Click(object sender, RoutedEventArgs e)
        {
            PageModulAra pModul = new PageModulAra();
            this.NavigationService.Navigate(pModul);
        }

        private void button_video_Click(object sender, RoutedEventArgs e)
        {
            PageVideoAra pVideo = new PageVideoAra();
            this.NavigationService.Navigate(pVideo);
        }

        private void button_sunu_Click(object sender, RoutedEventArgs e)
        {
            PageSunuAra pSunu = new PageSunuAra();
            this.NavigationService.Navigate(pSunu);
        }

    }
}
```

```
<Page x:Class="WPF_ders_1.PageModul"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="300"
      xmlns:k="http://schemas.microsoft.com/kinect/2014"
       Title="PageImageAra">
    <Page.Resources>
        <DataTemplate x:Key="ItemTemplate">
            <StackPanel Orientation="Vertical">
                <Image Width="800" Height="1000" Stretch="Fill"
Source="{Binding}"/>

            </StackPanel>
        </DataTemplate>
    </Page.Resources>

    <Grid x:Name="movie_grid">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition />
        </Grid.RowDefinitions>

        <Button Grid.Row="0" Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="150" Height="50" Background="Transparent"
FontSize="40" FontFamily="Verdana" Opacity="0.8" BorderBrush="{x:Null}"
Click="Button_Click" />
        <ListView Grid.Row="1"
                      k:KinectRegion.IsHorizontalRailEnabled="true"
k:KinectRegion.IsVerticalRailEnabled="true"
                      k:KinectRegion.ZoomMode="Enabled"
HorizontalAlignment="Center"
              Name="Thumbnails"
```

```
                ItemTemplate="{StaticResource ItemTemplate}"
                ItemsSource="{Binding Path = movie_posters_list}">
            <ListView.ItemsPanel >
                <ItemsPanelTemplate>
                    <UniformGrid Columns="1" />
                </ItemsPanelTemplate>
            </ListView.ItemsPanel>

        </ListView>



    </Grid>



</Page>
```

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for PageModul.xaml
    /// </summary>
    public partial class PageModul : Page
    {
        string selected;
        public PageModul(string val):this()
        {
            InitializeComponent();
            Loaded+=PageModul_Loaded;
            selected = val;



        }

        public PageModul()
        {
            // TODO: Complete member initialization
        }
```

```
        private void PageModul_Loaded(object sender, RoutedEventArgs e)
        {
            DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\pdf_images\"+selected);
            FileInfo[] images = folder.GetFiles("*.jpg");
            foreach (FileInfo img in images)
            {

                Thumbnails.Items.Add(img.FullName);


            }
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu pMenu = new PageMenu();
            this.NavigationService.Navigate(pMenu);
        }




    }
}
```

```
<Page x:Class="WPF_ders_1.PageModulAra"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="300"
       xmlns:k="http://schemas.microsoft.com/kinect/2014"
      Title="PageModulAra">

    <Page.Resources>
        <DataTemplate x:Key="ItemTemplate">
            <StackPanel Orientation="Vertical">
                <Image Width="250" Height="250" Stretch="Fill"
Source="img/pdf-icon.png"/>
                <TextBlock Text="{Binding}" Width="150"/>
            </StackPanel>
        </DataTemplate>
    </Page.Resources>

    <Grid x:Name="movie_grid">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition />
        </Grid.RowDefinitions>
        <Button Grid.Row="0" Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="150" Height="50" Background="Transparent"
FontSize="40" FontFamily="Verdana" Opacity="0.8" BorderBrush="{x:Null}"
Click="Button_Click" />
        <ListView Grid.Row="1" SelectionChanged="Thumbnails_SelectionChanged"
```

```
                    k:KinectRegion.IsHorizontalRailEnabled="true"
k:KinectRegion.IsVerticalRailEnabled="true"
                    k:KinectRegion.ZoomMode="Enabled"
              Name="Thumbnails"
              ItemTemplate="{StaticResource ItemTemplate}"
              ItemsSource="{Binding Path = movie_posters_list}">
            <ListView.ItemsPanel>
                <ItemsPanelTemplate>
                    <UniformGrid Columns="10" />
                </ItemsPanelTemplate>
            </ListView.ItemsPanel>

        </ListView>

    </Grid>

</Page>
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for PageModul.xaml
    /// </summary>
    public partial class PageModul : Page
    {
        string selected;
        public PageModul(string val):this()
        {
            InitializeComponent();
            Loaded+=PageModul_Loaded;
            selected = val;



        }

        public PageModul()
        {
            // TODO: Complete member initialization
        }

        private void PageModul_Loaded(object sender, RoutedEventArgs e)
        {
```

```
            DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\pdf_images\"+selected);
            FileInfo[] images = folder.GetFiles("*.jpg");
            foreach (FileInfo img in images)
            {

                Thumbnails.Items.Add(img.FullName);


            }
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu pMenu = new PageMenu();
            this.NavigationService.Navigate(pMenu);
        }




    }
}
<Page x:Class="WPF_ders_1.PageSunu"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="300"
       xmlns:k="http://schemas.microsoft.com/kinect/2014"
      Title="PageSunu">

    <Grid Background="DarkGray" >

        <Grid.RowDefinitions>
            <RowDefinition Height="*"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>



        <ScrollViewer Grid.Row="0"
                      HorizontalScrollBarVisibility="Auto"
VerticalScrollBarVisibility="Auto"
                      k:KinectRegion.IsHorizontalRailEnabled="true"
k:KinectRegion.IsVerticalRailEnabled="true"
                      k:KinectRegion.ZoomMode="Enabled">
            <Image x:Name="image_view_1" Stretch="None" Width="1366"
Height="700">
            </Image>
        </ScrollViewer>

            <Button x:Name="previous_button"  Width="200" Height="150"
Click="previous_button_Click"  Margin="10" Opacity="0.7"
VerticalAlignment="Center" HorizontalAlignment="left" BorderThickness="0">
            <Button.Background>
```

```
                <ImageBrush ImageSource="img/button_previous.png"/>
            </Button.Background>
        </Button>
            <Button x:Name="next_button"  Width="200" Height="150"
Click="next_button_Click" Margin="10" Opacity="0.7"
VerticalAlignment="Center" HorizontalAlignment="Right" BorderThickness="0">
            <Button.Background>
                <ImageBrush ImageSource="img/button_next.png"/>
            </Button.Background>
        </Button>



        <Button Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="200" Click="Button_Click" Height="150"
Canvas.Left="10" Canvas.Top="20" Background="Transparent" FontSize="40"
FontFamily="Verdana" Opacity="0.8" BorderThickness="0" />

    </Grid>
</Page>
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Office.Interop.PowerPoint;
using Microsoft.Office.Core;
using PowerPoint = Microsoft.Office.Interop.PowerPoint;
using System.Windows.Media.Animation;
namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for PageSunu.xaml
    /// </summary>
    public partial class PageSunu : Page
    {
        string filePath = @"C:\Users\salihgunduz\Documents\visual studio
2013\Projects\WPF ders 1\WPF ders 1\bin\x64\Debug\test.pptx";
        string[] imagePathsArray;
        int selected = 0;
        public PageSunu(string val): this()
        {

            InitializeComponent();
            string selected_sunu = val;
            filePath = @"C:\Users\salihgunduz\Documents\visual studio
2013\Projects\WPF ders 1\WPF ders 1\bin\x64\Debug\sunular\" + selected_sunu;
            pptImageTransform();
            showImage(imagePathsArray[selected], selected);
```

```
        }

        public PageSunu()
        {
            // TODO: Complete member initialization
        }


        private void pptImageTransform()
        {
            Microsoft.Office.Interop.PowerPoint.Application appPpt
     = new Microsoft.Office.Interop.PowerPoint.Application();

            Microsoft.Office.Interop.PowerPoint.Presentation
objActivePresentation
                = appPpt.Presentations.Open(filePath,

Microsoft.Office.Core.MsoTriState.msoCTrue,

Microsoft.Office.Core.MsoTriState.msoTriStateMixed,

Microsoft.Office.Core.MsoTriState.msoFalse);
            int elemanSayısı = objActivePresentation.Slides.Count;
            imagePathsArray = new string[elemanSayısı];
            int i = 0;
            foreach (Microsoft.Office.Interop.PowerPoint.Slide objSlide
                                                                in
objActivePresentation.Slides)
            {
                //Names are generated based on timestamp.

                string
imageName=Convert.ToString(objSlide.SlideID)+"large"+".jpg";
                objSlide.Export(imageName, "PNG", 960, 720);


                imagePathsArray[i] = imageName;
                i++;
            }

        }
        private void showImage(string img, int i)
        {
            BitmapImage b1 = new BitmapImage();
            b1.BeginInit();
            b1.UriSource = new Uri(@"sunular\"+imagePathsArray[i],
UriKind.Relative);
            b1.CacheOption = BitmapCacheOption.OnLoad;
            b1.EndInit();
            TimeSpan ts = TimeSpan.FromSeconds(0.5);
            var fadeInAnimation = new DoubleAnimation(1d, ts);

            if (image_view_1.Source != null)
            {
                var fadeOutAnimation = new DoubleAnimation(0d, ts);

                fadeOutAnimation.Completed += (o, e) =>
```

```
                {
                    image_view_1.Source = b1;
                    image_view_1.BeginAnimation(Image.OpacityProperty,
fadeInAnimation);
                };

                image_view_1.BeginAnimation(Image.OpacityProperty,
fadeOutAnimation);
            }
            else
            {
                image_view_1.Opacity = 0d;
                image_view_1.Source = b1;
                image_view_1.BeginAnimation(Image.OpacityProperty,
fadeInAnimation);
            }



        }

        private void previous_button_Click(object sender, RoutedEventArgs e)
        {
            if (selected == 0)
            {
                selected = imagePathsArray.Length - 1;
                showImage(imagePathsArray[selected], selected);
            }
            else
            {
                selected = selected - 1; showImage(imagePathsArray[selected],
selected);
            }
        }

        private void next_button_Click(object sender, RoutedEventArgs e)
        {
            if (selected == imagePathsArray.Length - 1)
            {
                selected = 0;
                showImage(imagePathsArray[selected], selected);
            }
            else
            {
                selected = selected + 1; showImage(imagePathsArray[selected],
selected);
            }

        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu p2 = new PageMenu();
            this.NavigationService.Navigate(p2);
        }

    }
}
<Page x:Class="WPF_ders_1.PageSunuAra"
```

```
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="300"
      xmlns:k="http://schemas.microsoft.com/kinect/2014"
       Title="PageSunuAra">

    <Page.Resources>
        <DataTemplate x:Key="ItemTemplate">
            <StackPanel Orientation="Vertical">
                <Image Width="150" Height="150" Stretch="Fill"
Source="img/powerpoint-icon.png"/>
                <TextBlock Text="{Binding}"/>
            </StackPanel>
        </DataTemplate>
    </Page.Resources>

    <Grid x:Name="movie_grid">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition />
        </Grid.RowDefinitions>
        <Button Grid.Row="0" Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="150" Height="50" Background="Transparent"
FontSize="40" FontFamily="Verdana" Opacity="0.8" BorderBrush="{x:Null}"
Click="Button_Click" />
        <ListView Grid.Row="1" SelectionChanged="Thumbnails_SelectionChanged"
                      k:KinectRegion.IsHorizontalRailEnabled="true"
k:KinectRegion.IsVerticalRailEnabled="true"
                      k:KinectRegion.ZoomMode="Enabled"
              Name="Thumbnails"
              ItemTemplate="{StaticResource ItemTemplate}"
              ItemsSource="{Binding Path = movie_posters_list}">
            <ListView.ItemsPanel>
                <ItemsPanelTemplate>
                    <UniformGrid Columns="10" />
                </ItemsPanelTemplate>
            </ListView.ItemsPanel>

        </ListView>

    </Grid>


</Page>
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
```

```
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for PageSunuAra.xaml
    /// </summary>
    public partial class PageSunuAra : Page
    {
        public PageSunuAra()
        {
            InitializeComponent();
            Loaded+=PageSunuAra_Loaded;
        }
        List<String> sunu_names = new List<String>();
        private void PageSunuAra_Loaded(object sender, RoutedEventArgs e)
        {
            DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\sunular\");
            FileInfo[] Sunular = folder.GetFiles("*.pptx");
            foreach (FileInfo sunu in Sunular)
            {

                Thumbnails.Items.Add(sunu.Name);
                this.sunu_names.Add(sunu.Name);
            }
        }

        private void Thumbnails_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {
            int selected = Thumbnails.SelectedIndex;
            string selected_sunu = sunu_names[selected];
            PageSunu pSunu = new PageSunu(selected_sunu);
            this.NavigationService.Navigate(pSunu);
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu pMenu = new PageMenu();
            this.NavigationService.Navigate(pMenu);
        }
    }
}
<Page x:Class="WPF_ders_1.PageVideoAra"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="300"
      xmlns:k="http://schemas.microsoft.com/kinect/2014"
      Title="PageSunuAra">

    <Page.Resources>
        <DataTemplate x:Key="ItemTemplate">
```

```
            <StackPanel Orientation="Vertical">
                <Image Width="150" Height="150" Stretch="Fill"
Source="img/play-icon.png"/>
                <TextBlock Text="{Binding}"/>
            </StackPanel>
        </DataTemplate>
    </Page.Resources>

    <Grid x:Name="movie_grid">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition />
        </Grid.RowDefinitions>
        <Button Grid.Row="0" Content="Geri" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="150" Height="50" Background="Transparent"
FontSize="40" FontFamily="Verdana" Opacity="0.8" BorderBrush="{x:Null}"
Click="Button_Click" />
        <ListView Grid.Row="1" SelectionChanged="Thumbnails_SelectionChanged"
                  k:KinectRegion.IsHorizontalRailEnabled="true"
k:KinectRegion.IsVerticalRailEnabled="true"
                  k:KinectRegion.ZoomMode="Enabled"
              Name="Thumbnails"
              ItemTemplate="{StaticResource ItemTemplate}"
              ItemsSource="{Binding Path = movie_posters_list}">
            <ListView.ItemsPanel>
                <ItemsPanelTemplate>
                    <UniformGrid Columns="10" />
                </ItemsPanelTemplate>
            </ListView.ItemsPanel>

        </ListView>

    </Grid>


</Page>
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WPF_ders_1
{
    /// <summary>
    /// Interaction logic for PageVideoAra.xaml
    /// </summary>
    public partial class PageVideoAra : Page
    {
```

```
        public PageVideoAra()
        {
            InitializeComponent();
            Loaded+=PageVideoAra_Loaded;
        }
        List<String> video_names = new List<String>();
        private void PageVideoAra_Loaded(object sender, RoutedEventArgs e)
        {
            DirectoryInfo folder = new
DirectoryInfo(System.IO.Path.GetDirectoryName(System.Diagnostics.Process.GetC
urrentProcess().MainModule.FileName) + @"\video\");
            FileInfo[] Sunular = folder.GetFiles("*.*");
            foreach (FileInfo sunu in Sunular)
            {

                Thumbnails.Items.Add(sunu.Name);
                this.video_names.Add(sunu.Name);
            }
        }

        private void Thumbnails_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {
            int selected = Thumbnails.SelectedIndex;
            string selected_sunu = video_names[selected];
            Page2 pSunu = new Page2(selected_sunu);
            this.NavigationService.Navigate(pSunu);
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            PageMenu pMenu = new PageMenu();
            this.NavigationService.Navigate(pMenu);
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Salih GÜNDÜZ
Uyruğu : TC.
Doğum Yeri ve Tarihi : NİZİP/1985
Telefon : 05547298106
Faks :
e-mail : Gunduz.salih@gmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise :	**İçel Anadolu Lisesi, Mersin**	**2004**
Üniversite :	**Şelçuk Üniversitesi Teknik Eğitim Fakültesi**	**2008**
Yüksek Lisans :	**Selçuk Üniversitesi Fen bilimleri Enstitüsü**	**-**
Doktora :		

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2008-2016	**Çalış Ortaokulu**	**Öğretmen**
20016-2017	**Şarköy Mesleki ve Teknik Eğitim Merkezi**	**Öğretmen**

UZMANLIK ALANI Görüntü işleme, Mobil uygulamalar, Yapay sinir ağları, Kinect, Eğitim.

YABANCI DİLLER İngilizce(Çok iyi).

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR

Cilt Hastalığı Görüntülerinin Mobil Cihazlar Vasıtasıyla Ön İşlemlerden Geçirilmesi

The Effects Of Using Kinect Sensors In Education On Class Management