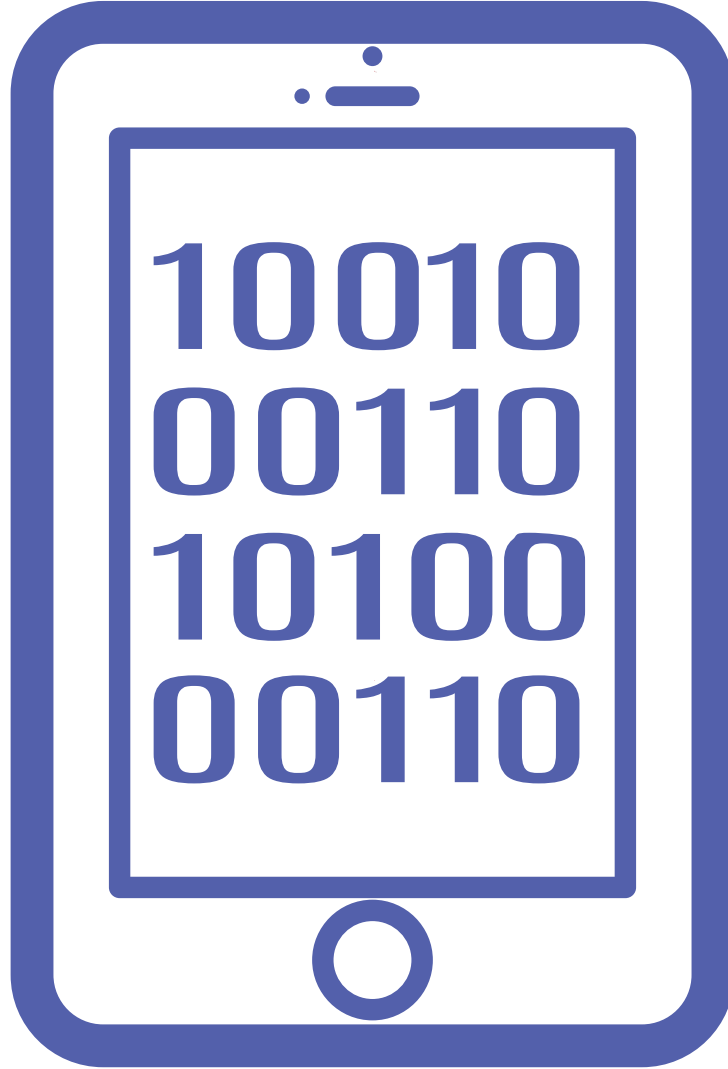




MOBİL UYGULAMA

ORTAOKUL



T.C. SANAYİ VE
TEKNOLOJİ BAKANLIĞI



T.C. GENÇLİK VE
SPOR BAKANLIĞI



TÜBİTAK



TÜRKİYE
TEKNOLOJİ
TAKIMI

TÜBİTAK Deneyap Kitapları 16

MOBİL UYGULAMA

ORTAOKUL

Çağdaş MOLLAOĞLU

© Türkiye Bilimsel ve Teknolojik Araştırma Kurumu, 2023

Bu kitabın bütün hakları saklıdır.
Yazılar ve görsel materyaller, TÜBİTAK'tan yazılı izin alınmadan
tümüyle veya kısmen çoğaltılamaz ve yayımlanamaz.
Kitabın PDF formatındaki elektronik nüshasına
<https://yayinlar.tubitak.gov.tr/deneyap-atolyesi> adresinden ulaşılabilir.
TÜBİTAK Deneyap Kitapları DENEYAP TÜRKİYE Projesi kapsamında hazırlanmıştır.

ISBN: 978-605-312-530-3
Yayıncı Sertifika No: 47703

Yayın Tarihi: 2023

TÜBİTAK Başkanı: Prof. Dr. Hasan MANDAL
Bilim ve Toplum Başkanı: Ömer KÖKÇAM
Genel Yayın Yönetmeni: Fatma BAŞAR
Editörler: Havva Hilal KAÇAR, Doç. Dr. Şahin İDİL
Düzeltili: Dr. Mustafa ORHAN
Telif İşleri Sorumlusu: Havva Hilal KAÇAR

TÜBİTAK Bilim ve Toplum Başkanlığı
Tunus Caddesi No: 80 Kavaklıdere 06680 Ankara
Tel: (312) 298 96 50
e-posta: deneyap@tubitak.gov.tr
<https://yayinlar.tubitak.gov.tr/deneyap-atolyesi>

DENEYAP
Teknoloji Atölyeleri

MOBİL UYGULAMA

ORTAOKUL

Çağdaş MOLLAOĞLU



İçindekiler

1. Hafta: Mobil Uygulama Geliştirme ve App Inventor.....	4
Ön Bilgi:	5
Haftanın Kazanımları:	5
Haftanın Amacı:	5
Kullanılacak Malzemeler:	5
Haftanın İşlenişi:	5
1. Gözle	6
1.1. Mobil Cihaz ve Mobil Uygulama	6
1.2. App Inventor	9
2. Uygulama	12
3. Tasarla.....	12
4. Üret.....	12
5. Değerlendir	12
2. Hafta: App Inventor Arayüzü ve Uygulama Geliştirme.....	13
Ön Bilgi:	13
Haftanın Kazanımları:	13
Haftanın Amacı:	13
Kullanılacak Malzemeler:	13
Haftanın İşlenişi:	14
1. Gözle	15
1.1. Tasarım sekmesi nasıl kullanılır?	15
1.2. Bloklar sekmesi nasıl kullanılır?	27
2. Uygula.....	28
3. Tasarla.....	28
4. Üret.....	28
5. Değerlendir	29
3. Hafta: App Inventor Medya, Döngü, Koşul ve Hazır Modüller Kullanımı	30
Ön Bilgi:	30
Haftanın Kazanımları:	30
Haftanın Amacı:	30
Kullanılacak Malzemeler:	30
Haftanın İşlenişi:	31

1. Gözle	32
1.1. App Inventor Medya Kullanımı.....	32
1.2. Koşul, Döngü ve İç içe Döngüler	42
1.3. Metin Okuma	45
1.4. Çevrimiçi Çeviri.....	47
2. Uygulama	47
3. Tasarla.....	48
4. Üret.....	48
5. Değerlendir.....	48
4. Hafta: Sensörler ve Haberleşme	49
Ön Bilgi:	49
Haftanın Kazanımları:	49
Haftanın Amacı:	49
Kullanılacak Malzemeler:	49
Haftanın İşlenişi:	50
1. Gözle ve Uygula	51
1.1. Sensörler	51
1.2. App Inventor ile Kablosuz Haberleşme	58
2. Tasarla.....	67
3. Üret.....	67
4. Değerlendir.....	67
5. Hafta: Projenin Derlenmesi ve Google Play’de Yayınlama	68
Ön Bilgi:	68
Haftanın Kazanımları:	68
Haftanın Amacı:	68
Kullanılacak Malzemeler:	68
Haftanın İşlenişi:	68
1. Gözle ve Uygula	69
1.1. Test Etme ve Derleme	69
1.2. Google Play Uygulama Mağazası	72
2. Tasarla.....	73
3. Üret.....	73
4. Değerlendir.....	74

6. Hafta: Kod Bloklarını Tekrar Kullanma ve Yayınlama - Proje Sunumları	75
Ön Bilgi:	75
Haftanın Kazanımları:	75
Haftanın Amacı:	75
Kullanılacak Malzemeler:	75
Haftanın İşlenişi:	76
1. Gözle ve Uygula	77
1.1. Kod Bloklarını Tekrar Kullanmak	77
1.2. App Inventor Proje Galerisi	80
2. Tasarla.....	84
3. Üret.....	84
4. Değerlendir.....	84

Sunuş

Bu kitap, özellikle Deneyap atölyelerine katılan ortaokul öğrencileri olmak üzere, mobil uygulama geliştirme yolculuğuna yeni başlayan herkes için tasarlanmış bir rehberdir.

Bu kitap, mobil uygulama geliştirmenin temel kavramlarına odaklanarak başlıyor ve MIT App Inventor ile adım adım ilerlemenize yardımcı oluyor. Her bölüm, mobil uygulama geliştirme süreçlerini anlatmanın yanı sıra, MIT App Inventor'ın avantajlarını ve sınırlarını da ortaya koyuyor. Kitap, sadece yeni başlayanlar için değil, aynı zamanda bu alandaki deneyimini geliştirmek isteyenler için de rehberlik edecek bir kaynak olarak tasarlandı.

6 haftalık bir eğitim programını desteklemek amacıyla hazırlanan kitapta aşağıdaki konu başlıklarına değindik:

- Mobil Uygulama Geliştirme ve App Inventor
- App Inventor Arayüzü ve Uygulama Geliştirme
- App Inventor Medya, Döngü, Koşul ve Hazır Modüller Kullanımı
- Sensörler ve Haberleşme
- Projenin Derlenmesi ve Google Play'de Yayınlama
- Kod Bloklarını Tekrar Kullanma ve Yayınlama - Proje Sunumları

Öğretim tasarımında Deneyap atölyelerinin diğer derslerinde de kullanılan Gözle, Uygula, Tasarla, Üret ve Değerlendir (GÜTUD) döngüsünü temel aldık. Bu döngüde Gözle basamağında temel bilgi ve uygulamaların öğrencilerle incelenmesinin ardından, Uygula basamağında çözümlü örnekler üzerinde çalışacağız. Her üniteye özgün çözümler üretmenizi beklediğimiz problemlerle Tasarla, Üret ve Değerlendir basamaklarını işleteceğiz. Bu problemler derste anlatılanların ötesinde bilgiler edinmenizi ve bağlantılar keşfetmenizi sağlayacak. Bölüm sonlarındaki İlave Etkinliklerle öğrendiklerinizi yeni durumlara transfer etmenizi bekliyoruz. GÜTUD temel bilgilerin sunulması yanında, bu bilgilerin gerçek problemlere uygulanmasını da içerdiğinden derslerin verimini artıracaktır.

Mobil uygulama geliştirmeye olan ilginizi ve tutkunuzu daha da derinleştirmenizi umuyoruz. Bu kitabı incelerken keyif almanızı ve kendi mobil uygulama projelerinizi hayata geçirmenizi diliyoruz. İyi okumalar ve başarılar dileriz!

Tüm paydaşlarımıza faydalı olması dileğiyle.

Çağdaş MOLLAOĞLU

İzmir,2023

1. Hafta: Mobil Uygulama Geliştirme ve App Inventor

Ön Bilgi:

- Temel bilgisayar bilgisi

Haftanın Kazanımları:

- Mobil cihaz tanımlar.
- Hangi cihazın mobil cihaz olarak isimlendirileceğini kavrar.
- Mobil işletim sistemini tanımlar ve örnekler verebilir.
- Mobil uygulama marketini tanımlar ve örnekler verebilir.
- Mobil uygulama örnekleri verebilir.
- Mobil uygulama geliştirme süreci hakkında bilgi sahibi olur.
- App Inventor hakkında bilgi sahibi olur.
- App Inventor ara yüzü hakkında bilgi sahibi olur.
- App Inventor'da AI Companion ile uygulama test eder.
- App Inventor'da Android Emülatör ile uygulama test eder.
- App Inventor'da USB bağlantısı ile uygulama test etme hakkında bilgi sahibi olur.

Haftanın Amacı:

İlk hafta öğrencilerin mobil cihazları tanıması, mobil işletim sistemlerini bilmeleri, mobil uygulama örnekleri görmeleri, mobil uygulama geliştirme süreci hakkında bilgi sahibi olmaları ve App Inventor ile tanışmaları amaçlanmaktadır. App Inventor ara yüzü hakkında bilgi sahibi olduktan sonra geliştirilen uygulamaların cihazlarda nasıl çalışacaklarını görebilme ve test edebilme konusunda örneklerle tanışması hedeflenmektedir.

Kullanılacak Malzemeler:

Bilgisayar, Mobil cihaz (cep telefonu ya da tablet), kalem, kâğıt

Haftanın İşlenişi:

Gözele: Mobil cihaz, mobil işletim sistemi, mobil uygulama, mobil uygulama geliştirme süreçleri, App Inventor, AI Companion, Android emülatör, uygulama test etme kavramları üzerine tartışma ve bilgi alma

Uygula: Örnek bir uygulamayı AI Companion ile mobil cihazda çalıştırma

Tasarla: App Inventor ara yüzünü keşfederek bir uygulama sayfası tasarlama

Üret: Tasarlanan uygulama sayfasını App Inventor'da hazırlamaya çalışma (görsel olarak)

Değerlendir: Hazırlanan uygulamayı AI Companion ile mobil cihazda çalıştırma

1. Gözle

Dersin başında Programlamaya Yönelik Tutum Ölçeği öğrencilere uygulanır. Ölçeğin uygulanması için internet bağlantısı (<https://forms.gle/s8zWSWTd8ssxQKvD9>) bütün öğrencilere gönderilir ve doldurmaları istenir. Bu form öğrencilerin e-posta bilgilerini toplamaktadır. Eğitim sonunda aynı form tekrar uygulanacaktır. Verilerin eşleştirilmesi amacıyla, öğrencilerinize her iki uygulamada da aynı e-posta adresini kullanmaları gerektiğini bildiriniz.

Eğitmen, öğrencilere sorular yönlendirir ve görüşlerini sunmaları için ortam hazırlar.

- Mobil cihaz deyince ne anlıyorsunuz?
- Bir cihazın mobil cihaz olarak isimlendirilmesi için hangi özelliklere sahip olması gerekmektedir?
- Mobil işletim sistemi nedir?
- Mobil işletim sistemlerine örnekler verebilir misiniz?
- Mobil uygulama marketleri nelerdir?
- Mobil uygulama örnekleri nelerdir?
- Mobil uygulama nasıl geliştirilir?

Sorular hakkında öğrencilerin tartışmaları istenir ve tartışmanın ilerlemediği durumlarda eğitimci konuyu yönlendirir. Eğitimci, konuyu yönlendirmede aşağıdaki bilgilerden yararlanabilir.

1.1. Mobil Cihaz ve Mobil Uygulama

Günümüz dünyasında teknolojik cihazların çoğunun mobil olması yolunda hızla ilerlemektedir. Teknolojik gelişmeler sayesinde çok daha küçük işlemcilerle ve devrelerle güçlü ve aynı zamanda gelişen pil teknolojileri ile de uzun kullanım süreleri sunan cihazlar yaygınlaşmaktadır. İçinde bulunduğumuz bilgi ve iletişim çağında bilgiye hızlıca erişmek ve diğer cihazlarla ve insanlarla hızlı bir iletişim içinde olmak için mobil cihazlar gün geçtikçe daha çok tercih edilmektedir.

1.1.1. Mobil cihaz deyince ne anlıyorsunuz?

Taşınabilir bilgisayarlar da mobil cihazlar sayılsa da her geçen gün kullanım alanları ve kullanıcı sayıları artan akıllı telefonlar ve tabletler mobil cihaz deyince ilk aklımıza gelen cihazlar olmaktadır. Bunların yanında akıllı saatler, cep bilgisayarları, spor takip cihazları, akıllı müzik çalar cihazlar da mobil cihazlar olarak kabul edilmektedir.

1.1.2. Bir cihazın mobil cihaz olarak isimlendirmesi için hangi özelliklere sahip olması gerekmektedir?

- Elektrikle bağlı olmadan kullanım sunan yani içinde bir pil bulunduran
- Tek elle ya da iki el yardımı ile kullanılabilen
- Fiziksel ya da sanal bir klavye veya yönetim ara yüzü bulunan
- Sadece görüntü veren ya da aynı zamanda dokunmatik de olan bir ekran
- Wi-Fi, Bluetooth, NFC ya da USB gibi bağlantı özelliklerinden bir veya birkaçını barındıran cihazlara mobil cihazlar adı verilmektedir.

1.1.3. Mobil işletim sistemi nedir?

Mobil cihazların kullanıcılar tarafından kullanılabilmesini sağlayan, cihazın donanımı ile kullanıcının iletişim kurmasını sağlayan yazılımlara mobil işletim sistemleri denilmektedir. Mobil işletim sistemleri, cihazın donanım özelliklerine göre düzenlenmiş ve cihazın bütün yeteneklerinin kullanıcılar tarafından sorunsuz kullanılmasını amaçlamaktadırlar. Mobil işletim sistemlerinin daha verimli ve kolay kullanılması için mobil uygulama adındaki yazılımlar bulunmaktadır. Mobil uygulamalar, kullanıcıdan aldığı komutları işletim sistemine aktararak mobil cihazın en verimli şekilde kullanılmasını sağlamaktadır.

1.1.4. Mobil işletim sistemlerine örnekler verebilir misiniz?

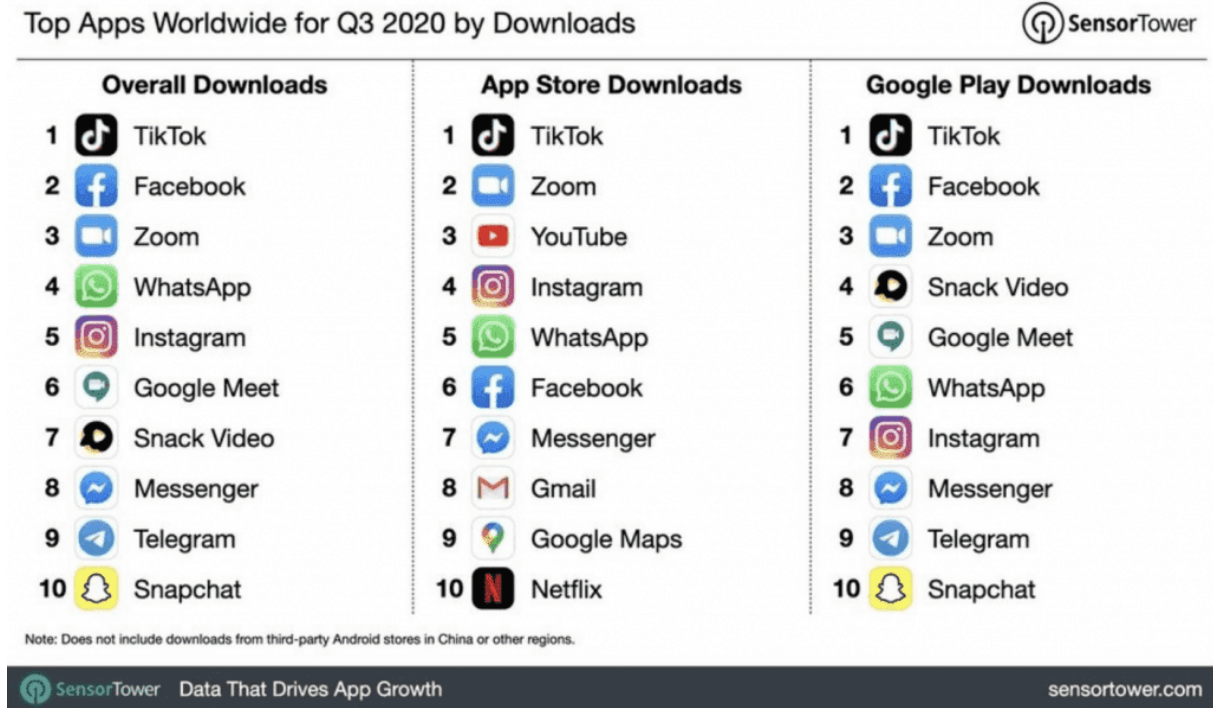
Mobil işletim sistemlerine örnekler vermek istersek, iOS, iPadOS, Android, WearOS, Tizen, Bada, Windows Phone, Symbian gibi örnekler verebiliriz. Bu örneklerin bazıları halen kullanılsa da bazılarının kullanımı ve geliştirilmesi sonlandırılmıştır. Örneğin Bada ve Symbian artık geliştirilmemektedir. Bu işletim sistemlerinin geliştirilmesinin durdurulmasının ana sebebi olarak uygulama marketlerinin yeterince gelişmemesini görebiliriz. Uygulama geliştiriciler çalışmalarının en fazla cihaza ulaşmasını amaçladıkları için en çok kullanılan işletim sistemlerine uygun uygulamalar geliştirmektedirler. Bu sebeple de farklı markaların oluşturdukları işletim sistemlerine yönelmek yerine farklı markaların ortak olarak kullanabilecekleri işletim sistemlerine yönelmeyi tercih etmişlerdir. Bu sebeple; güncel mobil işletim sistemlerine baktığımız zaman iOS ve Android en güncel ve en yaygın işletim sistemleri olarak öne çıkmaktadır.

1.1.5. Mobil uygulama marketleri nelerdir?

Kullandığınız mobil cihaza mobil uygulamaları kurmak için kurulum dosyalarına ulaşmanız ve bu dosyaları mobil cihazınıza indirmeniz gerekmektedir. Bu mobil uygulamaları bulabilmek ve indirmek için mobil işletim sistemlerinin uygulama marketleri bulunmaktadır. Bu uygulama marketlerine örnekler verebilir misiniz? Örneğin eğer iOS veya iPadOs işletim sistemi kullanan bir cihazınıza uygulama indirmek istiyorsanız kullanacağınız uygulama marketi App Store'dur. Ya da Android bir cihaza uygulama indirmek istiyorsanız kullanacağınız uygulama marketi Google Play Store'dur. En çok cihaz tarafından kullanılan işletim sistemleri olarak öne çıkan Android ve iOS cihazların uygulama marketleri bu sebeple en çok kullanılan uygulama marketleri olarak görülmektedir.

1.1.6. Mobil uygulama örnekleri nelerdir?

Mobil cihazlarınızın işletim sistemi nedir? Kullandığınız mobil cihazlarda en çok kullandığınız mobil uygulamalar nelerdir? Sizce en çok kullanılan mobil uygulamalar nelerdir? Yapılan araştırmalara göre en çok kullanılan uygulama marketler olan App Store ve Google Play Store'da 2020 yılının 3. çeyreğinde en çok indirilen uygulamalar aşağıda verilmiştir.



Ekran Görüntüsü 1. 2020 Yılı'nın 3. Çeyreğinde En Çok İndirilen Uygulamalar

Aynı araştırmaya göre uygulama marketlerin yine aynı dönemde kazandığı para incelendiğinde ise App Store 19 milyar dolar gelir elde ederken, Google Play Store ise 10,3 milyar dolar gelir elde etmiştir. Ama bu verilerde göz önünde bulundurulması gereken Çin'de Google Play Store kullanılmadığı ve Android uygulamaların Çin'de farklı uygulama marketlerden indirildiğidir. Yani Android uygulamaların aynı dönemde elde ettiği gelir 10,3 milyar doların da üzerindedir.

1.1.7. Mobil uygulama nasıl geliştirilir?

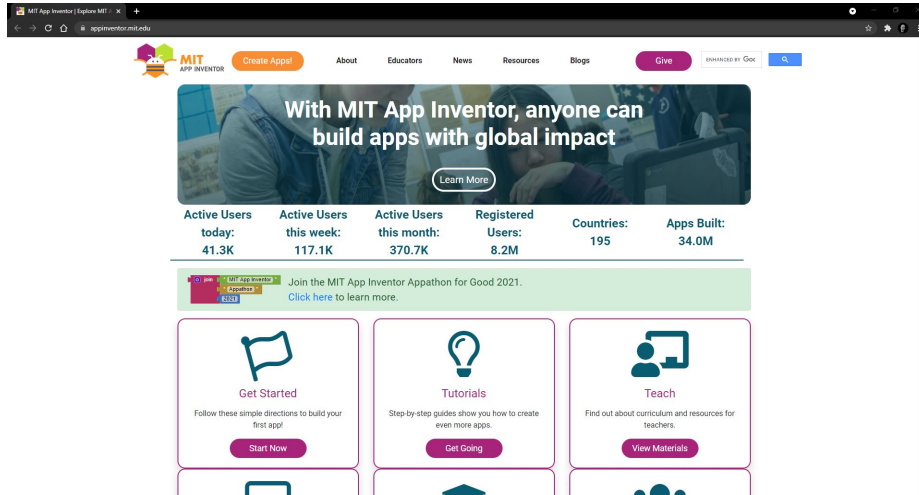
Bütün yazılım geliştirme süreçlerinde olduğu gibi mobil uygulama geliştirme sürecinde de programlama dilleri kullanılmaktadır. Kotlin, Python, C++, Java, Swift gibi programlama dilleri bu programlama dillerine örnek olarak verilebilir. Teknolojinin gelişmesi programlama dillerinin de gelişmesine ve çeşitlenmesine sebep olmuştur. Gün geçtikçe programlama dilleri kullanıcıların daha kolay öğrenebileceği ve yazılım geliştirebileceği şekilde gelişmektedir.

Mobil cihaz kullanımındaki hızlı artış da özellikle mobil uygulama geliştirme süreçlerinin daha çok yaygınlaşmasını sağlamıştır. Mobil uygulama geliştirmede kullanılabilecek programlama dilleri sayısı da gün geçtikçe hızla artmakta ve çeşitlenmektedir. Örnek vermek gerekirse iOS ya da iPadOS için uygulama geliştirmek istiyorsanız Swift ya da Objective C dillerinden birini kullanabilirsiniz. Android uygulama geliştirmek için ise C#, Java, Kotlin ya da Java Script dillerinden birini kullanabilirsiniz.

Programlama dilleri ile uygulama geliřtirmek dıřında uygulama geliřtirmenizi daha kolay hale getiren programlar ve kütüphaneler de bulunmaktadır. Bu program ve kütüphaneler hazır kod parçacıklarının üzerinde sizin hazırlayacağınız uygulamayı derleyerek iřletim sisteminin anlayabileceğı hale getirmektedir. Bu tarz uygulama geliřtirmenin en büyük avantajı çok daha kolay olmasıdır. Dezavantajlarını ele alırsak da aracı bir program kullandığınız için hazırlayacağınız uygulama kullandığınız aracı programın belirlediğı kısıtlılıklar içinde hareket etmektedir. Biraz daha açıklamak gerekirse kullandığınız aracı program size yapabileceğiniz iřlemi belli bir aralıkta yapmanıza izin veriyorsa sizin de uygulamanızda bu aralık içerisinde kalmanız gerekmektedir. Bir diğerk dezavantajı da hazırlayacağınız uygulama çalışmak için kullandığınız aracı programın da zorunlu dosyalarını içinde bulundurmak durumunda olacaktır. Bu ek dosyalar da uygulamanızın çıktısının boyutunu büyütecektir.

1.2. App Inventor

App Inventor ilk olarak **Google** tarafından geliştirilmiş bir mobil uygulama geliştirme aracıdır. En önemli özelliğı sürükle bırak ve blok kodlama ile uygulama geliştirilebiliyor olmasıdır. Google ortaya çıkarmış ve sadece 1 sene sonra bütün kaynak kodlarını açıklayarak projeden çekilmiştir. Sonrasında **MIT (Massachusetts Institute of Technology)** üniversitesi tarafından geliřtirilmeye başlanmıştır. Tamamen ücretsiz olan App Inventor, kullanmak için sadece bir **google** hesabı ile oturum açılmasını gerektirmektedir. <https://appinventor.mit.edu/> adresinden ulaşılabilir. Mobil uygulama geliřtirmeyi amaçlayan App Inventor, Android cihazlarda çalışabilecek uygulamalar geliřtirebilmek için hazırlanmıştır.



Ekran Görüntüsü 2. App Inventor Ana sayfa

1.2.1. App Inventor'ın avantajları nelerdir?

Mobil uygulama geliştirme sürecinde herhangi bir kod yazmaya gereksinim duymadan sadece kod blokları oluşturarak ve sayfa tasarımı için de sürükle bırak tekniği ile kolaylıkla görsel tasarımlar yapılabilmesi sebebiyle App Inventor her yaştan kullanıcıya hitap etmektedir.

Tamamen çevrimiçi çalışması sebebiyle güçlü donanımlar gerektirmeden uygulama geliştirme yapılabilir. Farklı platformlarda uygulama geliştirirken güçlü donanıma sahip bilgisayarlar gerekebilirken (Örneğin; Android Studio), App Inventor sadece internete bağlanabilen ve internet tarayıcısını çalıştırabilen her cihazda kolaylıkla kullanılabilmektedir.

Çevrimiçi çalışması sebebiyle geliştirme yapılan projenin bütün dosyaları bulut ortamında depolanmaktadır. Böylelikle örneğin okulda başladığınız projenize, yanınızda herhangi bir dosya taşımadan evde devam edebilir ya da arkadaşınızın evinde çalışmanıza son halini verebilirsiniz.

1.2.2. App Inventor'ın dezavantajları nelerdir?

Farklı platformlarda uygulama geliştirirken, çalışmanızı test etmek için genellikle aynı platform içinde kullanabileceğiniz ara yüzler bulunurken (Örneğin; AI Starter, Android Studio) App Inventor'da hazırladığınız uygulamanın nasıl çalıştığını görmek için farklı programlara ya da cihazlara ihtiyaç duyarsınız. Bu uygulama ya da cihazlardan dersin devamında detaylı bir şekilde bahsedeceğiz.

Kod yazmadan uygulama geliştirebildiğimiz için daha önce de bahsettiğimiz üzere App Inventor'ın kendi kod kütüphaneleri kullanırız. Bu durum uygulamalarımızın App Inventor'ın bize izin verdiği ölçüde işlemler yapmasına sebep olmaktadır. Aynı zamanda bu kod kütüphaneleri uygulamamızın boyutunun daha büyük olmasına neden olmaktadır.

Kod yazılarak oluşturulan native (yerel) uygulamalar daha hızlı tepki verip, daha hızlı işlemler yapabilirken, App Inventor gibi bir aracı program kullanılarak hazırlanan uygulamalar daha yavaş tepki verip, daha yavaş işlem yapabilmektedir.

1.2.3. App Inventor ile uygulama geliştirme ara yüzüne nasıl ulaşılır?

<https://appinventor.mit.edu/> internet bağlantısından açılan sayfadan sol üst kısımdaki "Create Apps!" düğmesi tıklanarak uygulama geliştirme ortamına giriş yapılabilir.

Bu kısım için eklerde bulunan video öğrencilere gösterilebilir. (<https://youtu.be/22UGvmFhA64>)

1.2.4. App Inventor ile uygulama geliştirme ara yüzü nasıl kullanılır?

Eğitmen tarafından App Inventor uygulama geliştirme ara yüzü açılır ve öğrencilerin de bu ara yüze girmeleri sağlanır. Sonrasında ara yüzdeki Designer ve Blocks sekmeleri öğrencilere gösterilir.

Designer sekmesinde kullanılabilecek düğmeler öğrencilere tanıtılır ve sürükle bırak yöntemi ile projeye nasıl ekleneceği anlatılır. Bu kısım için eklerde bulunan video öğrencilere gösterilebilir (https://youtu.be/9b_ZzzwPUgc).

Blocks sekmesindeki kod parçacıkları öğrencilere tanıtılır ve sürükle bırak yöntemi ile nasıl projeye eklenebileceği gösterilir. Bu kısım için eklerde bulunan video öğrencilere gösterilebilir (<https://youtu.be/lBgHCQjsTjc>).

1.2.5. App Inventor ile geliştirilen uygulamaların çıktıları nasıl görüntülenebilir?

App Inventor ile geliştirilen uygulamaların çıktıları görebilmek için 3 farklı yöntem vardır.

Birincisi AI Companion uygulamasıdır. Bu uygulamayı Android ya da iOS mobil cihazınıza kurduktan sonra geliştirme yaptığınız mobil cihazla aynı yerel ağa bağlanması gerekmektedir. Bu kısım için eklerde bulunan video öğrencilere gösterilebilir (https://youtu.be/trJ6lx6jS_I).

İkinci bir yöntem Android emülatör kullanımıdır. Emülatörün kullanılabilmesi için bilgisayarınıza AI Starter yazılımının yüklü olması gerekmektedir. AI Starter'ın kurulumu ve emülatörün kullanımı ile ilgili eklerde bulunan video öğrencilere gösterilebilir (<https://youtu.be/rJRmopb9sQg>).

Üçüncü ve son yöntem USB bağlantısı ile bir Android cihazın bilgisayara bağlanması ile gerçekleştirilebilir. Bilgisayarınızda Android cihaza ait sürücülerin yüklü olması ve Android cihazda gerekli ayarların yapılmış olması gerekmektedir.

2. Uygulama

Eđitmen tarafından ders 6ncesinde bir 6rnek App Inventor uygulaması hazırlanabilir ya da eklerde bulunan 6rnek uygulama dosyası eđitmen tarafından derste g6sterilebilir. Bu 6rnek uygulama 6đrencilerle de paylaşılr ve 6đrencilerin bu uygulamayı AI Companion yazılımını kullanarak mobil cihazlarında g6r6nt6leyebilmesi beklenir.

3. Tasarla

App Inventor uygulama geliřtirme ara y6z6 hakkında bilgi alan 6đrencilerden bu ara y6z6 keřfetmesi ve neler yapabileceklerini kendilerinin fark etmesi istenir. Sonrasında 6đrencilerden bir konu belirleyerek App Inventor ile geliřtirebilecekleri bir mobil uygulama 6rneđi fikri bulmaları istenir. Buldukları bu fikirleri not etmeleri ve kâđıt kalem yardımı ile uygulamalarını g6rsel olarak tasarlamaları i6in s6rece rehberlik edilir.

4. 6ret

6đrencilerin kâđıt 6zerinden oluřturdukları uygulama tasarımlarını App Inventor'da g6rsel olarak oluřturması istenir. Resim, d6đme, liste, video gibi par6aları ekrana s6r6kle bırak y6ntemi ile kolayca yerleřtirebileceklerini kavramaları beklenir. Eđitmen bu s6re6te rehber g6revi g6rmektedir.

5. Deđerlendir

6đrencilerin App Inventor'da oluřturdukları ilk projeleri olan uygulama tasarımlarının AI Companion uygulaması ile mobil cihazlarında g6r6nt6lenmesi ve test edilmesi sađlanır. 6đrencilerin kendi uygulama ekranlarını akranlarına g6sterebilmesi i6in gerekli ortam hazırlanır.

6đrencinin 6z deđerlendirme yapabilmesi i6in 6z Deđerlendirme Formu - 1 (<https://forms.gle/jiofyNvaDj48fJ9P8>) 6đrenciye uygulanır.

2. Hafta: App Inventor Arayüzü ve Uygulama Geliştirme

Ön Bilgi:

- Temel bilgisayar bilgisi
- Temel yazılım bilgisi

Haftanın Kazanımları:

- App Inventor arayüz nesnelerini tanır.
- App Inventor ile uygulama ara yüzü geliştirebilir.
- App Inventor kod bloklarını tanır ve kullanımlarını kavrar.
- App Inventor ile uygulama kodlayabilir.
- App Inventor ile veri giriş-çıkış nesnelerini kullanabilir.
- App Inventor ile değişken tanımlayabilir.
- App Inventor operatörlerini kullanabilir.
- App Inventor proje geliştirme sürecini kavrar ve proje geliştirme sürecine başlar.

Haftanın Amacı:

App Inventor ile uygulama oluşturmak için iki aşama olduğunu ve bunların tasarım ve kodlama aşamaları olduğunu öğrenir. Tasarım aşamasındaki nesneleri ve kullanımını, kodlama aşamasındaki kod bloklarını ve kullanımını öğrenir ve deneyimler. Kullanıcıdan veri alabilecek, aldığı veriyi operatörlerle işleme dahil edebilen ve bir çıktı verebilen bir uygulama hazırlayabilir. Proje geliştirme süreci hakkında bilgi sahibi olur ve kendi projesini oluşturmak için yol haritasını hazırlar.

Kullanılacak Malzemeler:

Bilgisayar, Mobil cihaz (cep telefonu ya da tablet), kalem, kâğıt

Haftanın İşlenişi:

Gözle: App Inventor tasarım ve kodlama bölümlerini, bu bölümlerdeki nesne ve kod bloklarını tanıma, özelliklerini öğrenme, veri giriş-çıkış süreçlerini kavrama, değişken yapısı, değişken tanımlama ve operatör kullanımı hakkında bilgi alma

Uygula: Veri giriş ve çıkışını, değişken tanımlama ve kullanımını, operatör kullanan bir uygulama hazırlama

Tasarla: Kullanıcıdan bir veri alarak bu aldığı veriyi operatör kullanarak bir işlemden geçiren ve sonucu ekrana yazdıran bir uygulamanın tasarımını yapma

Üret: Kullanıcıdan bir veri alarak bu aldığı veriyi operatör kullanarak bir işlemden geçiren ve sonucu ekrana yazdıran bir uygulamanın görsel tasarımını ve kodlamasını yapma

Değerlendir: Proje geliştirme süreci için yol haritasını belirleme ve uygulama konusunu seçme

1. Gözle

Bir önceki haftalarda kullanmaya başladığımız App Inventor ile uygulama geliştirme sürecinde öğrencilere aşağıdaki sorular yöneltilir.

- Tasarım sekmesinde bulunan nesneleri incelediğinizde kullandığınız uygulamalarda da bulunan nesneler hangileridir?
- Bu nesnelerin ne işe yaradıkları ile ilgili düşünceleriniz nelerdir?
- Uygulamaya ekleyeceğimiz nesnelerle uygulamamızı yapmasını istediğimiz her işlemi gerçekleştirebilir miyiz?
- Bir uygulamada kodlamaya neden ihtiyaç duyarız?
- Değişken nedir ve uygulama geliştirirken neden değişkenlere ihtiyaç duyarız?
- Kodlama yaparken kullanılan “operatör” nedir? Operatör çeşitleri neler olabilir?

Sorular hakkında öğrencilerin tartışmaları istenir ve tartışmanın ilerlemediği durumlarda öğretmen konuyu yönlendirir. Öğitmen, konuyu yönlendirmede aşağıdaki bilgilerden yararlanabilir.

1.1. Tasarım sekmesi nasıl kullanılır?

Eğitmen App Inventor uygulama geliştirme ekranını açar. Ara yüzde bulunan bölümler tanıtılır ve işlevleri açıklanır.

1.1.1. Palette (Palet) sekmesi

Uygulama geliştirirken kullanılacak nesneler bu sekmede bulunmaktadır. Bu nesneler sürükle-bırak yapılarak uygulamaya eklenebilir. Palette sekmesi nesnelerin özelliklerine göre 12 bölüme ayrılmıştır. Bu bölümlerden ayrı ayrı bahsetmek gerekirse;

1.1.1.1. User Interface (Kullanıcı Ara yüzü)

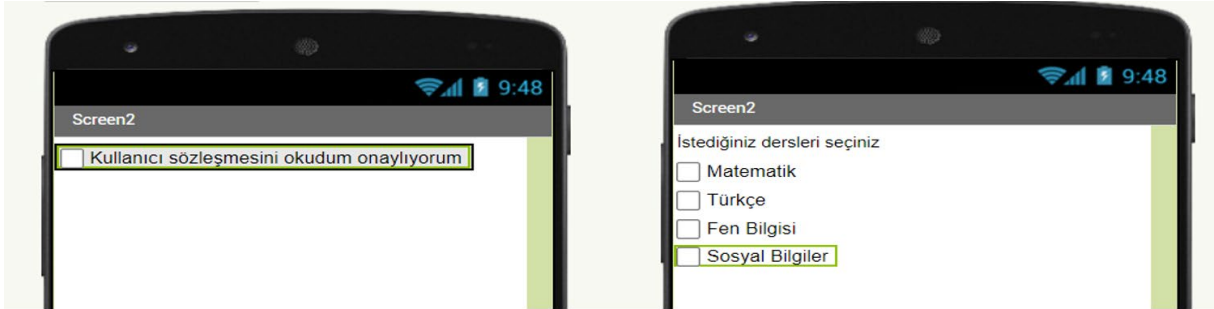
Bu bölümdeki nesneler uygulamanın kullanıcı tarafından kullanılmasını sağlayacak nesnelerdir. Kullanıcı ve uygulama arasındaki etkileşimi sağlayacak bu nesneleri kısaca tanımlamak istersek;

Button (Buton): Bir işlem yapmak için gereken tetikleme işleminin yapılacağı nesnedir. Örnek kullanım olarak; kullanıcıdan istenilen bilgiler doldurulduktan sonra “Gönder” işlemi yapılması için kullanılabilir.



Ekran Görüntüsü 3. Buton nesnesi örneği

CheckBox (Onay Kutusu): Kullanıcıdan bir seçim bilgisi alınmak istediğinde kullanılır. Örnek kullanım olarak; “Kullanıcı sözleşmesini okudum, onaylıyorum” şeklindeki bilgilendirmelerin yanında onay almak için kullanılabilir ya da ders seçimi yapılabilecek bir uygulamada kullanıcının istediği dersleri seçmesini sağlaması için kullanılabilir.



Ekran Görüntüsü 4. Onay Kutusu nesnesi örneği

DatePicker (Tarih Seçici): Kullanıcıdan bir tarih bilgisi alınmak istendiğinde kullanılabilir. Örnek kullanım olarak; kullanıcının doğum tarihini girebilmesi için kullanılabilir.



Ekran Görüntüsü 5. Tarih seçici nesnesi örneği

Image (Resim): Uygulama ekranında bir resim eklenmek istendiğinde kullanılabilir. Örnek kullanım olarak; uygulama ekranlarında bulunan her görsel kullanımı düşünülebilir.

Label (Etiket): Uygulama ekranında bulunan bütün sabit (değişmeyecek) metinler için kullanılabilir. Örnek kullanım olarak; uygulama ekranındaki bütün açıklama yazıları düşünülebilir.

ListPicker (Liste Seçici): Uygulamada yeni bir sayfa şekilde açılan bir listeden kullanıcının bir seçim yapması sağlanabilir. Bir tane seçim yapmaya izin vermektedir, çoklu seçim yapılamaz. Örnek kullanım olarak; kullanıcıdan notlarını görmek istediği bir dersi seçmesi istendiği bir uygulamada kullanılabilir.



Ekran Görüntüsü 6. Liste seçici nesnesi örneği

ListView (Liste Görünümü): Uygulama ekranında bir liste gösterilmek istediğinde kullanılabilir. Kullanıcıdan tek bir seçim yapması beklenir. Ekranın belli bir bölümünde o bölüme sığabilecek nesneden daha fazla bilgi verilmek istendiğinde kullanılır. Örnek kullanım olarak; 8 tane ders adının listelenmesi gereken bir uygulamada bu ders isimlerinin bütün sayfayı kaplamayıp ekranda 3 tanesinin görüneceği, kullanıcı isterse listeyi aşağı yukarı hareket ettirerek diğer dersleri de görebileceği ve istediği bir dersi seçebileceği bir uygulamada kullanılabilir.



Ekran Görüntüsü 7. Liste görünümü nesnesi örneği

Notifier (Bildirim Gösterme): Kullanıcıya bir uyarı mesajı vermek için kullanılır. Görünmeyen bir bileşendir. Kod blokları ile düzenlenir.



Ekran Görüntüsü 8. Bildirim gösterme nesnesi örneği

PasswordTextBox (Şifre Yazı Kutusu): Metin girişi için kullanılan bir nesnedir, en önemli özelliği kullanıcı metin girişi yaparken yazılan karakterler gözükmez. Özellikle şifre bilgisi alınırken kullanılır.



Ekran Görüntüsü 9. Şifre yazı kutusu nesnesi örneği

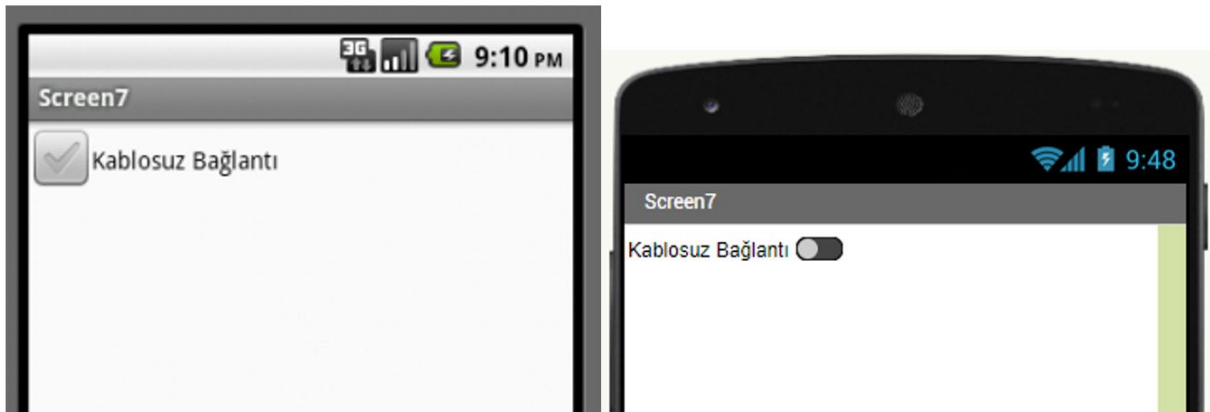
Slider (Slayt Gösterici): Ekranı birden çok görselin gösterilebileceği ve kullanıcının sağa sola kaydırarak bunlar arasında geçiş yapabileceği bir slayt gösterisi eklemek için kullanılır.

Spinner (Döndürücü): Kullanıcının bir listeden seçim yapmasının istendiği durumlarda kullanılır. Ekranı bir liste kutusu olarak eklenir ve tıklandığında ekrana bir pop-up açarak kullanıcının seçim yapmasını sağlar. Seçilen değer liste kutusu içerisinde gösterilir. Özellikle kullanıcının çok fazla seçenek arasından seçim yapabileceği durumlarda kullanılır.



Ekran Görüntüsü 10. Döndürücü nesnesi örneği

Switch (Anahtar): Kullanıcının iki durum arasında seçim yapmasının istendiği zamanlarda kullanılır. Örneğin açık-kapalı gibi seçimlerin kullanıcıdan alınmasını sağlar. Farklı Android sürümlerinde değişik görünebilir, örneğin Android 4 ve öncesi sürümlerde onay kutusu şeklinde görülürken sonraki sürümlerde anahtar şeklinde görünür.



Ekran Görüntüsü 11. Anahtar nesnesi örneği

TextBox (Metin Kutusu): Kullanıcıdan veri girişı isteneceęi zaman kullanılır. Çok farklı amaçlarda kullanılabilir. Hem metin ve hem de sayısal veri giriřleri yapılabilir olsa da girişı yapılan sayısal veriler de tür olarak string'dir (sayısal işlem yapılamayan veri türü).



Ekran Görüntüsü 12. Metin kutusu nesnesi örneęi

TimePicker (Saat Seçici): Kullanıcıdan bir saat bilgisi alınacağı zaman kullanılır. Ekranda saat bilgisinin girilebilmesi için bir pop-up belirir ve kullanıcı buradan saat bilgisi girişı yapar.



Ekran Görüntüsü 13. Saat seçici nesnesi örneęi

WebView (İnternet Görüntüleyici): Uygulama içerisinde bir internet sayfasının gösterilmesi için kullanılır. Sayfada belirlenen alan içerisinde belirtilen internet sayfası görüntülenir ve bu internet sitesi üzerinde bütün işlemler de yapılabilir. Yani uygulamamıza bir tarayıcı ekleyerek kullanıcının internette gezebilmesini sağlar.

1.1.1.2. Layout (Yerleşim)

Bu bölümdeki nesneler kullanıcıların kullandığı tasarım öğelerinin ekranda nasıl görüneceği ile ilgili düzenlemelerini yapmalarını sağlar. Nesnelerin birbirlerine göre ekrana göre nasıl hizalanacağı bu bölümdeki nesneler ile tasarlanır.

Horizontal Arrangement (Yatay Düzenleme): Bu nesne ekrandaki nesnelerin yatay olarak hizalanmasını sağlamak amacıyla kullanılır. Ekrana bu nesne ile görünmeyen bir hizalama aracı eklenir. Eklenen bu hizalama aracı içerisine eklenen nesnelerin hangi konumda duracağı, hizalama aracının ekrandaki pozisyonu ve boyutu ile ilgili ayarlar Özellikler kısmından ayarlanabilir.

Horizontal Scroll Arrangement (Yatay Kaydırılabilir Düzenleme): Horizontal Arrangement ile aynı özelliklere sahip bu araç ise ekranda ne kadarlık bir yer kaplayacağı ayarlanabilen ve eğer ekrana tek seferde sığabilecek nesneden fazlası eklenmesi durumunda araç içerisinde yatay bir kaydırma işlemi yaparak nesneler arasında geçiş yapmayı sağlamaktadır.

Table Arrangement (Tablo Hizalama): Bu nesne ile ekrana görünmeyen bir tablo eklenerek bu tablonun hücrelerine nesneler eklenmesi ile hizalama yapılmasını sağlamaktadır. Özellikler kısmından tablonun boyutu ve nesnelerin yerleşimi ile ilgili ayarlar yapılabilir.

Vertical Arrangement (Dikey Hizalama): Bu nesne ekrandaki nesnelerin dikey olarak hizalanmasını sağlamak amacıyla kullanılır. Ekrana bu nesne ile görünmeyen bir hizalama aracı eklenir. Eklenen bu hizalama aracı içerisine eklenen nesnelerin hangi konumda duracağı, hizalama aracının ekrandaki pozisyonu ve boyutu ile ilgili ayarlar Özellikler kısmından ayarlanabilir.

Vertical Scroll Arrangement (Dikey Kaydırılabilir Düzenleme): Vertical Arrangement ile aynı özelliklere sahip bu araç ise ekranda ne kadarlık bir yer kaplayacağı ayarlanabilen ve eğer ekrana tek seferde sığabilecek nesneden fazlası eklenmesi durumunda araç içerisinde dikey bir kaydırma işlemi yaparak nesneler arasında geçiş yapmayı sağlamaktadır.

1.1.1.3. Media (Medya)

Bu bölümden uygulamanıza medya nesneleri yani resim, video kaydı, ekran kaydı, ses veya video oynatıcı gibi nesneler ekleyebilirsiniz. Bu nesneler sonraki konularda detaylı olarak ele alınacaktır.

1.1.1.4. Drawing and Animations (Çizim ve Animasyonlar)

Bu bölümde uygulamanızda çizim nesneleri kullanabilir ya da animasyonlar ekleyebilirsiniz. Bu bölümdeki nesneler sonraki konularda detaylı olarak ele alınacaktır.

1.1.1.5. Maps (Haritalar)

Bu bölümden uygulamanıza navigasyon nesneleri ya da harita üzerinde konumlandırma yapabileceğiniz nesneler ekleyebilirsiniz. Bu nesneler sonraki konularda detaylı olarak ele alınacaktır.

1.1.1.6. Sensors (Sensörler)

Bu bölümden uygulamanıza mobil cihazınızın sensörlerini kullanabileceğiniz nesneler ekleyebilir ve bu sensörlerden alınan verilerle işlemler yapmanıza izin veren uygulamalar hazırlayabilirsiniz. Bu bölümdeki nesneler sonraki konularda detaylı olarak ele alınacaktır.

1.1.1.7. Social (Sosyal)

Bu bölümden uygulamanıza sosyal medya uygulamaları üzerinden yapılabilecek işlemler ekleyebileceğiniz gibi telefon numarası, e-posta adresi gibi iletişim araçlarını da kullanarak paylaşımlar yapabileceğiniz düzenlemeler yapabilirsiniz. Bu bölümdeki nesneler sonraki konularda detaylı olarak ele alınacaktır.

1.1.1.8. Storage (Depolama)

Uygulamanızda herhangi bir depolama işlemi yapacaksanız ya da bir veri tabanı bağlantısı yapacak ve bu veri tabanını uygulamanızda kullanacaksanız bu bölümdeki nesneleri kullanabilirsiniz. Bu nesneler sonraki konularda detaylı olarak ele alınacaktır.

1.1.1.9. Connectivity (Bağlantı)

Bu bölümden uygulamanıza bluetooth ya da internet bağlantısı gibi elektronik haberleşme özellikleri ekleyebilir ve bunları yönetebilirsiniz. Bu bölümdeki nesneler sonraki konularda detaylı olarak ele alınacaktır.

1.1.1.10. LEGO® MINDSTORMS®

Lego mindstorms setleri ile haberleşebilecek uygulamalar hazırlayarak bu setlere komutlar vermenizi sağlayacak nesneler bu bölümde bulunmaktadır.

1.1.1.11. Experimental (DeneySEL)

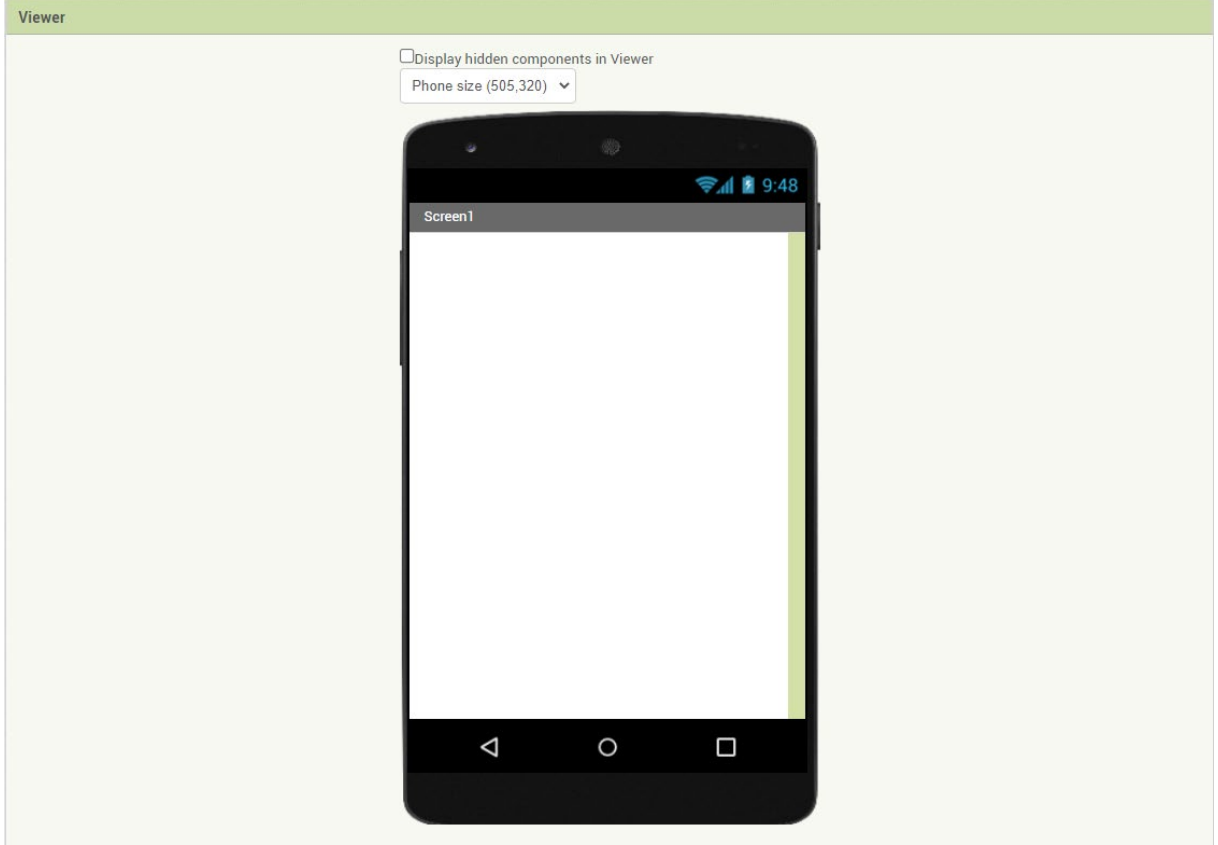
Bu bölümde App Inventor geliştirme sürecine yeni dahil edilen özellikler bulunmaktadır. Bu bölümdeki nesneleri uygulamanızda kullanabilirsiniz fakat henüz geliştirme süreci devam ettiği sürece bazı hatalarla karşılaşma olasılığınız da bulunmaktadır. Yeni eklenen nesneler ile ilgili bütün denemeler tamamlanıp sorunsuz çalışır hale geldikten sonra bu bölümden kaldırılıp ilgili bölümde kullanılmaya başlanabilir ya da çözülemeyecek sorunlarla karşılaşılması durumunda App Inventor tarafından desteği tamamen kaldırılıp bu bölümden de kaldırılabilir.

1.1.1.12. Extension (Eklenti)

App Inventor geliştirilmesi halen devam eden ve açık kaynak kodlu bir uygulama geliştirme platformudur. Açık kaynak kodlu olması sebebiyle uygulama dışından yeni eklentiler yüklenerek bunları projenizde kullanmanıza da olanak sağlamaktadır. Bu bölüm sizin yeni eklentiler yükleyerek projenizde kullanmanızı sağlamaktadır.

1.1.2. *Viewer (Görüntüleyici) sekmesi*

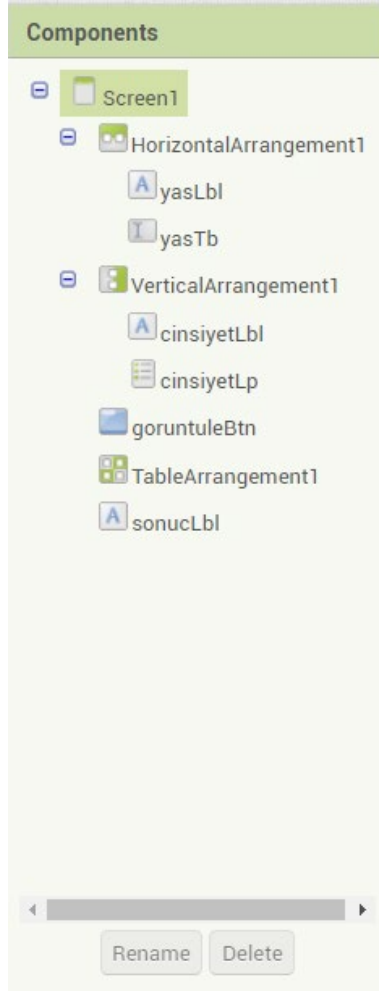
Uygulamaya eklenen bütün nesnelerin örnek bir mobil cihazda nasıl görüneceği ile ilgili bir ön izleme görebileceğimiz bölümdür. Ön izlemede kullanılacak mobil cihazın nasıl bir cihaz olmasını istediğimiz ile ilgili ayarlar yapmamıza olanak sağlar.



Ekran Görüntüsü 14. *Viewer (Görüntüleyici) sekmesi*

1.1.3. Components (Bileşenler) sekmesi

Uygulamaya eklenen nesneleri liste şeklinde görebileceğimiz, nesneleri hiyerarşik olarak yerleştirmemize olanak sağlayan, eklenen nesnelerin silinmesini ya da adının değiştirilmesini yapabileceğimiz bölümdür.



Ekran Görüntüsü 15. Components (Bileşenler) sekmesi

1.1.4. Media (Medya) sekmesi

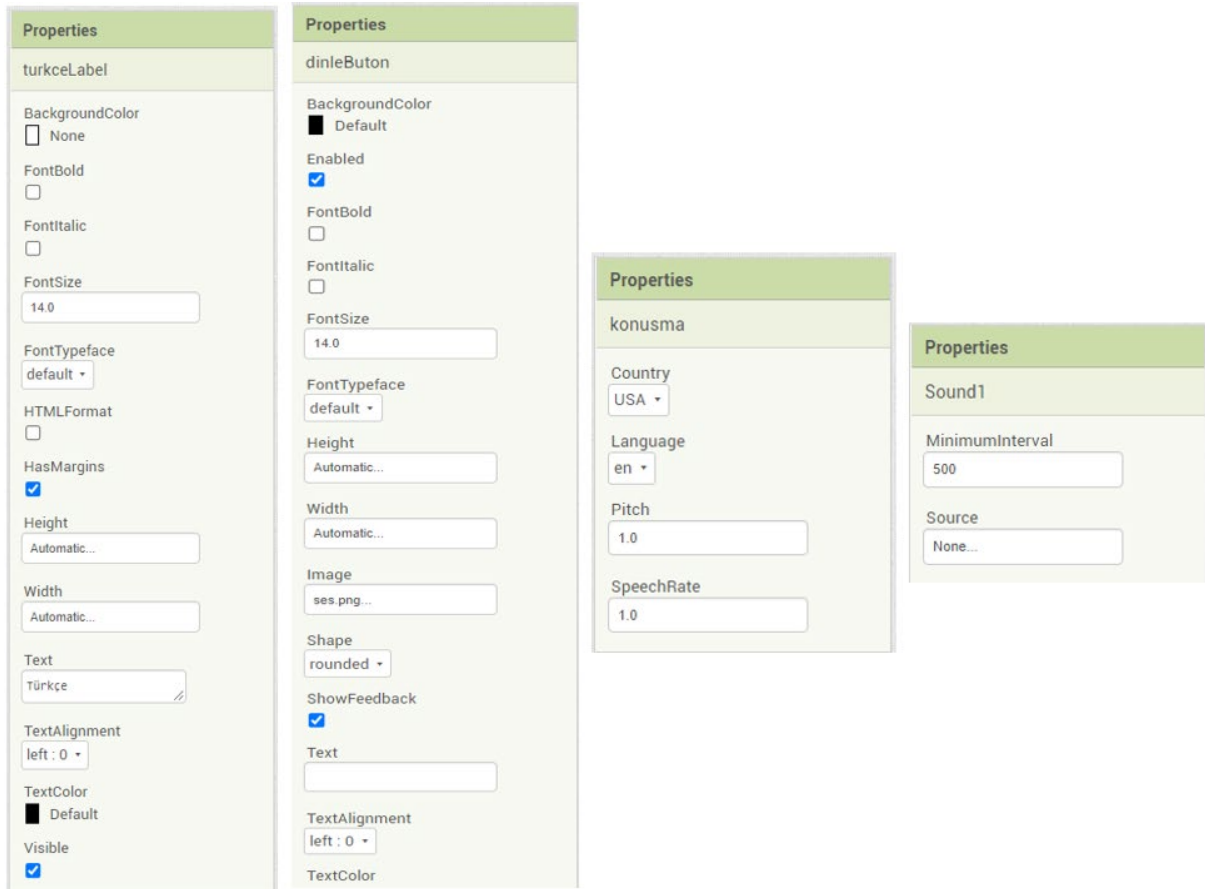
Uygulamaya medya (resim, video, ses) dosyalarının yüklenmesini sağlayan, yüklenen medya dosyalarının listelendiği, ön izlenebildiği ve silinebildiği sekmedir.



Ekran Görüntüsü 16. Media (Medya) sekmesi

1.1.5. Properties (Özellikler) sekmesi

Uygulamaya eklediğimiz nesneler ile ilgili özellikleri görüntüleyebileceğimiz ve düzenleyebileceğimiz sekmedir. Eklenen nesneye göre özellikler ekranı değişiklik gösterir. Hangi nesne ile ilgili özellikleri görmek ya da değiştirmek istiyorsak ilgili nesneyi components(bileşenler) sekmesinden seçebiliriz.



Ekran Görüntüsü 17. Birkaç Properties (Özellikler) sekmesi örneği

1.2. Bloklar sekmesi nasıl kullanılır?

Uygulama geliştirme sürecinde tasarım ekranında yapılan düzenlemeler ile uygulamanın nasıl görüneceği ayarlanabilir fakat uygulamanın yapacağı işlemleri belirlemek ve bunları yönetmek için yapılması gereken işlemler için Blocks (Bloklar) sekmesi kullanılır. Bu sekmedeki nesneler aşağıda kısaca açıklanmıştır.

1.2.1. Built-in (Yerleşik)

Bu bölümde App Inventor içerisinde sürükleyip bırak metoduyla kullanılabilecek yerleşik bütün kod blokları bulunmaktadır. Bu bölümdeki kod blokları sağ tarafta bulunan Viewer (Görüntüleyici) bölümüne sürükleyerek götürülebilir ve burada gerekli düzenlemeler yapılarak çalışacak kod parçacıkları haline getirilebilir. Bu bölümdeki kod blokları aşağıda tanımlanmıştır.

Control (Kontrol): Bu bölümde uygulamada karar verilmesi gereken durumlarda kullanılacak kod blokları bulunmaktadır. Gerçekleştirilecek bir işlemin tanımlanması veya bu işlemin başka bir koşula bağlanması gibi durumlar bu kod blokları ile tanımlanır.

Logic (Mantık): Kod bloklarında kullanılacak mantıksal operatörler bu bölümde bulunmaktadır. Mantıksal operatörler; doğru-yanlış, herhangi bir eşitlik durumu, birden fazla durumun doğruluğunun belirtilmesi gibi durumlar olarak tanımlanabilir. Örnek vermek gerekirse kullanılacak bir kontrol kod bloğunda kontrol edilmesi beklenen koşulun tanımlanması bu kod blokları ile yapılır.

Math (Matematiksel): Uygulamada kullanılacak matematiksel işlemlerle ilgili operatörler bu bölümde bulunmaktadır. Bütün sayısal değer içeren işlemler, bütün matematiksel işlemler bu kod blokları ile tanımlanır. Örneğin değeri bir sayı olan değişkene değer ataması yapılırken ve bu sayısal değerle yapılacak matematiksel işlemler, örneğin dört işlem, bu bölümdeki kod blokları kullanılır.

Text (Metin): Metin tabanlı işlemler bu kod blokları ile yapılmaktadır. Örneğin metin değeri alacak bir değişkenin tanımlanması ya da bu değişken ile yapılacak bir karşılaştırma işlemi bu bölümdeki kod blokları ile yapılır.

Lists (Listeler): Değişkenlerle oluşturulacak diziler ve bu dizilerde yapılacak işlemler bu bölümdeki kod blokları ile oluşturulur. Örneğin yeni bir dizi oluşturacağınız zaman ve bu diziye eleman ekleme çıkarma işlemleri yapacağınızda bu kod bloklarını kullanabilirsiniz.

Dictionaries (Sözlükler): Bu bölümdeki kod blokları bir "sözlük" oluşturmanızı sağlar. Sözlük kavramı diğer kodlama ortamlarında açıklayıcı diziler, veri setleri veya haritalar şeklinde de isimlendirilebilir. Basit bir veri tabanı olarak da düşünebileceğiniz sözlükler, size belirlendiğiniz bir anahtara belli bir değer atamanızı, bu şekilde bir liste oluşturmanızı ve bu değerlerle farklı işlemler yapmanızı sağlamaktadır. Örnek olarak oluşturduğunuz bir sözlükteki elma, armut, patlıcan, muz anahtarlarına sırasıyla meyve, meyve, sebze, meyve değerlerini verebilir ve sonrasında kaç tane meyve olduğuna göre farklı işlemler yapacağımız kod blokları hazırlamak için bu bölümdeki kod bloklarını kullanabilirsiniz.

Colors (Renkler): Uygulamanızda kullandığınız renklerin tanımlanması ya da kontrol ve karşılaştırma işlemleri yaparken bu bölümdeki kod blokları kullanılabilir.

Variables (Değişkenler): Uygulamanızda tanımlayacağınız ya da değer atama ve değiştirme işlemi yapacağınız değişkenlerle ilgili işlemler bu bölümdeki kod blokları ile yapılır. Değişkenlere bağlı oluşturulan karşılaştırma ya da yorumlama kod blokları bu bölümdeki kod blokları ile birlikte çalışırlar. Örneğin “yemek” ve “adet” adlı iki değişken tanımlayıp bu değişkenlere kullanıcıdan alınan değerler ile değer ataması yapmak ve sonrasında da bu değerlere göre verilecek siparişin oluşturulmasını sağlayacağınız bir uygulama hazırlarken bu bölümdeki kod blokları kullanılabilir.

Procedures (Prosedürler): Bu bölümdeki kod blokları bir işlem şablonu oluşturmanızı ve bu şablonu değiştirmenizi ve çalıştırmanızı sağlar. Belli adımlardan oluşan ve program içerisinde birden çok kez kullanılabilecek işlemlerin bir grup halinde oluşturulmasını ve her seferinde tekrar bu işlemlerin gerçekleşmesini sağlar. Örneğin kullanıcıdan alınan bir sayının her seferinde 2 çıkarılmasını, 5 ile çarpılmasını ve 15 eklenmesini istiyorsak bunun için bir prosedür oluşturup kullanıcıdan alınan sayıya bu prosedürün uygulanmasını sağlayabilmek için bu bölümdeki kod bloklarını kullanabilirsiniz.

1.2.2. Screen (Ekran)

Uygulamanıza eklediğiniz sayfalar ile ilgili işlemleri bu bölümden düzenleyebilirsiniz. Yukarıda anlatılan yerleşik kod blokları dışında uygulamanıza tasarım ekranında eklenen her nesne, kod blokları ekleyebilmeniz ve dolayısıyla işlemler yapmanız için Ekran bölümüne eklenmektedir. Örneğin uygulamanızın Screen1’inde (Ekran1) kullandığınız bir butonun tıklama işlemi sonrası yapacağı işlemleri Screen1 altında yer alan butonunuzun adıyla oluşan bölümden ayarlayabilirsiniz.

2. Uygula

Kullanıcıdan yaş bilgisi ve cinsiyet bilgisi isteyen ve aldığı bu bilgileri birleştirip ayrı bir satırda kullanıcının yaş ve cinsiyet bilgilerini cümle olarak gösteren bir uygulama hazırlayalım.

3. Tasarla

Kullanıcı hedef kitlesi öğrenciler olan bir uygulama fikri hazırlayalım. Uygulama, öğrencilerin bir dönemde aldığı derslerden birinin adını yazacağı, sınav sonuç bilgilerini girebileceği ve bu sınav sonuç bilgilerinin ortalamasını alarak bir dönem geçme notu hesaplayacak bir uygulamanın algoritmasını tasarlayalım.

4. Üret

Tasarla bölümünde algoritmasını tasarladığımız uygulamayı App Inventor programında önce Design bölümünden ara yüz hazırlayıp sonra da Blocks bölümünden kodlama adımlarını gerçekleştirerek çalışabilen bir uygulama oluşturup test edelim.

5. Değerlendir

Üret adımında oluşturulan uygulama dışa aktar yapılarak kaydedilir. Uygulama dosyası eğitmen ve diğer öğrencilerle paylaşılır. Bütün öğrenciler kendi uygulamasının yanında yapılan diğer uygulamaları da inceler. Öğrencinin öz değerlendirme yapabilmesi için **Öz Değerlendirme Formu – 1** (<https://forms.gle/jiofyNvaDj48fJ9P8>) öğrenciye uygulanır.

Ayrıca dersin sonunda sunumu yapılmak üzere öğrencilerin bir proje fikri belirleyip her hafta edindikleri bilgileri de kullanabilecekleri bir proje geliştirmeleri istenir. Bu projenin özgün, belirlenmiş amacı gerçekleştirebilen ve test edilebilir düzeyde tamamlanmış olması beklenmektedir. Öğrencilerinizi bu konuda bilgilendirerek, bir sonraki haftaya kadar proje fikri oluşturmalarını isteyiniz.

3. Hafta: App Inventor Medya, Döngü, Koşul ve Hazır Modüller Kullanımı

Ön Bilgi:

- Temel bilgisayar bilgisi
- Temel yazılım bilgisi

Haftanın Kazanımları:

- App Inventor medya ekleme ve yönetme nesnelerini tanır.
- App Inventor ile geliştirilen uygulamalara görüntü ve video dosyaları ekleyebilir.
- App Inventor ile geliştirilen uygulamalara ses dosyaları ekleyebilir.
- App Inventor ile geliştirilen uygulamalarda koşul yapılarını kavrar.
- App Inventor ile geliştirilen uygulamalarda döngü komutlarını kavrar.
- App Inventor ile geliştirilen uygulamalarda iç içe döngüleri kullanabilir.
- App Inventor metin okuma (TextToSpeech) modülünü kullanabilir.
- App Inventor Yandex Translate (çeviri) modülünü kullanabilir.

Haftanın Amacı:

App Inventor ile uygulama geliştirirken görsel ve işitsel medyaları uygulamaya dahil edebilir ve bunların kontrollerini yapabilmeyi, eklenen medyaların isimlendirmesini ve yönetilmesini öğrenir. Uygulama geliştirirken blok kodlama sürecinde koşul yapılarını kullanır, döngüler oluşturabilir. Kullanılacak algoritmanın gereksinimlerine göre iç içe döngü yapıları geliştirebilir. Uygulama içerisindeki metinlerin seslendirilmesi için TextToSpeech modülünü kullanabilir. Bir sözlük uygulaması hazırlamak için App Inventor içerisinde bulunan Yandex Translate (çeviri) modülünü uygulamasına ekleyip internet destekli bir çeviri uygulaması hazırlayabilir.

Kullanılacak Malzemeler:

Bilgisayar, Mobil cihaz (cep telefonu ya da tablet), kalem, kâğıt

Haftanın İşlenişi:

Gözle: App Inventor'da uygulama geliştirirken görsel ve işitsel medya öğelerini ekleme ve yönetme, koşul yapılarını kavrama ve kullanma, döngü ve iç içe döngü yapılarını oluşturma, TextToSpeech modülünü kullanma, Yandex Translate modülünü kullanma

Uygula: Görsel ve ses dosyaları eklediği, komutların belli koşullar durumunda çalışmasını sağlayacağı, kullanıcıdan alınan kelimenin istenilen bir dile çevrilebildiği ve metinlerin uygulama içerisinde okutulabileceği bir uygulama hazırlama

Tasarla: Bir sözlük uygulamasının nasıl olması gerektiği ile ilgili ekran tasarımı hazırlama

Üret: Tasarla bölümünde görsel tasarımı hazırlanan sözlük uygulamasının kod bloklarının hazırlanarak Emulator ya da AI Companion yardımı ile test edilmesi

Değerlendir: Hazırlanan sözlük uygulamalarının değerlendirilmesi ve proje geliştirme süreci ile ilgili ara değerlendirme yapılması

1. Gözle

Bir önceki haftalarda kullanmaya başladığımız App Inventor ile uygulama geliştirme sürecinde öğrencilere aşağıdaki sorular yöneltilir.

- Kullandığınız uygulamalarda kullanılan medya elemanları nedir?
- Bir uygulamada görseller mi yoksa yazılar mı daha çok dikkat çekmektedir?
- Kullandığınız uygulamalarda ses olmasını mı olmamasını mı tercih edersiniz?
- Uygulama geliştirirken işlemlerin her zaman değil de belli durumlarda çalışmasını sağlamak için ne yapmalıyız?
- Bir uygulama geliştirirken bir işlemin birden çok kez yapılması gereken durumlarla karşılaşabilir miyiz, bu durumlarda ne yapmalıyız?
- Kullandığınız bir uygulama ekrana yazdığınız metinleri okuyabiliyorsa bunu nasıl gerçekleştiriyor olabilir?
- Kullandığınız Türkçe-İngilizce sözlük uygulamaları internet olmadan da çalışabiliyor mu?

Sorular hakkında öğrencilerin tartışmaları istenir ve tartışmanın ilerlemediği durumlarda eğitimci konuyu yönlendirir. Eğitimci, konuyu yönlendirmede aşağıdaki bilgilerden yararlanabilir.

1.1. App Inventor Medya Kullanımı

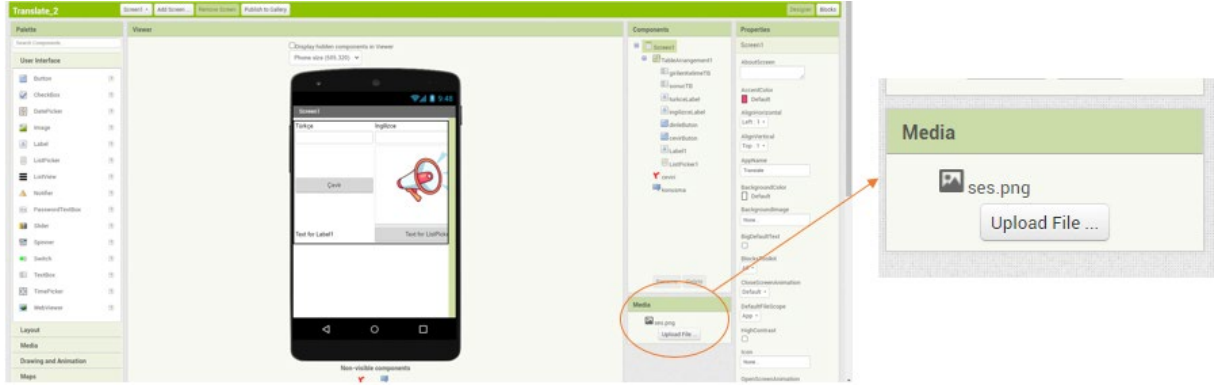
App Inventor ile geliştirdiğiniz uygulamalara ses, fotoğraf ve video gibi medyalar ekleyebilirsiniz. Bu eklediğiniz medyalar uygulamanızın hem daha kullanıcı dostu hem de daha ilgi çekici olmasını sağlayacaktır.

Uygulamada medya kullanımı; hedef kitle göz önünde bulundurularak onlara uygun şekilde yapılmalıdır. Örnek vermek gerekirse 18 yaş altı çocuklar için hazırlanan uygulamalar ile yetişkinler için hazırlanan uygulamalarda kullanılan görseller de sesler de videolar da farklılıklar gösterir. Bu sebeple öncelikle uygulamanın hedef kitlesi belirlenmeli ve medya kullanımı buna göre yapılmalıdır.

Ayrıca uygulamanızın hedef kitlesinde engelli bireylerin de bulunduğu durumlarda uygulamanızın erişilebilirlik yönünden daha etkin olması için medya kullanımı uygulamanızı zenginleştirecektir. Örneğin görme engelliler için hazırlanan bir uygulamada ses kullanımı olmazsa olmaz duruma gelmektedir. Ya da işitme engelli bireyler hazırlanan uygulamalarda kullanılan videolarda altyazı olması veya betimleme bulunmasına özen göstermek gerekebilir.

1.1.1. Medya Dosyalarını Sunucuya Yükleme

App Inventor'da medya kullanımı için medya dosyalarımızı öncelikle çalışmamıza dahil etmek için sunucuya yüklememiz gerekmektedir.



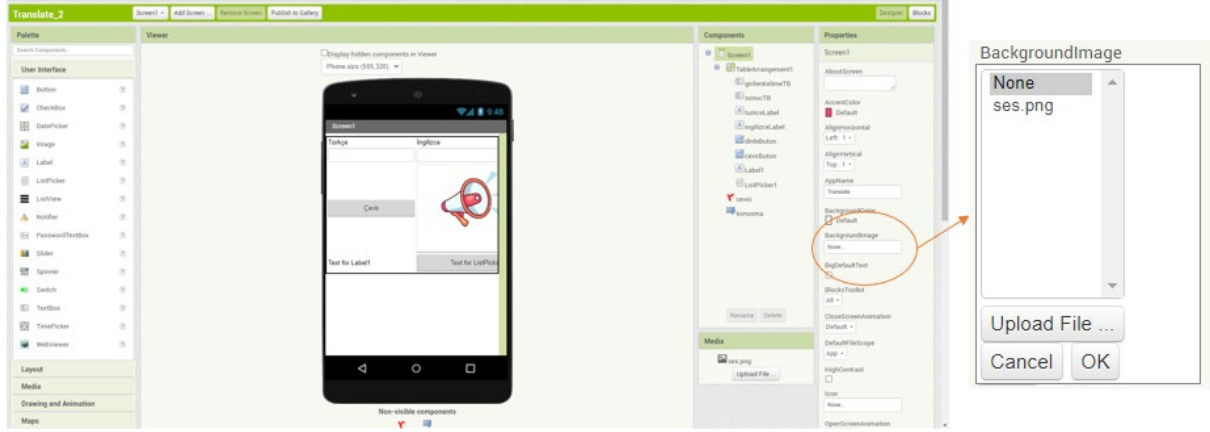
Ekran Görüntüsü 18. Medya Ekleme Modülü

Yukarıdaki görselde de görüldüğü üzere “Media” bölümünden çalışmamıza ses, görüntü veya video dosyaları ekleyebilirsiniz. Ses dosyaları için; wav, wma, mp3 gibi bilinen ses dosyalarının tümünü, görüntü dosyaları için; jpg, png gibi bilinen görüntü dosyalarının tümünü, video dosyaları için; mp4, 3gp, avi, webm gibi bilinen video dosyalarının tümünü kullanabilirsiniz. Eklediğiniz bütün dosyalar için 5 MB dosya limiti bulunmaktadır. Yani 5 MB’dan büyük dosyaları eklemek için öncelikle boyutlarını düşürmeniz gerekmektedir. Dosya boyutunu küçültmek için video dosyalarında çözünürlüğü düşürebilir ya da video dosyasının belli bir bölümünü kesip o bölümü kullanabilirsiniz, görüntü dosyalarında çözünürlük düşürebilir ya da farklı sıkıştırma yöntemleri kullanan dosya türlerine çevirebilirsiniz, ses dosyalarında da ses kalitesini düşürebilir ya da sesin tamamını eklemek yerine belli bir bölümünü kesip o bölümünü ekleyebilirsiniz.

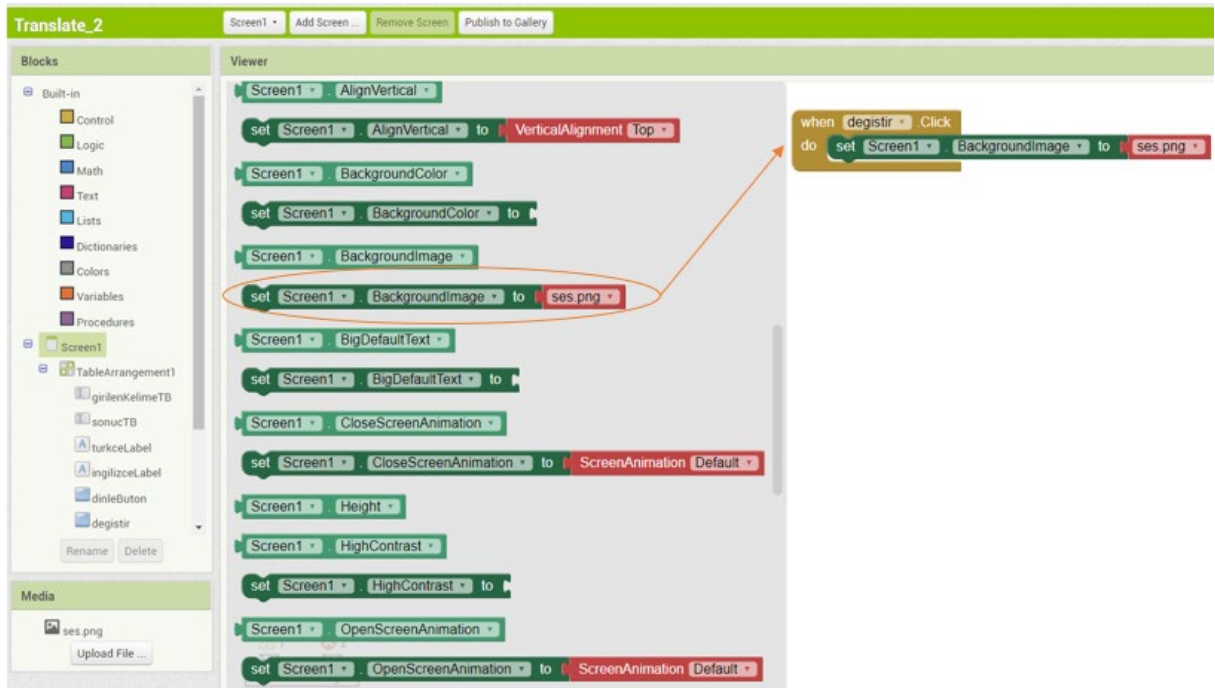
1.1.2. Görüntü Dosyalarını Kullanma

Uygulamamızda görüntü dosyalarını kullanmak için App Inventor içerisinde birden çok yöntem bulunmaktadır. Bunlardan biri uygulamanın arka planı olarak bir görsel kullanmakken, bir diğeri görüntüyü göstermek için bir “Image” modülü kullanmaktır. Bir diğer yöntem ise etkileşime girebilecek bir görüntü dosyası kullanmak için bir butonun arka planı olarak görsel kullanmaktır. Birçok görüntü formatı desteklense de “.png” ve “.jpeg” dosya formatları önerilmektedir.

Uygulamamızın arka planı olarak bir görsel kullanmak istiyorsak yapabileceğimiz iki farklı yöntem bulunmaktadır. Bunlardan biri “Designer” sekmesini kullanmak diğeri ise “Blocks” sekmesini kullanmaktır. “Designer” sekmesinde “Components” bölümünde istediğimiz ekranı seçtikten sonra sağ tarafta bulunan “Properties” bölümünden “BackgroundImage” ayarı bulunarak daha önce sunucuya yüklediğimiz görüntü dosyalarından birini seçebiliriz. Diğer yöntem ise “Blocks” sekmesinden istediğimiz ekranın bölümü altındaki “set (ilgili ekran) BackgroundImage to (istenen görüntü)” kod bloğu kullanılabilir. Bu örnekleri aşağıda görebilirsiniz.

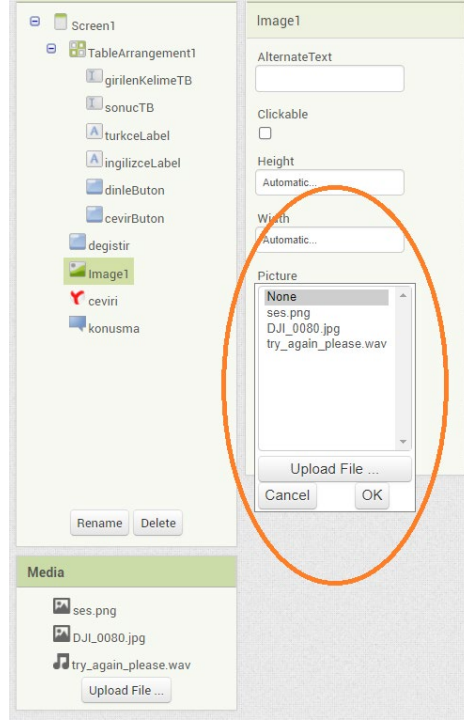


Ekran Görüntüsü 19. Designer sekmesinden arka plan ayarlama



Ekran Görüntüsü 20. Blocks sekmesinden arka plan ayarlama

Uygulamamızda “Image” modülünü kullanarak görüntü kullanmak için de yine ya “Designer” sekmesini ya da “Blocks” sekmesini kullanabiliriz. “Designer” sekmesinde “Palette” bölümündeki “User Interface” altında “Image” nesnesini çalışmamıza ekledikten sonra “Components” bölümünden bu “Image” nesnesini seçip “Properties” bölümünden “Picture” ayarına gelerek önceden sunucuya yüklediğimiz görüntülerden birini seçerek kullanabiliriz. “Blocks” sekmesinden bu “Image” nesnesini yönetebilmek için ise yine sol taraftan “Image” nesnesi seçilir ve buradaki kod blokları kullanarak görüntü ile ilgili istenen değişiklikler yapılabilir. Bu örnekleri aşağıda görebilirsiniz.



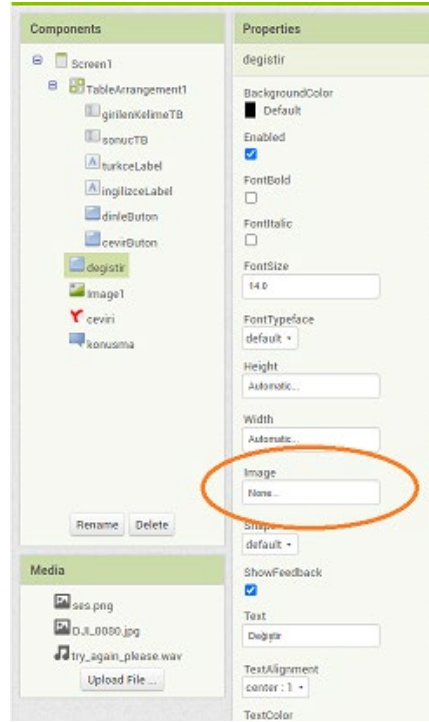
Ekran Görüntüsü 21.Designer sekmesinden resim ekleme



Ekran Görüntüsü 22.Blocks sekmesinden resim ekleme ve yönetme

Uygulamamıza eklediğimiz butonların arka planlarına görüntü ekleyerek tıklanabilir görüntüler oluşturabiliriz. Bunu yapmak için yine iki yöntemimiz bulunmaktadır. Bunlardan biri “Designer” sekmesi diğeri de “Blocks” sekmesidir. “Designer” sekmesinden eklediğimiz butonu “Components” bölümünden seçip “Properties” bölümünden “Image” ayarına gelerek daha önce sunucuya yüklenen resimlerden istenilen bir tanesini seçilir. “Blocks” sekmesinden ise arka planı

değiştirilmek istenen buton seçilerek buradaki kod bloklarından “Image” özelliğini içeren kod bloğu kullanılabilir. Bu örnekleri aşağıda görebilirsiniz.



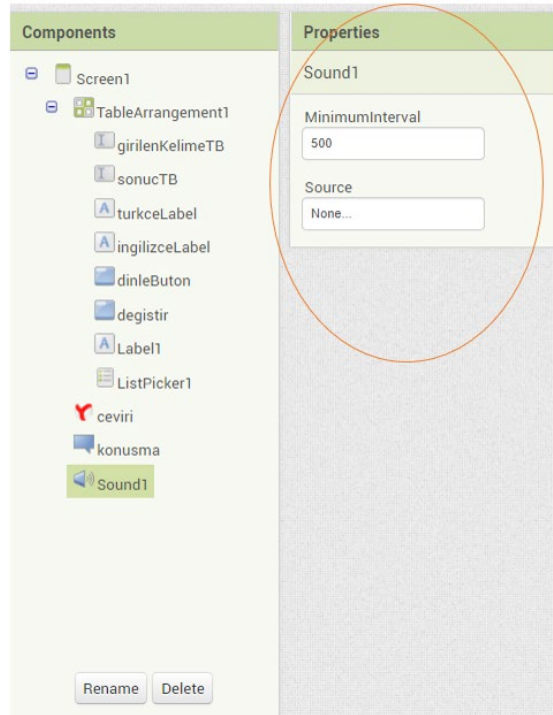
Ekran Görüntüsü 23. Designer sekmesinden buton arka plan resmi ekleme



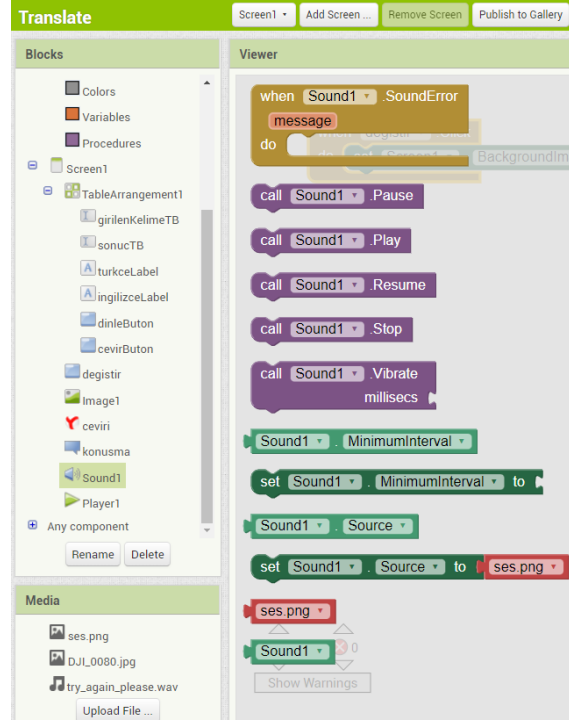
Ekran Görüntüsü 24. Blocks sekmesinden buton arka plan resmi ekleme

1.1.3. Ses Dosyalarını Kullanma

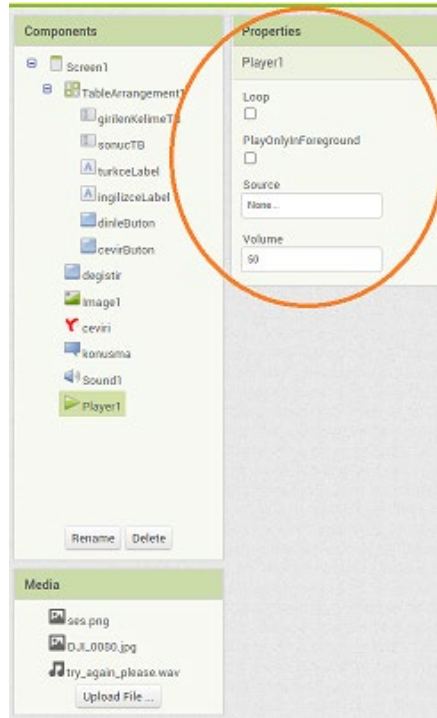
App Inventor içerisinde kullanabileceğimiz ses dosyaları iki farklı kategoriye ayrılmıştır. Bunlardan bir tanesi kısa sesler (bildirim sesleri, uyarı sesleri gibi), bir diğeri ise daha uzun ses dosyalarıdır. (Müzikler, arka plan ses dosyaları gibi). Bu iki farklı ses dosyalarının kullanımı için kullanacağımız modüller de farklılık göstermektedir. Kısa ses dosyalarını kullanmak için “Palette” sekmesindeki “Media” altında bulunan “Sound” nesnesini, uzun ses dosyaları için ise yine aynı yerde bulunan “Player” nesnesini kullanabiliriz. Bu iki nesne de uygulamamıza eklediğimizde ekranda görünmeyecek gizli nesnelerdir. Uygulamamıza bu nesneleri ekledikten sonra ister “Designer” sekmesinden istersek de “Blocks” sekmesinden gerekli ayarlamaları yapabiliriz. Birçok ses dosyası türü desteklense de “.wav” veya “.mp3” dosya formatları önerilmektedir.



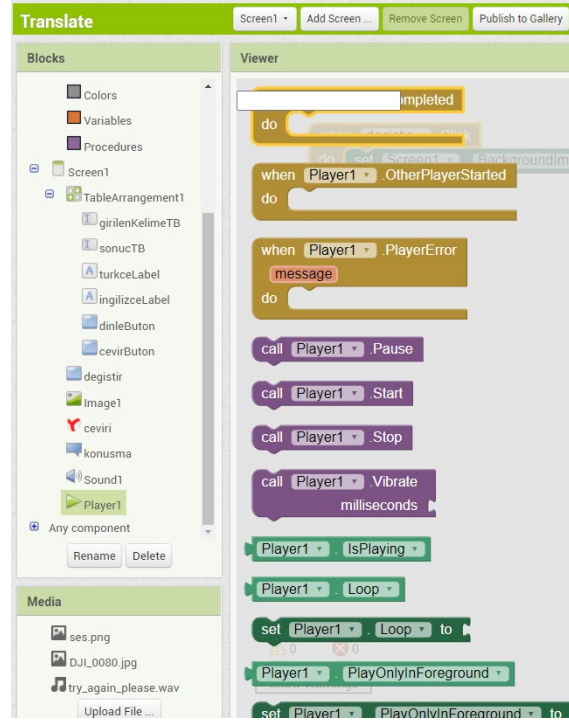
Ekran Görüntüsü 25. Designer sekmesinden sound nesnesi ekleme



Ekran Görüntüsü 26. Blocks sekmesinden sound nesnesi ekleme ve yönetme

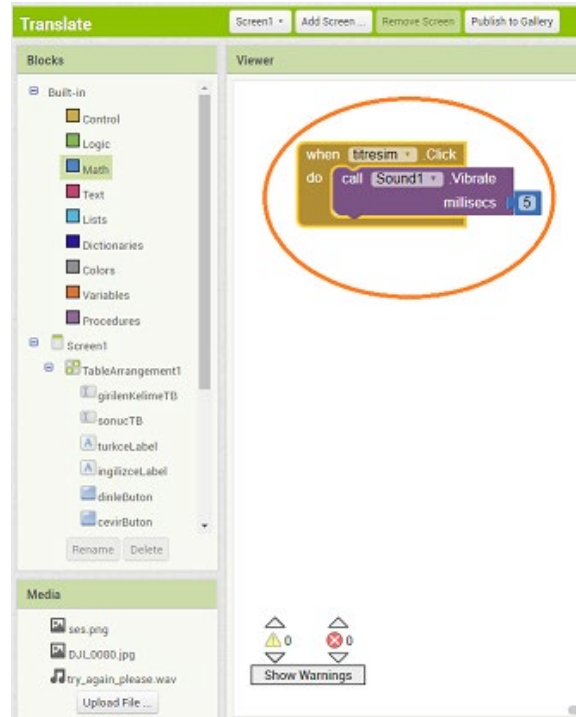


Ekran Görüntüsü 27. Designer sekmesinden player nesnesi ekleme



Ekran Görüntüsü 28. Blocks sekmesinden player nesnesi ekleme ve yönetme

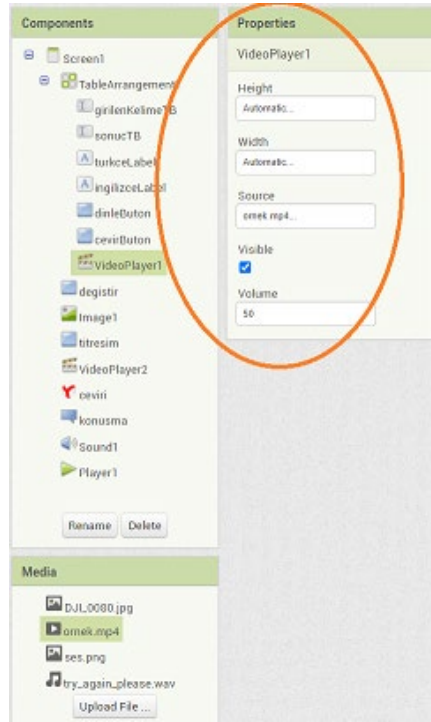
Uygulamamızda ses dosyalarını kullanırken mobil cihazımızın ses çalması dışında eğer titreşim özelliği bulunuyorsa titremesini de sağlayabiliriz. Bunu yapabilmek için “Blocks” sekmesinde bulunan “call (ilgili ses dosyası) .Vibrate milliseccs” kod bloğu kullanılabilir. Bu kod bloğu hem “Sound” hem de “Player” nesnesinde bulunmaktadır. Örnek kullanım aşağıda görülmektedir.



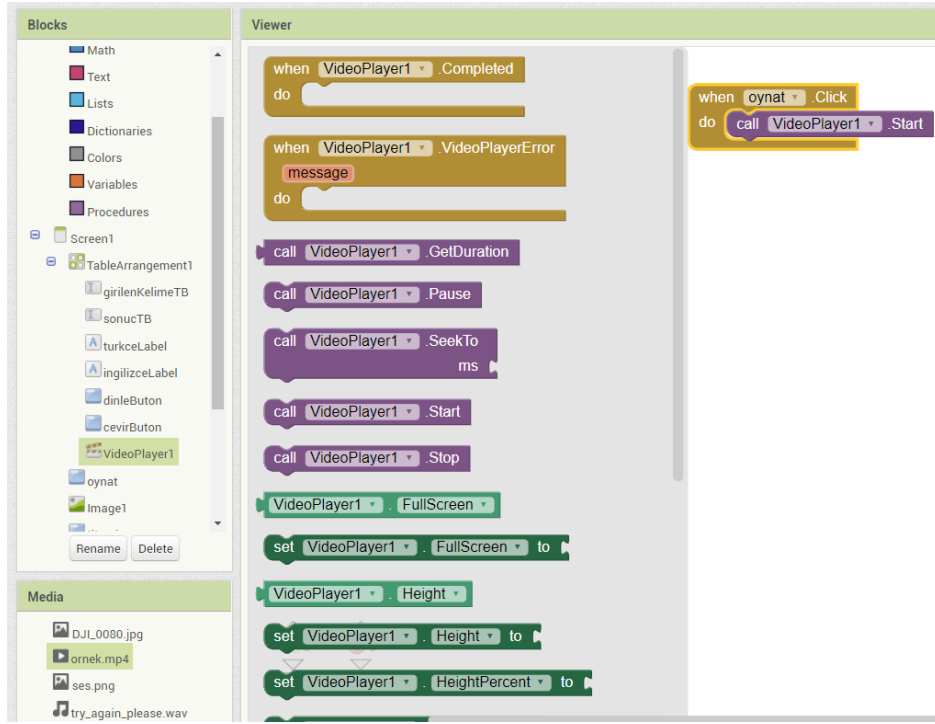
Ekran Görüntüsü 29. Blocks sekmesinden titreşim ekleme ve yönetme

1.1.4. Video Dosyalarını Kullanma

App Inventor ile geliştirdiğimiz uygulamamızda video dosyalarını kullanmak istiyorsak öncelikle çalışmamıza bu video dosyasını eklemiş olmamız gerekiyor. Sonrasında uygulamamıza “Media” bölümü altındaki “VideoPlayer” nesnesini eklemeliyiz. “VideoPlayer” nesnesini kullanmak için iki farklı yöntemimiz bulunmaktadır. Bunlardan birincisi “Designer” sekmesinde eklediğimiz “VideoPlayer” nesnesini tıkladıktan sonra sağ taraftaki “Properties” sekmesinden gerekli ayarlamaları yapmak, diğer yöntem ise “Blocks” sekmesinde ilgili kod bloklarını kullanarak video ayarlamaları yapmaktır. Birçok video dosyası türü desteklense de “.wmv”, “.mp4” veya “.3gp” dosya formatları önerilmektedir. Örnek kullanımları aşağıda görebilirsiniz.



Ekran Görüntüsü 30. Designer sekmesinden videoplayer nesnesi ekleme



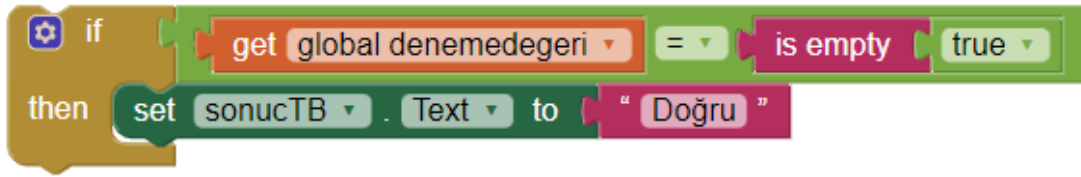
Ekran Görüntüsü 31. Blocks sekmesinden videoplayer nesnesi ekleme ve yönetme

1.2. Koşul, Döngü ve İç içe Döngüler

Uygulamanızın kodlama kısmını hazırlarken “Blocks” ekranından yararlanılır. Bir uygulamanın kodlanmasında en çok yararlandığımız durumlar koşullar, döngüler ve iç içe döngülerdir. Uygulamanın bir durum karşısında karar vermesini ve bu karara göre farklı işlemler yapmasını sağlamak için koşul yapılarına ihtiyacımız vardır. Bunun yanında uygulama içerisinden bir işlemi birden çok gerçekleştirmek istediğimiz zamanlarda döngülerden ve iç içe döngülerden yararlanarak uygulamamızı kodlayabiliriz.

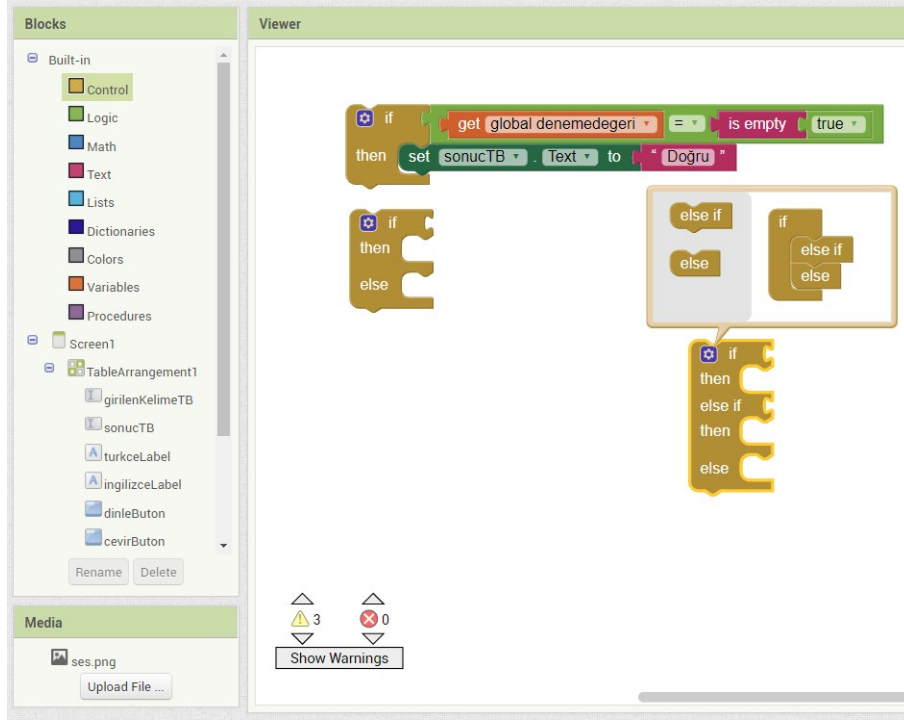
1.2.1. Koşul İfadeleri

Uygulamamızda bir durumun sonucunda bir eylemin yapılması şeklinde oluşturduğumuz yapılarla uygulamanın çalışmasını sağlayabiliriz. Bu yapıyı “if then” kod bloğu ile düzenleyebiliriz. “If then” bloğunu çalışmamıza “Control” bölümü altından ekleyebiliriz. Örnek vermek istersek belirlediğimiz bir değişkenin aldığı değere göre yapacağımız işlemi belirleyebiliriz, ya da bir textbox’a yazılan değer ne olduğuna göre uygulamamızın izleyeceği yolu belirleyebiliriz. Aşağıda örnek bir kullanımı görebilirsiniz.



Ekran Görüntüsü 32. Örnek koşul (if) yapısı

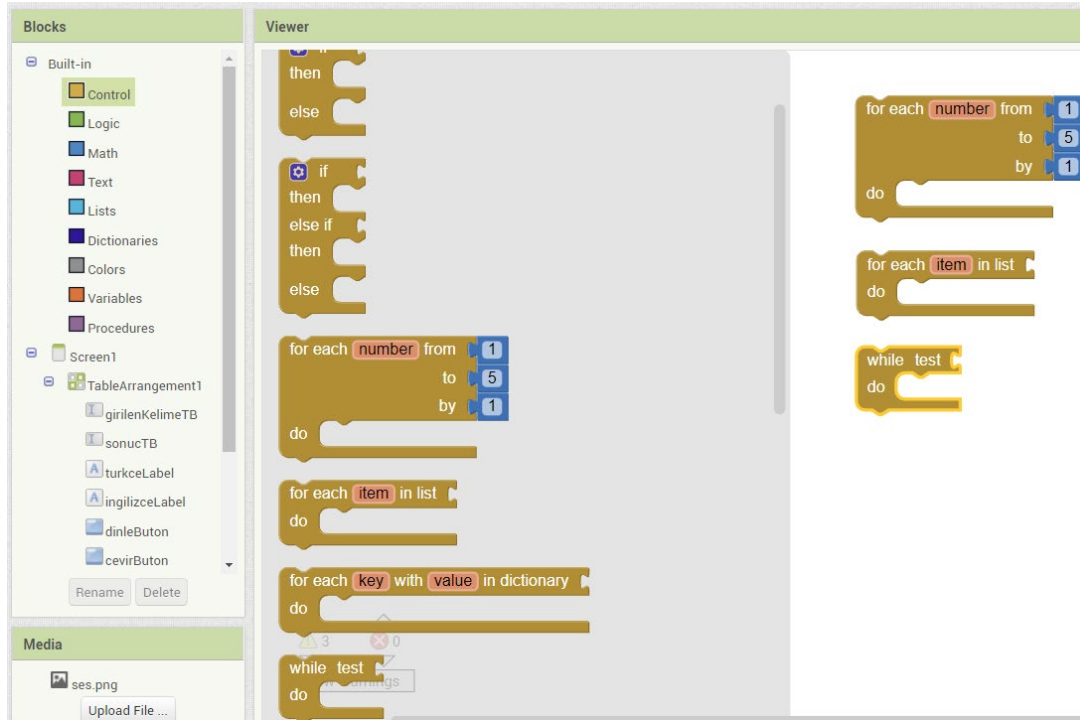
Koşul yapısı oluştururken gerekli olan yapıya göre “if then” yapısı düzenlenebilir. Örneğin bir koşulun gerçekleşmesi durumunda yapılacak eylemi belirtmenin yanında koşulun gerçekleşmemesi durumu için de bir eylem belirtmek istiyorsak “it then else” yapısı kullanılabilir. Bu ayarlamalar eklenen “if then” bloğu üzerinde sol üst köşede bulunan ayarlar (çark şeklinde) butonuna tıklanarak yapılabilir. Örnek kullanımları aşağıda görebilirsiniz.



Ekran Görüntüsü 33. Örnek koşul (if, else, then) yapısı

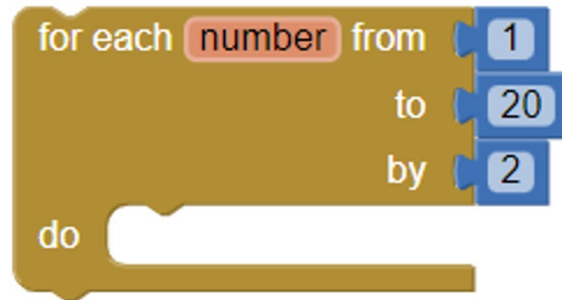
1.2.2. Döngü ve İç İçe Döngüler

Bir uygulama geliştirirken gerçekleşmesini istediğimiz bir eylemin birden çok kez gerçekleşmesini istiyorsak oluşturmamız gereken yapıya döngü yapısı adı verilmektedir. Döngü yapıları hazırlanan programın daha verimli çalışmasını amaçlamaktadır. Daha az kod yazılarak daha fazla işlem yapılması amaçlanmaktadır. Yapılacak işlemler bir şablon içerisinde tekrarlıyorsa döngülerden yararlanırız. Döngüler App Inventor’da birden çok yöntemle oluşturulabilir. En sık kullanılan döngü yapıları; “for each” ve “while” yapıları olarak öne çıkmaktadır. Bu yapıları kullanmak için “Blocks” sekmesi altında “Control” bölümünden ilgili kod bloklarını kullanabiliriz.



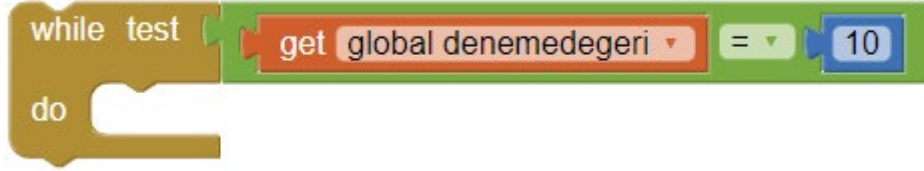
Ekran Görüntüsü 34. Control bölümü kod örnekleri

“for each” döngüsü belirlenen iki sayı aralığında verilen değer kadar değişkeni artırarak oluşturulan döngüler oluşturulmasını sağlar. Örnek vermek istersek; 1’den başlayan ve 20’değerine kadar 2’şer 2’şer bütün değerler için döngü içerisinde verilen eylemi yerine getiren bir yapı oluşturmak istediğimizde aşağıdaki yapı kullanabiliriz.



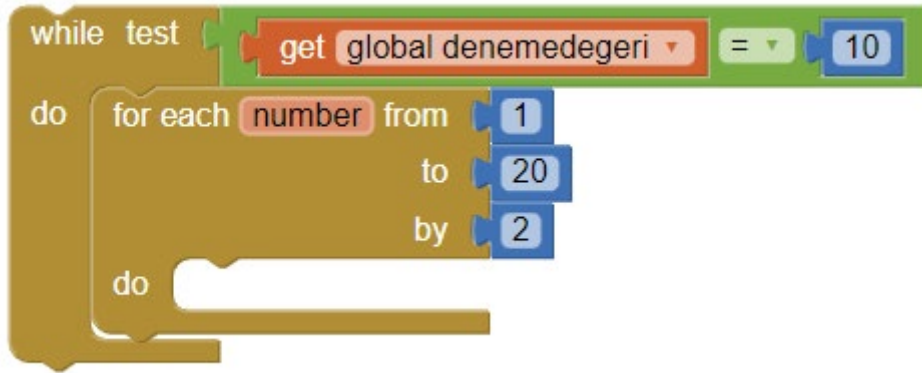
Ekran Görüntüsü 35. Örnek döngü (for each) yapısı

Belirlenen bir koşulun gerçekleşme durumuna göre kod bloğu içerisinde bulunan eylemin sürekli gerçekleşmesini istediğimiz durumlarda ise “while” döngü bloğu kullanılabilir. Örnek vermek istenirse; döngü içerisinde değeri teker teker artırılan bir değişkenin değerinin 10 olmasına kadar kod bloğu içerisinde belirtilecek işlemi tekrarlayan kod yapısını aşağıda görebilirsiniz.



Ekran Görüntüsü 36. Değişkenin değeri 10 olana kadar devam edecek bir döngü yapısı

Oluşturmak istediğimiz döngü yapısı hem bir koşula dayalı hem de belli bir aralıkta sürekli tekrarlınsın istiyorsak oluşturabileceğimiz döngü kod bloğu ise iç içe döngülere bir örnek olarak verilebilir. Bu örneği aşağıda görebilirsiniz.



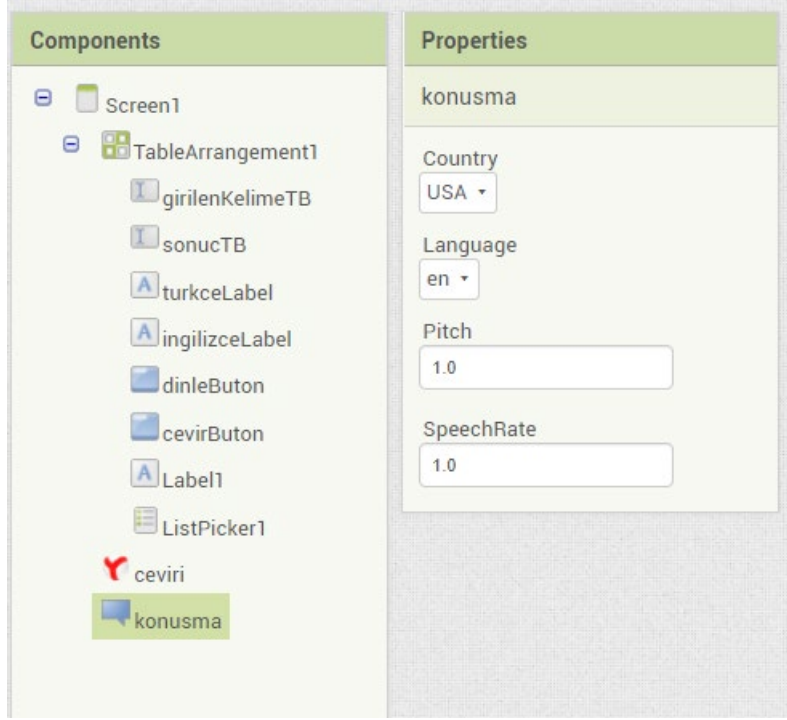
Ekran Görüntüsü 37. İç içe döngü yapısı

1.3. Metin Okuma

Uygulamanıza ses dosyası eklemeyi daha önce öğrenmiştik. Bir metnin okunması ile ilgili bir ses kaydı kullanmak istiyorsanız bunu uygulamanıza ses dosyası ekleyerek yapabilirsiniz. Fakat örneğin ekrana kullanıcı tarafından girilen bir metin seslendirilmesini ya da uygulama içerisinde sonuç olarak bir metnin okunmasını istiyorsanız bunun için uygulamanıza yeni bir nesne eklemeniz gerekmektedir. “Designer” sekmesi altındaki “Media” bölümünde bulunan “TextToSpeech” nesnesi bu işlemi bizim için yapmaktadır.

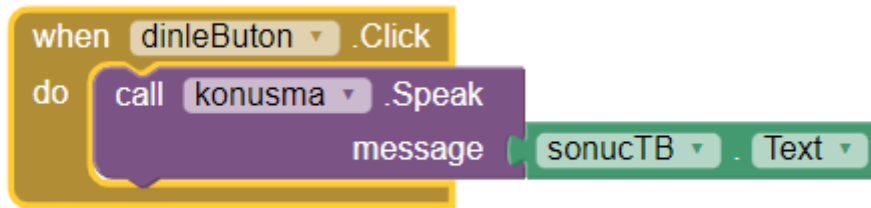
“TextToSpeech” nesne çalışabilmek için uygulamanın çalıştığı cihazın aktif bir internet bağlantısının bulunmasına ihtiyaç duymaktadır. Seslendirilmesini istediğiniz metni “Blocks” sekmesinde kod blokları ile belirleyebilir ve okunmasını sağlayabilirsiniz. Fakat bu nesne sadece İngilizce, Almanca, Fransızca, İspanyolca ve İtalyanca dillerinde seslendirme yapabilmektedir. Bu diller dışında bulunan dilleri seslendirirken istenilen dilin kurallarına hâkim olmadığı için kelimeleri yanlış telaffuz edebilmektedir.

Uygulamamıza eklediğimiz “TextToSpeech” nesnesi ile ilgili ayarları “Designer” sekmesinde “Properties” bölümünden yapabiliriz. Bu ayarlar arasında ülke seçimi ve dil seçimi dışında ses ile ilgili ayarlar da bulunmaktadır. Ülke seçimi ilgili dilin seçilen dildeki aksanı ile okunmasını sağlayacaktır.



Ekran Görüntüsü 38. Designer sekmesinde TextToSpeech nesnesi ekleme

Uygulamamıza eklediğimiz “TextToSpeech” nesnesi gizli bir nesne olarak eklenmektedir, yani ekranda görünmemekte fakat “Properties” bölümünden düzenlenebilir ve “Blocks” sekmesinden kod blokları ile çalışması düzenlenebilmektedir. “Blocks” sekmesi altında bir butonun tıklanması sonucunda bir “TextBox” içerisindeki metni okuyan kod bloğu aşağıda örnek olarak verilmiştir.

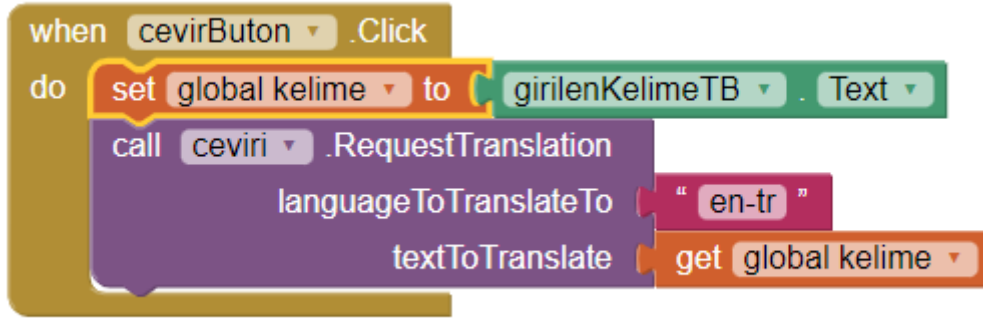


Ekran Görüntüsü 39. Metin kutusu içindeki metni seslendiren kod yapısı

1.4. Çevrimiçi Çeviri

App Inventor’da hazırladığınız bir uygulamada farklı diller arasında çeviri işlemi yapabilirsiniz. Bu işlemi yapabilmek için çalışmanıza “Designer” sekmesinde “Media” bölümündeki “YandexTranslate” nesnesini eklemeniz gerekmektedir. Bu nesne gizli bir nesne olarak çalışmamıza eklenir yani ekranda görünmez.

Adından da anlaşılacağı gibi bu nesne çalışabilmek için Yandex servislerini kullanmaktadır, yani çalışmak için uygulamanın kullanıldığı cihazın aktif bir internet bağlantısı olması gerekmektedir. Yandex servisleri birçok dili desteklemektedir. Desteklenen dillere ulaşmak için Yandex’in ilgili internet bağlantısı kullanılabilir (<https://yandex.com/dev/translate/doc/dg/concepts/api-overview.html>). Çeviri yapılmak istenen dile ait dil kodunu “Blocks” sekmesinde kod bloklarımızda kullanarak çeviri işlemi istediğimiz şekilde ayarlanabilmektedir. Örnek olarak butona tıklandığı zaman “TextBox”da verilen Türkçe metni İngilizceye çeviren ve bunu belirtilen değişkene değer olarak atayan kod bloğu aşağıda verilmiştir.



Ekran Görüntüsü 40. Metin kutusu içindeki metni İngilizceye çeviren ve belirtilen metin kutusuna yazdıran kod yapısı

2. Uygulama

Öğrencilerin ekranda iki tane “TextBox”, iki tane “Label”, iki tane “Button” bulunan bir uygulama hazırlaması istenir. Uygulama, buton metni “Çevir” olan butona tıklandığı zaman ilk “TextBox”da bulunan Türkçe kelimeyi İngilizceye çevirip ikinci “TextBox”a yazacaktır. Eğer Türkçe kelimenin girilmesi gereken “TextBox” boş bırakıldıysa ve buton tıklandıysa ilgili “TextBox”ın içine uyarı olarak “Lütfen bir kelime giriniz” uyarısı verecektir. Diğer butonun ise arka plan resmi olarak bir hoparlör görseli kullanılarak hazırlanacaktır. Bu butona tıklandığında ise çevirisi yapılarak İngilizcesi ikinci “TextBox”a yazdırılan kelimeyi İngilizce dilinde ve ABD (Amerika Birleşik Devletleri) aksanında seslendirecektir.

Uygulamanın tamamlanmış proje dosyası ekler kısmında bulunabilir.

3. Tasarla

Öğrencilerin farklı dillerde çeviri yapılabilecek ve hangi diller arası çeviri yapılacağını seçebilecekleri bir uygulama tasarımları istenecektir. Bu aşamada ilgili uygulamanın görsel tasarımı yapılacak hangi nesnelerin kullanılacağı belirlenecektir.

4. Üret

Tasarla aşamasında görsel tasarımı hazırlanan uygulamanın kod blokları da hazırlanıp uygulamanın çalışır hale gelmesi sağlanacaktır.

5. Değerlendir

Üret aşamasında hazırlanan uygulama Emulator ya da AI Companion aracılığı ile çalıştırılacak ve test edilecektir. Öğrenciler kendi hazırladıkları uygulamaları diğer öğrencilerin de test edebilmelerini sağlayacaktır. Öğrencinin öz değerlendirme yapabilmesi için Öz Değerlendirme Formu – 1 (<https://forms.gle/jiofyNvaDj48fJ9P8>) öğrenciye uygulanır.

İkinci hafta başlanan proje geliştirme süreci ile ilgili Öz Değerlendirme Formu – 2'yi (<https://forms.gle/gq4b5NCLFCAYp2c78>) doldurması istenecektir. Bu rapor eğitmen tarafından da incelenip destek verilmesi gereken öğrenciler belirlenecektir.

4. Hafta: Sensörler ve Haberleşme

Ön Bilgi:

- Temel bilgisayar bilgisi
- Temel yazılım bilgisi

Haftanın Kazanımları:

- Sensör kavramını tanır.
- Sensör çeşitlerini tanır.
- Mobil cihazlardaki sensörleri tanır.
- Sensör kullanım amacını kavrar.
- App Inventor ile kullanılabilecek sensörleri öğrenir.
- App Inventor ile sensör kullanımını yapılandırabilir.
- Mobil cihazların birbirleri ile haberleşmesini kavrar.
- Bluetooth haberleşme kavramını kavrar.
- App Inventor ile mobil cihazların bluetooth ile haberleşmesini yapılandırabilir.
- App Inventor ile mobil cihazların elektronik devre kartları ile kablosuz iletişim kurmasını sağlar.

Haftanın Amacı:

App Inventor ile hazırlanan bir uygulamanın mobil cihazın sensörlerini kullanabilmesini ve bu sensörlerden gelen verilere göre işlemler yapabilmesini sağlar. App Inventor ile kullanabileceği sensörleri ve bu sensörlerin ne tür veriler ürettiğini anlar. App Inventor ile hazırlanan uygulamaların bluetooth haberleşme ile diğer mobil cihazlarla iletişim kurmasını sağlar. Mobil cihazların kablosuz olarak elektronik devre kartları ile iletişim kurabilmeleri için gerekli yapılandırmaları yapar.

Kullanılacak Malzemeler:

Bilgisayar, Mobil cihaz (cep telefonu ya da tablet), kalem, kâğıt

Haftanın İşlenişi:

Gözle: Sensör kavramı nedir, sensör türleri nelerdir, sensörler ne tür veriler üretirler, sensörlerin ürettiği veriler nasıl kullanılır ve App Inventor'da hangi sensörler kullanılabilir, mobil cihazlar birbirleri ile kablosuz olarak haberleşmesini sağlar, bluetooth modülü App Inventor ile nasıl kullanılır, elektronik devre kartı ile mobil cihazın kablosuz haberleşmesi ile elektronik devre kartında işlemler nasıl yapılır.

Uygula: Mobil cihazın bulunduğu konumu belirleyip bunu ekranda gösteren bir uygulama hazırlama, elektronik devre kartı üzerinde yerleştirilmiş farklı renklerdeki ledlerin App Inventor ile hazırlanan bir uygulama ile kablosuz şekilde kontrol edilebilecek bir uygulama hazırlama

Tasarla: App Inventor sensörlerini kullanabileceği ve farklı mobil cihazlarla kablosuz iletişim kurabilecek bir uygulama tasarımı hazırlama

Üret: Tasarla bölümünde tasarımı hazırlanan uygulamanın üretilmesi

Değerlendir: Üret bölümünde hazırlanan uygulamaların değerlendirilmesi ve proje geliştirme süreci ile ilgili ara değerlendirme yapılması

1. Gözle ve Uygula

Bir önceki haftalarda kullanmaya başladığımız App Inventor ile uygulama geliştirme sürecinde öğrencilere aşağıdaki sorular yöneltilir.

- Bir uygulamanın çalışması için gereken veriler sadece kullanıcılar tarafından mı oluşturulur?
- Mobil cihazlar veri üretebilir mi?
- Mobil cihazlar ne tür veriler üretebilirler? Kullandığınız mobil uygulamaları düşünün, nasıl veriler üretiyor ve kullanıyor olabilirler?
- Sensör nedir? Bildiğiniz sensör örnekleri verebilir misiniz?
- Mobil cihazlarda kullanılan harita uygulamaları nasıl çalışıyor olabilir?
- Mobil cihazlar herhangi bir kablo bağlantısı olmadan diğer cihazlara veri gönderebilir mi?
- Mobil cihazlar arasında iletilen veriler ile neler yapılabilir?
- Elektronik devre kartlarını kablosuz olarak yönetebilir miyiz?

Sorular hakkında öğrencilerin tartışmaları istenir ve tartışmanın ilerlemediği durumlarda öğretmen konuyu yönlendirir. Öğretmen, konuyu yönlendirmede aşağıdaki bilgilerden yararlanabilir.

1.1. Sensörler

İnsanlar içinde bulundukları ortamları algılamak için duyu organlarını kullanırlar. Örnek vermek gerekirse; bir çiçeği görmek için gözümüzü, kokusunu almak için burnumuzu kullanırız, dokusunu anlamak için elimizle dokunuruz. Bu örneği düşündüğümüz zaman gözümüz, burnumuz ve ellerimiz bizim çevreyi algılamak için kullandığımız organlarımızdır. Elektronik cihazların da çevreyi algılamak için buna benzer parçalara ihtiyaçları vardır. Bunlara genel olarak sensör adı verilmektedir.

Tanım yapmak istersek; sensör, çevreyi ve çevredeki değişiklikleri algılayıp elektronik cihazlara veri sağlayan donanım parçalarıdır diyebiliriz. Sensörler tek başlarına bir işlem yapmaz, sağladıkları verileri elektronik cihazlara ileterek cihazların çalışabilmesini sağlar. Bu yapıyı anlamak için bir nesneyi görmek için gözümüzün algıladıklarını beynimize iletmesini ve beynimizin o nesneyi algılamasını düşünebiliriz. Bu örnekte göz görme işlemini gerçekleştirse de bunu algılayan ve bu bilgiyi kullanabilir bir hale getiren beynimizdir. Kısaca özetlemek gerekirse sensörler, elektronik cihazların duyu organlarıdır diyebiliriz.

1.1.1. Sensör Çeşitleri

İnsanlarda görme işlemini gerçekleştirmek için göz, duyma işlemini gerçekleştirmek için kulak kullanıldığı gibi elektronik cihazlarda da farklı türdeki değişkenleri algılamak için farklı türlerde sensörler bulunmaktadır. Algılanan bütün değişkenler bir veri haline getirilir ve bu sayede elektronik cihaz tarafından yapılacak işlemlerde kullanılabilir.

Cep telefonları, fotoğraf makineleri ya da kameralar fotoğraf veya video çekmek için sensörlerini kullanırlar. Bir başka örnek olarak ortam sıcaklığını ölçebilen elektronik cihazlar (Örnek: termometre, ateş ölçer) ısı sensöründen yardım alırlar.



Görsel 1 Cep telefonu kamerası



Görsel 2 Ateş Ölçer

Elektronik cihazların bulundukları ortamı algılaması için en çok kullanılan diğer sensör türlerine örnek vermek gerekirse; ışık sensörü, konum sensörü, manyetik alan sensörü, basınç sensörü ve ivme sensörü örneklerini verebiliriz.

1.1.2. Mobil Cihazlardaki Sensörler

Mobil cihazlarda en çok kullanılan sensörleri incelemek hazırlayacağımız uygulamalarda hangi sensörleri kullanmamız konusunda bize yardımcı olacaktır. En yaygın olarak bulunan sensörleri kullanacağımız uygulamalar, birçok cihazda sorun yaşamadan çalışacaktır. Eğer hazırladığımız bir uygulamada yararlanmayı amaçladığımız sensör uygulamanın yüklendiği mobil cihazda bulunmuyorsa uygulamamız ya hata verecek ya çalışmayacak ya da hatalı çalışacaktır.

Günümüzdeki mobil cihazlardaki sensörlerden bazıları; ivme ölçer, jiroskop, manyetik alan sensörü, küresel konumlandırma sistemi (GPS), yakınlık sensörü, ortam ışığı sensörü ve barkod okuyucu sensördür.

1.1.2.1. İvme Ölçer

İvme ölçer, mobil cihazın dikey mi yatay mı olduğunu, herhangi bir dönme ya da sarsılma eylemi gerçekleştiriyor mu, titriyor mu ya da doğrusal bir yöndekini hızının ne kadar olduğunu algılar. Örnek vermek gerekirse cep telefonlarının yatay döndüğünde ekranın yatay moda, dikey döndürüldüğünde dikey moda çevrilmesi bu sensörün yardımı ile gerçekleşmektedir.

1.1.2.2. Jiroskop

Jiroskop, ivme ölçerden farklı olarak sadece dikey yatay değil bütün eksenlerdeki döndürmeleri daha hassas bir şekilde algılayabilir. Örneğin bazı oyunlarda yönlendirme yapılırken telefonu hareket ettirmemiz istenmektedir, bu telefon hareketini algılayıp oyundaki karakterin hareket etmesini sağlayan bu sensördür.

1.1.2.3. Manyetik Alan Sensörü

Mobil cihazlardaki özellikle harita uygulamalarının sorunsuz çalışmak için bu sensöre ihtiyaç duymaktadırlar. Pusulaların da çalışma mantığı olan dünyanın manyetik alanını algılayıp ona göre dünya üzerindeki konumu algılama işlemi, mobil cihazlarda bu sensör yardımı ile gerçekleştirilir.

1.1.2.4. Küresel Konumlandırma Sistemi (GPS)

Küresel konumlandırma sistemi, dünya çevresinde dönen uydular yardımı ile mobil cihazından konumunun büyük bir hassasiyetle belirlenmesini sağlar. İnternet bağlantısına ihtiyaç duymadan bu uydularla iletişim kurup konumu koordinatlar şeklinde belirleyen sensör günümüzdeki neredeyse bütün mobil cihazlarda bulunmaktadır. Kullandığımız harita uygulamaları temelde bu sensör yardımı ile konumumuzu belirlerler.

1.1.2.5. Yakınlık Sensörü

Mobil cihaza herhangi bir nesne yaklaştığında ya da uzaklaştığında bunu algılayan sensördür. Genellikle bu işlemi kızılötesi bir ışık göndererek gerçekleştirirler. Cep telefonumuzla arama yaptığımızda telefonu kulağımıza yaklaştırdığımız zaman ekranın kapanmasını sağlayan veriyi üreten bu sensördür.

1.1.2.6. Ortam Işığı Sensörü

Mobil cihazın bulunduğu ortamın ışığını algılayan sensördür. Ortamdaki ışığın seviyesini algılar ve cihazın bu bilgi ile işlemler yapmasını sağlar. Örneğin karanlık ortamlarda cep telefonunun ışığının daha kısık olmasını ya da aydınlık ortamlarda ekranın daha iyi belli olması için ekran ışığının artırılması için gerekli olan veriyi üretir.

1.1.2.7. Barkod Okuyucu Sensör

Bu sensör mobil cihazın kamerasını kullanarak barkodları okumayı sağlar. Barkodlar, görsel olarak oluşturulmuş anlamsız gibi görünse de analog değerler içeren görüntülerdir. Bu sensör, kamera yardımı ile görünen barkodlardaki analog değerleri sayısal değerlere çevirerek işlem yapılmasını sağlar.

1.1.3. App Inventor ile Sensör Kullanımı

App Inventor ile geliştirilen uygulamalarda sensörlerden yararlanarak uygulamamızın ortamı algılamasını istiyorsak kullanmamız gereken modül "Sensors" modülüdür. "Sensors" modülü "Palette" sekmesinde bulunabilir.

Uygulamamızda sensörlerden yararlanmak istiyorsak öncelikle "Designer" sekmesinden uygulamamıza sensör nesnesini eklememiz gerekmektedir. Daha önceki haftalarda da deneyimlediğimiz gibi uygulamamıza eklemek istediğimiz nesneyi sürükleyip bırak yöntemi ile çalışmamıza ekleyebiliriz. Sensör nesnelerinin hepsi görünmez nesneler olarak uygulamamıza eklenirler, yani ekranda görünmez fakat ekranın alt kısmında "Non-visible Components" (Görünmeyen Nesneler) bölümünde görüneceklerdir. Bu şekilde uygulamamıza eklediğimiz sensör nesneleri ile ilgili yapılandırma ayarlarını "Blocks" sekmesinden gerçekleştirebiliriz.

Uygulamamıza eklediğimiz sensörleri kullanabilmek için uygulamanın çalıştırılacağı mobil cihazda kullandığımız sensörlerin bulunduğu dikkat etmeliyiz. Eğer mobil cihazda kullanmak istediğimiz sensör donanımsal olarak bulunmuyorsa uygulamamız çalışmayabilir ya da hatalı çalışabilir.

App Inventor içerisinde bulunan sensör nesnelerini kısaca açıklamak istersek;

1.1.3.1. AccelerometerSensor (İvme Ölçer Sensörü)

Uygulamamıza eklediğimizde mobil cihazımızın basit hareketlerini ve sallanmasını algılamamızı sağlar. 3 boyuttaki hareketi algılamamızı sağlar. Bu boyutlar; yatay(x), dikey(y) ve boylamsal(z) eksenidir. Koordinat sistemine göre düşünülecek eksenler mobil cihazın her yöne hareketini algılayabilir.

1.1.3.2. BarcodeScanner (Barkod Okuyucu)

Uygulamamıza "Designer" sekmesinden eklediğimiz barkod okuyucu ile mobil cihazımızın kamerası yardımı ile barkod verilerini okuyup bunlarla işlemler yapabilir. Yapacağımız işlemleri "Blocks" sekmesinden yapılandırabiliriz.

1.1.3.3. Barometer (Barometre)

Mobil cihazımızın hava basıncını algılamasını sağlayan bu sensörü uygulamamıza ekleyebilir ve bu ölçülen hava basıncı ile ilgili işlemler yapabiliriz. Fakat çoğu Android cihaz barometre sensörü içermediği için çalışabileceği cihaz sayısı sayılı olacaktır.

1.1.3.4. Clock (Saat)

Mobil cihazımızın saat bilgisini almamızı ve uygulama içerisinde bu bilgi ile ilgili işlemleri gerçekleştirebilmemiz için bu sensör nesnesini uygulamamıza eklememiz gerekmektedir.

1.1.3.5. GyroscopeSensor (Jiroskop Sensörü)

İvme ölçer sensörden farklı olarak mobil cihazımız 3 boyuttaki hareketini daha hassas bir şekilde ve hareketin açısal hızı gibi daha fazla ayrıntısını algılayabilecek bu sensörü uygulamamıza ekleyerek “Blocks” sekmesinden gerekli yapılandırmaları yapabiliriz.

1.1.3.6. Hygrometer (Nem Ölçer)

Bu sensör ile ortamdaki nem miktarını ölçüp bu değere göre işlemler yapabiliriz. Bu sensör de çoğu Android cihazda bulunmamaktadır.

1.1.3.7. LightSensor (Işık Sensörü)

Ortamdaki ışık miktarını ölçebilen bu sensör ile uygulamamızda ortam ışığına göre düzenlemeler yapabiliriz.

1.1.3.8. LocationSensor (Konum Sensörü)

Mobil cihazımızın özelliklerine göre GPS, Wi-fi ya da mobil veri bağlantısını kullanan bu sensör ile cihazımızın konumunu belirleyebilir ve bu konum bilgileri ile işlemler yapabiliriz. Konum bilgileri, koordinat sisteminde cihazın dünya üzerindeki yerini verebilmektedir. App Inventor içerisinde bulunan “Maps” bölümündeki nesneler ile bu konum bilgilerini birlikte kullanarak uygulamamıza bir görsel bir konum yardımcısı ekleyebiliriz.

1.1.3.9. MagneticFieldSensor (Manyetik Alan Sensörü)

Bu sensör yardımı ile mobil cihazın jeomanyetik konumunu öğrenebiliriz. Bu konum dünyanın manyetik alanından yararlanarak bulunduğunuz konuma göre kutupların hangi yönde olduğunu anlamaya yarayacaktır. Pusula gibi cihazlar bu sensörün algıladığı verilere göre çalışırlar.

1.1.3.10. NearField (Yakın Alan Sensörü)

Bu sensör NFC adıyla da bilinmekte olan teknolojiye kullanılmaktadır. Bütün mobil cihazlarda bulunmasa da bu sensörü içeren mobil cihaz sayısı hızla artmaktadır. Bir işaretleyici yardımı ile çalışan bu sensör barkod okuyucuya benzer işlem yapsa da kameradan yararlanmadan mobil cihaza yaklaştırılan işaretleyicideki verileri okuyarak işlem yapar. Örneğin belli bir işaretleyici algılayınca mobil cihazın sesinin kısılması, belli bir sesi çalması, Wi-fi özelliğini açması ya da kapatması gibi işlemleri yapmasını sağlayabiliriz.

1.1.3.11. OrientationSensor (Oryantasyon Sensörü)

Mobil cihazın hangi düzlemde hangi yönde olduğunu anlamak için kullanılır. İvme ölçer ya da jiroskop sensöründen en büyük farklılığı cihaz hareket etmiyorken hangi durumda olduğunu algılamasıdır. Örnek vermek gerekirse cihazın hangi tarafı daha yukarıda veya daha aşağıda, ya da cihaz kuzey- güney yönünden mi yoksa doğu – batı yönünde algılamak için bu sensör kullanılabilir.

1.1.3.12. Pedometer (Adım Sayar)

Bu sensör ivme ölçer sensörden de yararlanarak mobil cihazı taşıyan kişinin kaç adım attığını hesaplamak için kullanılabilir. Mobil cihazların çoğunda bulunan adım sayma özelliği bu sensör yardımı ile kullanılır. Uygulamamıza ekledikten sonra gerekli yapılandırma ayarlarını “Block” sekmesinden yapabiliriz.

1.1.3.13. ProximitySensor (Yakınlık Sensörü)

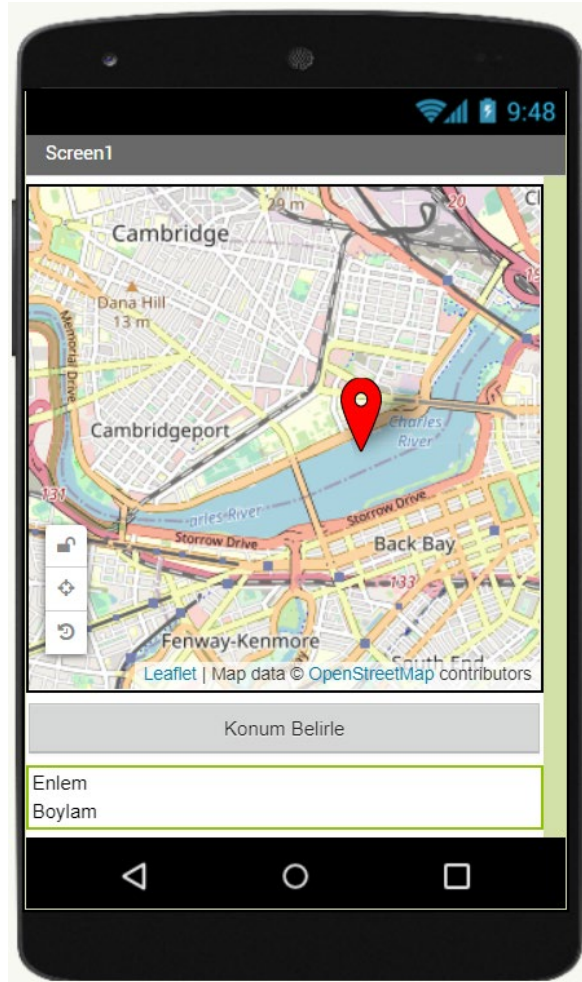
Mobil cihaza herhangi bir nesnenin yaklaşmasını algılamak için bu sensör kullanılır. Diğer sensör nesneleri gibi yine uygulamamıza “Designer” sekmesinden görünmeyen nesne olarak eklenir ve “Blocks” sekmesinden gerekli yapılandırma ayarları yapılır.

1.1.3.14. Thermometer (Termometre)

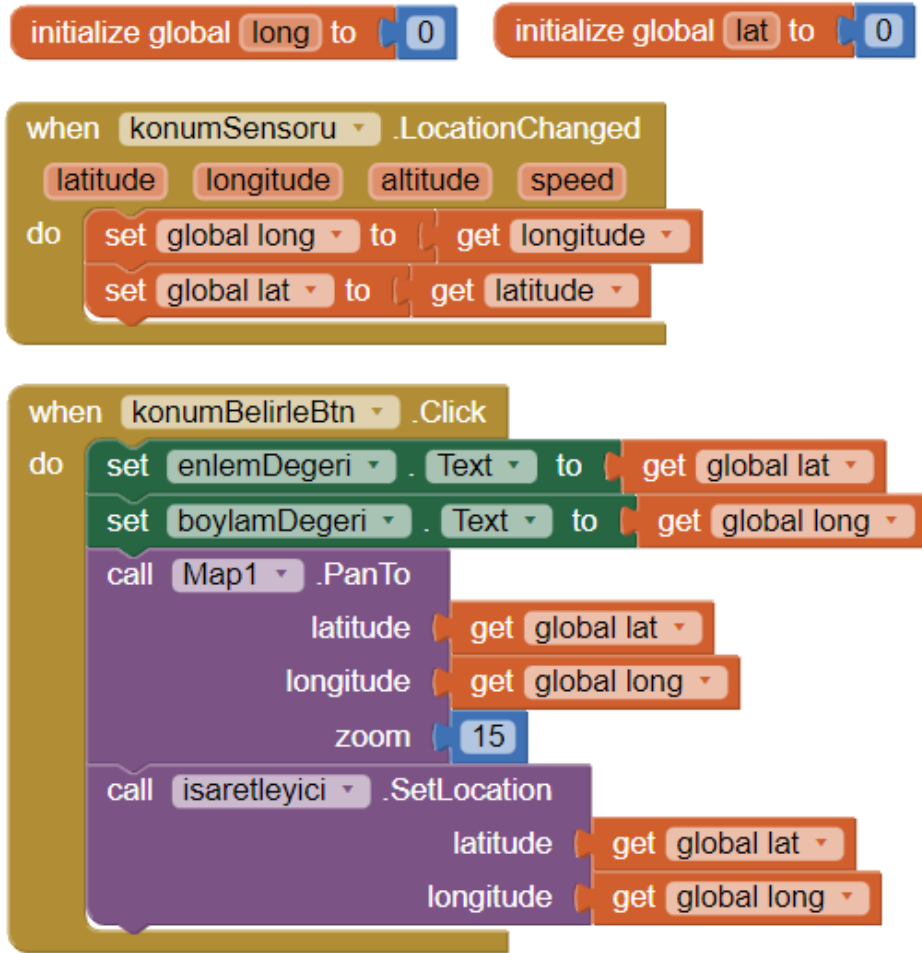
Bu sensör ortam sıcaklığını algılamak için kullanılır. Çoğu Android cihazda bulunmayan bu sensör uygulamanıza eklendiğinde ve bu donanımı içeren bir mobil cihazla birlikte kullanılırsa ortam sıcaklığını algılar ve uygulamanızın bu veri ile işlemler yapmasını sağlar.

1.1.4. App Inventor ile Konum Sensörü Uygulaması

Bu bölümde mobil cihazın o an bulunduğu konumu “LocationSensor” kullanarak belirleyen ve bu değerleri ekrana yazdıran, aynı zamanda “Maps” modülündeki “Map” nesnesinin de eklendiği ve “LocationSensor”den alınan verilerin “Marker” ile işaretlendiği bir uygulama geliştirelim.



Ekran Görüntüsü 41. Viewer(Görüntüleyici) Görüntüsü



Ekran Görüntüsü 42. Blocks Görüntüsü

Uygulamanın tamamlanmış proje dosyası ekler kısmında bulunabilir.

1.2. App Inventor ile Kablosuz Haberleşme

Mobil cihazların en büyük avantajlarının biri de çok farklı haberleşme teknolojilerini kullanarak birbirleri ile iletişim kurabilmeleridir. En sık kullanılan kablosuz haberleşme yöntemleri olarak web ve bluetooth haberleşme yöntemlerini sayabiliriz.

1.2.1. Web Kablosuz Haberleşme

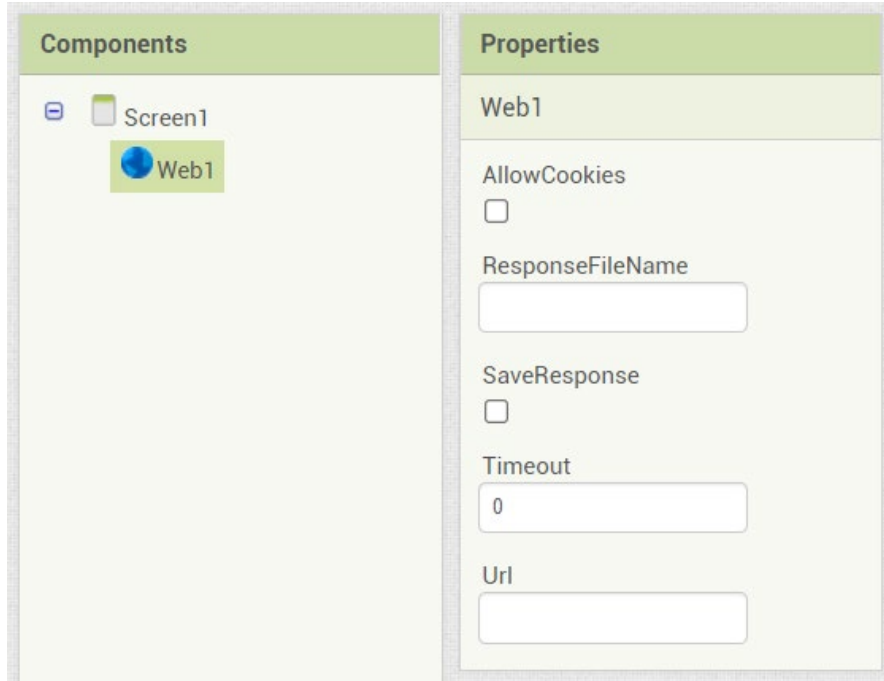
Mobil cihazların birbirleri ile haberleşmek için en yaygın olarak kullandıkları yöntemlerin başında web gelmektedir. Web, internet ile birbirine bağlı olan cihazlar arasında veri aktarılmasını sağlayan bir yöntemdir.

Web yardımı ile veri aktarırken kullanılan kuralların bütününe ise http adı verilmektedir. Http, iletilecek verilerin hangi türde, hangi yapıda olması ile ilgili kuralları belirler ve bu şekilde de veriyi alacak olan cihazın veriyi anlamasını sağlar.

Http protokolünde veri aktarımı için post, get, put, delete gibi yöntemler kullanılabilir. Post, veriyi karşı tarafa göndermek için, get karşı taraftan bir veriyi almak için, put veriyi karşı taraftaki bir konuma yerleştirmek için, delete konumu ise belli bir veriyi silmek için kullanılabilir.

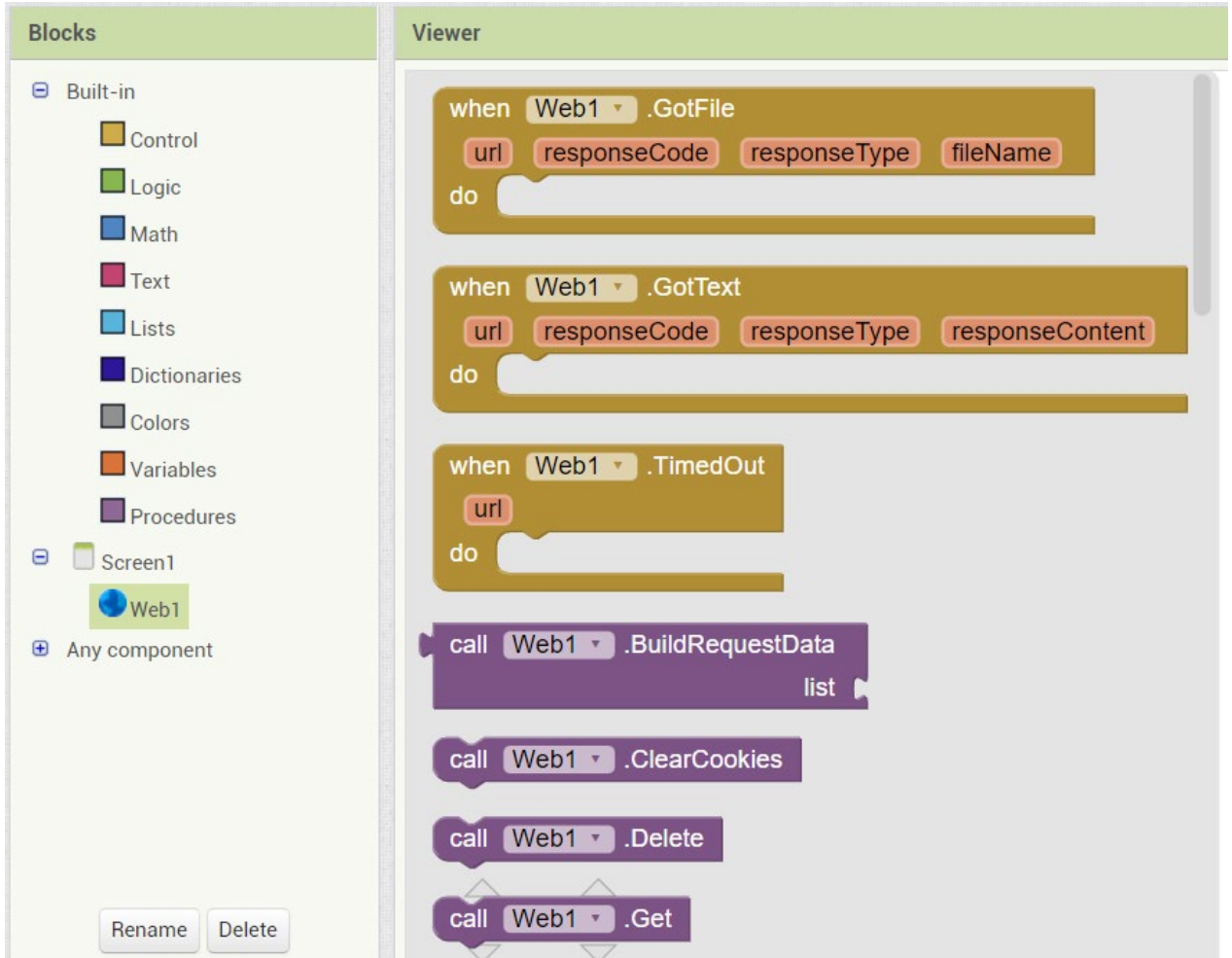
1.2.2. App Inventor ile Web Kablosuz Haberleşme

Uygulamamızın web haberleşme ile diğer cihazlarla haberleşmesini istiyorsak öncelikle çalışmamıza “Web” nesnesini eklememiz gerekmektedir. “Web” nesnesini, “Designer” sekmesindeyken “Palette” bölümündeki “Connectivity” modülünde bulabiliriz. Çalışmamıza görünmeyecek bir şekilde eklenecek olan “Web” nesnesi, ekranın alt tarafında görünmeyecek nesneler kısmına eklenecektir. Ayrıca “Properties” bölümünden nesne ile ilgili yapılandırma ayarları yapılabilmektedir.



Ekran Görüntüsü 43. Web Nesnesi Özellikler Bölümü

Uygulamamıza “Web” nesnesini ekledikten sonra gerekli yapılandırmaları “Blocks” sekmesinde yapmaya devam edebiliriz.



Ekran Görüntüsü 44. Web Nesnesi Blocks Sekmesi

1.2.3. Bluetooth Kablosuz Haberleşme

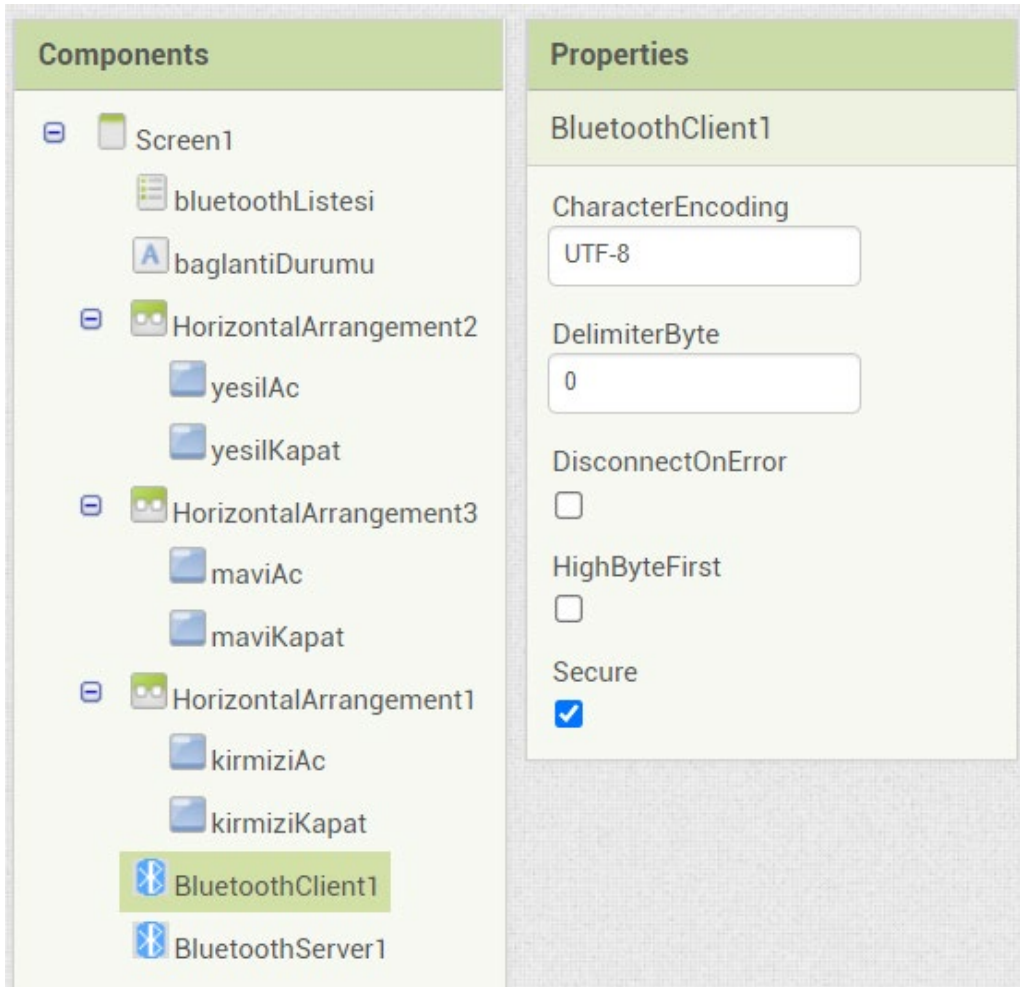
Günümüzde en çok kullanılan bir diğer kablosuz haberleşme yöntemi ise bluetooth'dur. Bluetooth, iletişim kurabilmek için cihazlar arasında fiziksel olarak bir yakınlığa ihtiyaç duymaktadır. Veri aktarımını gerçekleştirebilmek için Web haberleşme yönteminde cihazların internet bağlantısı olması yeterliken, bluetooth haberleşme sırasında cihazların ikisinin de bluetooth donanımlarının olması ve fiziksel olarak da birbirlerine yakın olması gerekmektedir.

Bluetooth haberleşme yönteminde cihazlar iki farklı şekilde çalışabilirler. Bunlardan birincisi sunucu (server) modu, diğeri istemci (client) modudur. Sunucu modunda, mobil cihaz diğer cihazlar tarafından bulunabilir ve bağlanılabilir durumdadır. İstemci modunda ise ortamdaki sunucu modundaki cihazları arayabilir ve onlara bağlanabilir durumdadır. Bluetooth ile veri aktarabilmek için cihazlar arasında bir bağlantı kurulmuş olması gerekmektedir. Bu bağlantının kurulabilmesi için de cihazlardan birinin sunucu, diğerin istemci modunda olması gerekmektedir.

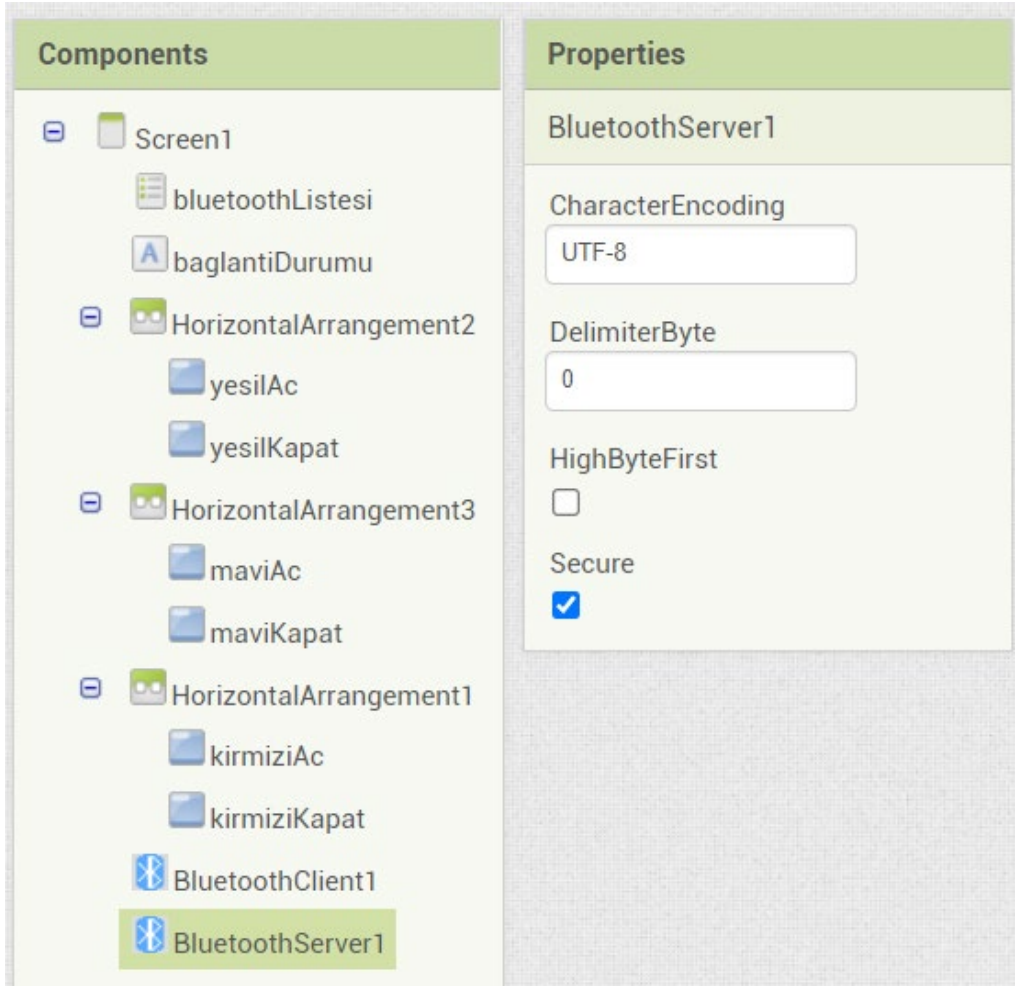
1.2.4. App Inventor ile Bluetooth Kablosuz Haberleşme

App Inventor ile hazırlayacağımız uygulamanın bluetooth haberleşmeyi kullanması için iki farklı nesne bulunmaktadır. Bu nesneler, "Designer" sekmesinde, "Palette" bölümünün altında yer alan "Connectivity" modülünde bulunmaktadır. "BluetoothClient" nesnesi uygulamanın yüklü olduğu cihazın bluetooth istemci olarak davranmasını, "BluetoothServer" nesnesi ise bluetooth sunucu olarak davranmasını sağlar.

"Designer" sekmesinden uygulamamıza eklediğimiz bluetooth nesnesi görünmeyen nesne olarak uygulamamıza eklenir. Bluetooth nesnesinin temel özelliklerini düzenlemek için "Components" bölümünden ilgili nesneyi seçtikten sonra "Properties" bölümünü kullanabiliriz.

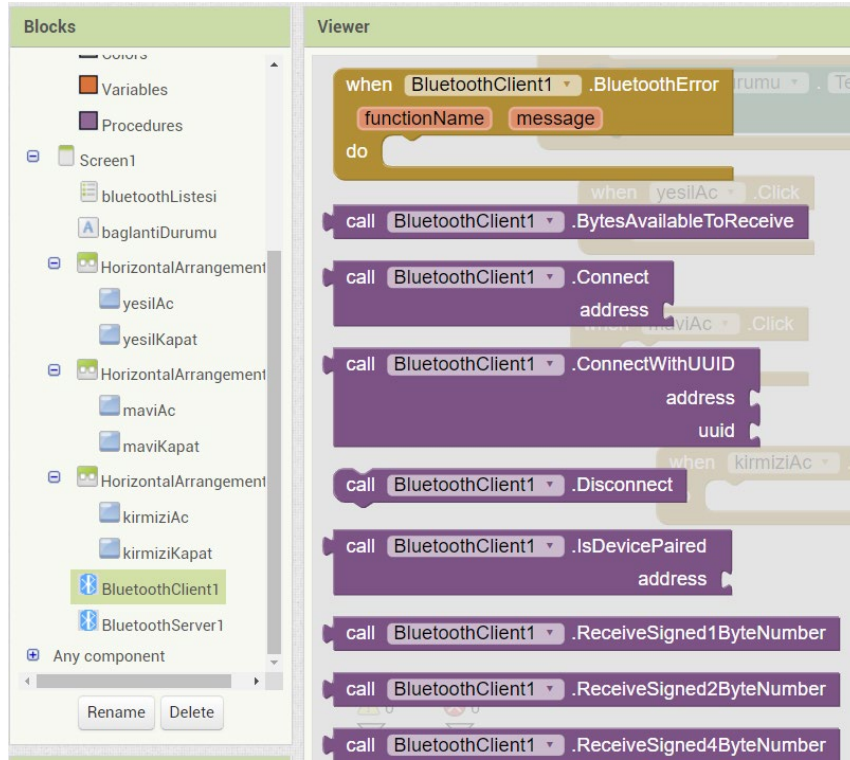


Ekran Görüntüsü 45. Bluetooth İstemci Özellikler Bölümü

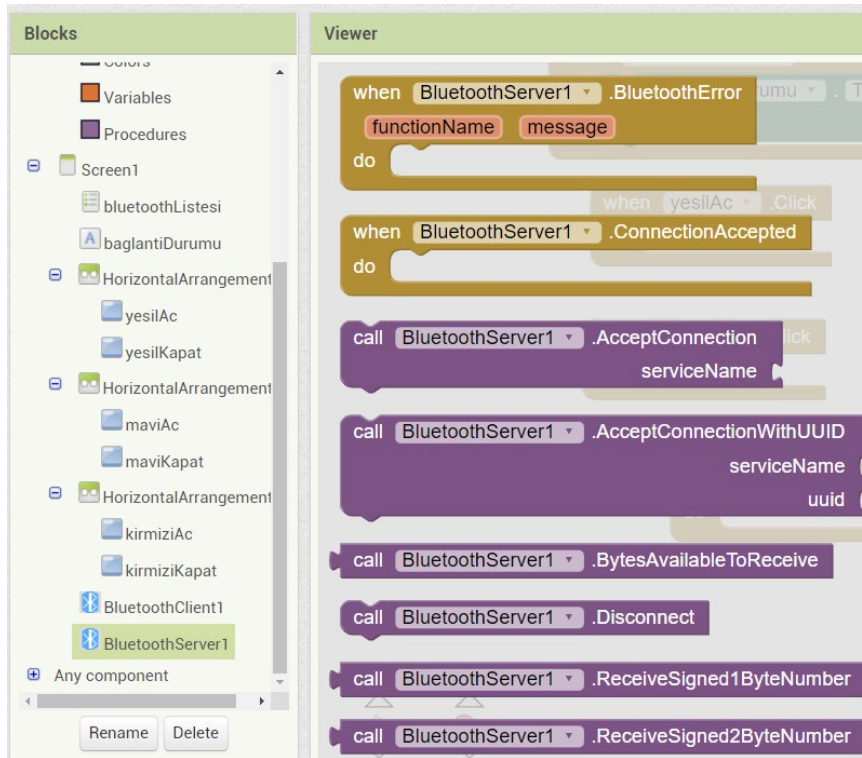


Ekran Görüntüsü 46. Bluetooth Sunucu Özellikler Bölümü

Uygulamaya eklediğimiz bluetooth nesnesi ile ilgili yapılandırma ayarlarını ve sonrasında gönderilecek veri veya alınacak verilerle ilgili işlemleri “Blocks” sekmesinden yapabiliriz.



Ekran Görüntüsü 47. Bluetooth İstemci Bloklar Ekranı

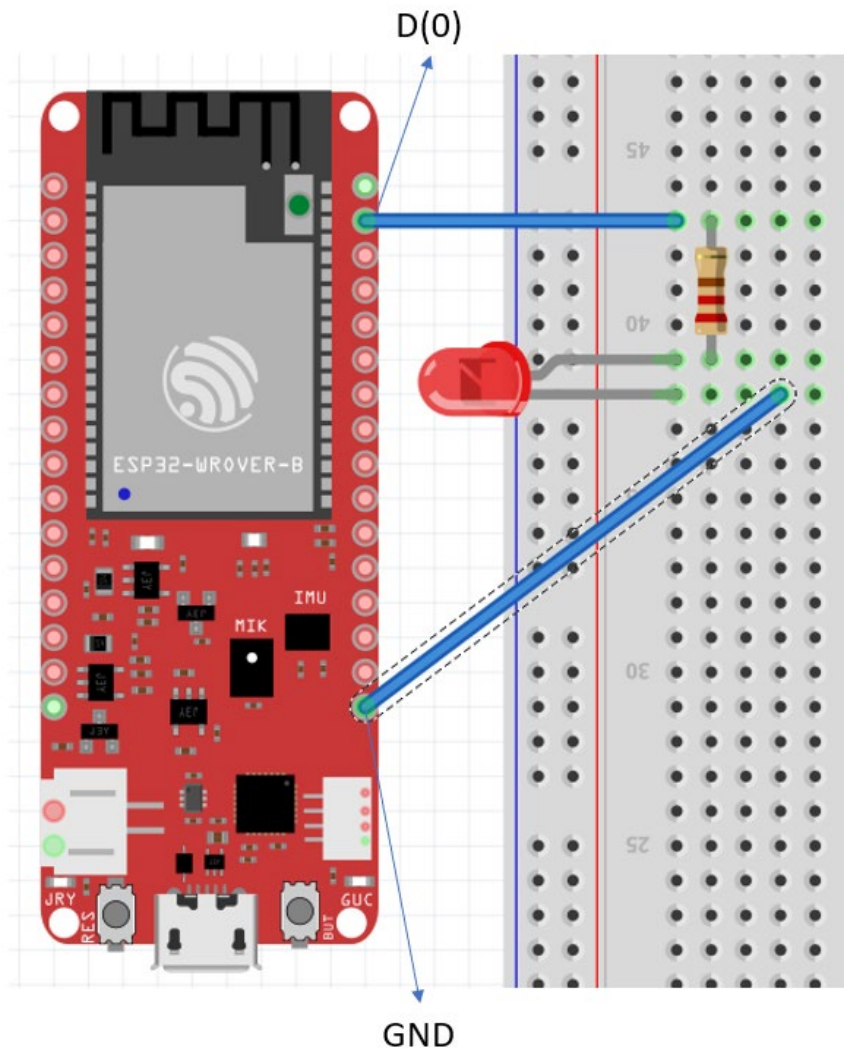


Ekran Görüntüsü 48. Bluetooth Sunucu Bloklar Ekranı

1.2.5. App Inventor ile Bluetooth Haberleşme Uygulaması (Deneyap Kart ve Arduino Elektronik Kart)

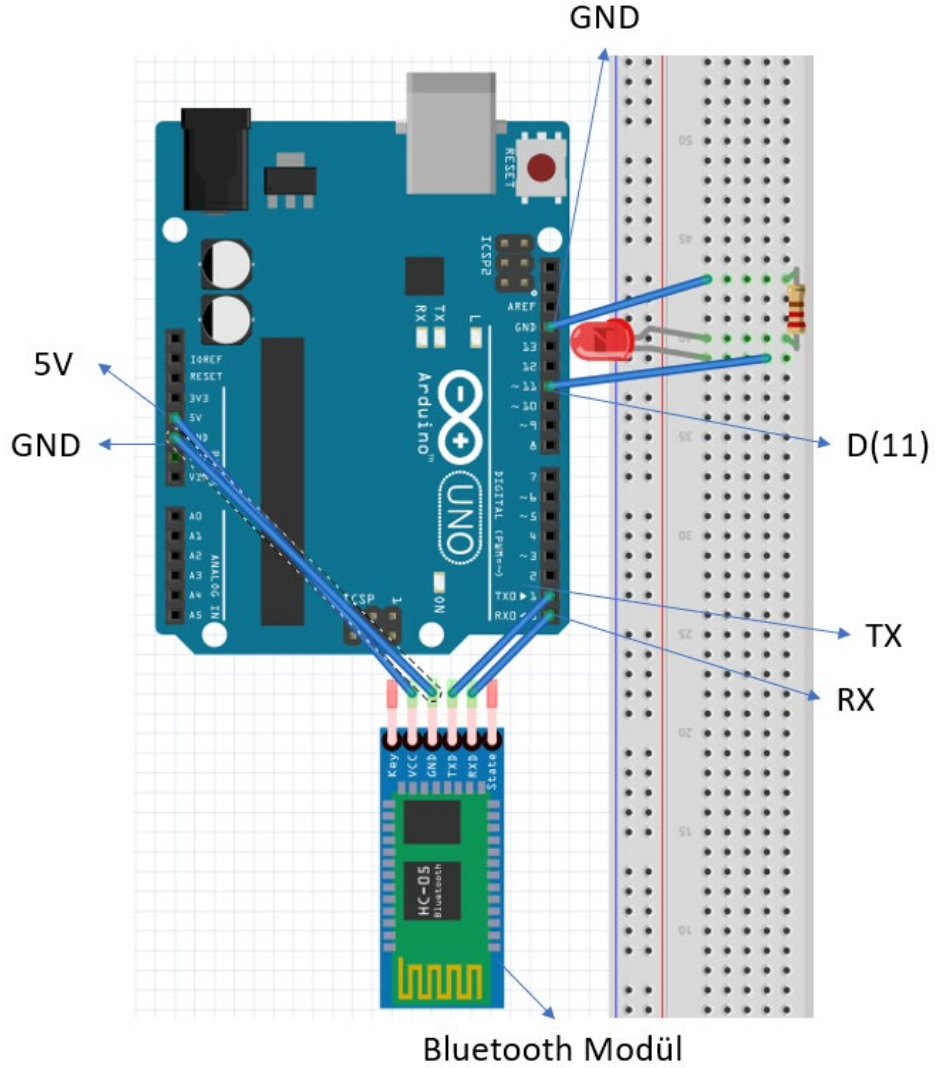
Mobil cihazımızın bir bluetooth istemci olarak davranacağı ve Deneyap veya Arduino elektronik devre kartına bağlanıp kart üzerinde ilgili soketlere bağlı yeşil, mavi ve kırmızı ledlerin yakıp söndürülebileceği bir uygulamayı birlikte hazırlayalım.

Öncelikle Deneyap veya Arduino elektronik devre kartının doğru şekilde hazırlanması gerekmektedir. Deneyap elektronik devre kartında, kart üzerinde D(0) ve GND pinlerini kullanarak Ekran görüntüsü 9'daki düzenek hazırlanır. Deneyap kartın kendi üzerinde Bluetooth modülü bulunması sebebiyle bununla ilgili bir bağlantı yapmamız gerekmemektedir.



Ekran Görüntüsü 49. Deneyap Elektronik Kart Devre Yapısı

Arduino elektronik devre kartında, yanmasını istediğimiz ledi kart üzerindeki D(11) ve GND soketlerine bağlamamız gerekmektedir. Bluetooth haberleşmenin çalışabilmesi için ise devre kartımıza bluetooth sensör modülünün doğru şekilde bağlanmış olması gerekmektedir. Bluetooth sensör modülü Arduino kartında, veri iletişimini sağlamak için TX-RX slotlarına, enerji alabilmesi için de sensör modülünün üstünde yazan voltaj değerine göre ilgili slota ve topraklama için de GND slotlarına bağlı olması gerekmektedir. Ekran görüntüsü 10'daki düzenek hazırlanır.



Ekran Görüntüsü 50. Arduino Elektronik Kart Devre Yapısı

Devre elemanları elektronik devre kartına doğru şekilde bağlandıktan sonra aşağıdaki örnek kodlar devre kartına yüklenir ve devre kartımız bağlantı için hazır hale getirilir. (Bu kodlar örnektir, farklı kod yapıları ile de aynı işlem yapılabilir.)

```
#include "deneyap.h"
#include "BluetoothSerial.h"

BluetoothSerial SerialBT;

char deger;

void setup() {
    pinMode(D0, OUTPUT);
}

void loop() {
    if(SerialBT.available()){
        deger=SerialBT.read();
        if( deger == '1'){
            digitalWrite(D0,HIGH);
        }
        if( deger == '2'){
            digitalWrite(D0,LOW);
        }
        delay(500);
    }
}
```

Ekran Görüntüsü 51. Deneyap Elektronik Devre Kartı için Kodlar

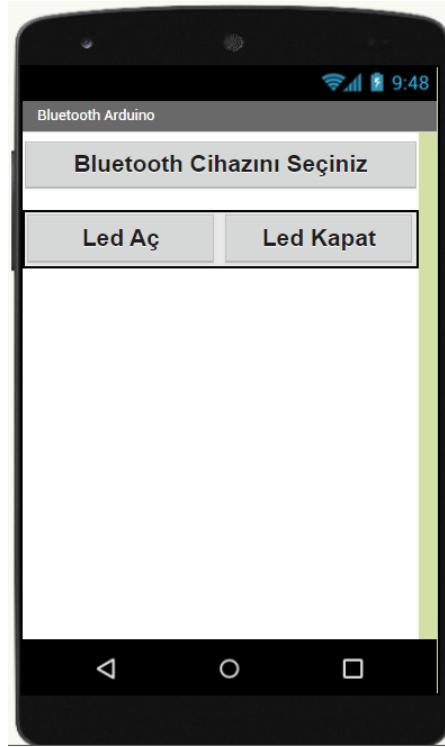
```
byte deger;

void setup() {
    pinMode(11, OUTPUT);
}

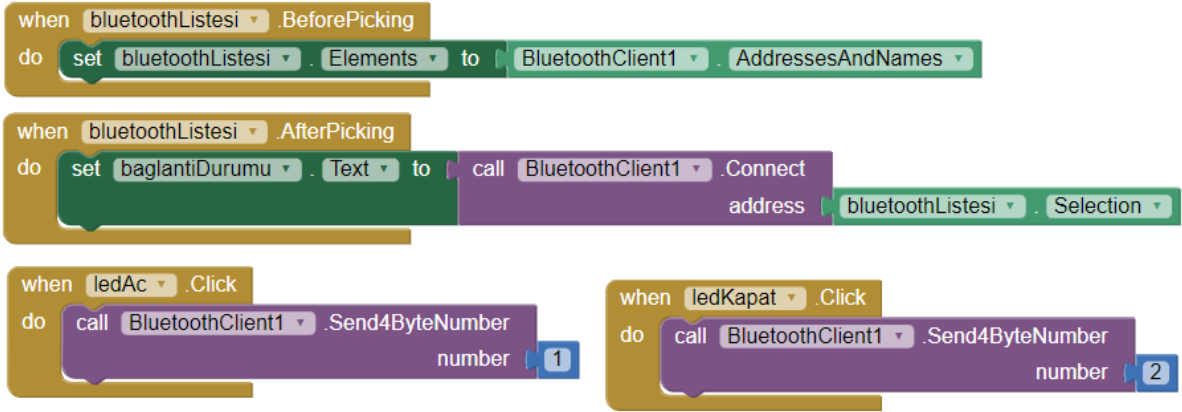
void loop() {
    deger=Serial.read();
    if( deger == '1'){
        digitalWrite(11,HIGH);
    }
    if( deger == '2'){
        digitalWrite(11,LOW);
    }
    delay(500);
}
```

Ekran Görüntüsü 52. Arduino Elektronik Devre Kartı Kod Ekranı

Elektronik devre kartı hazırlandıktan sonra uygulamamızı da aşağıdaki ekran görüntüsü örnek oluşturacak şekilde hazırlayabiliriz. Uygulamanın tamamlanmış proje dosyası ekler kısmında bulunabilir.



Ekran Görüntüsü 53. Viewer (Görüntüleyici) Görüntüsü



Ekran Görüntüsü 54. Blocks Görüntüsü

2. Tasarla

Öğrencilerin bu hafta gösterilen sensörler ve haberleşme yöntemlerinden istediklerini kullanarak bir uygulamanın tasarımı hazırlaması istenir. Bu tasarlama sürecinde öğrencilerin, uygulamanın hangi amaca hizmet edeceği ve sayfa tasarımlarını belirlemeleri beklenir.

3. Üret

Tasarla bölümünde tasarımını belirledikleri uygulamanın bu bölümde hem görsel tasarımını hem de kod yapılarını hazırlamaları ve çalışır bir uygulama hazırlayarak test etmeleri beklenir.

4. Değerlendir

Üret aşamasında hazırlanan uygulama Emulator ya da AI Companion aracılığı ile çalıştırılacak ve test edilecektir. Öğrenciler kendi hazırladıkları uygulamaları diğer öğrencilerin de test edebilmelerini sağlayacaktır. Öğrencinin öz değerlendirme yapabilmesi için Öz Değerlendirme Formu – 1 (<https://forms.gle/jiofyNvaDj48fJ9P8>) öğrenciye uygulanır.

İkinci hafta başlanan proje geliştirme süreci ile ilgili Öz Değerlendirme Formu – 2'yi (<https://forms.gle/gq4b5NCLFCAYp2c78>) doldurması istenecektir. Bu rapor eğitmen tarafından da incelenip destek verilmesi gereken öğrenciler belirlenecektir.

5. Hafta: Projenin Derlenmesi ve Google Play'de Yayınlama

Ön Bilgi:

- Temel bilgisayar bilgisi
- Temel yazılım bilgisi

Haftanın Kazanımları:

- Tamamlanan uygulamayı test eder.
- Derleme terimini tanımlar.
- APK(Android Package Kit) dosyalarını açıklar.
- AAB(Android App Bundle) dosyalarını açıklar.
- Google Play uygulama mağazasını tanır.
- Google Play Console hakkında bilgi sahibi olur.

Haftanın Amacı:

App Inventor ile hazırlanan bir uygulamanın tamamlanması sonrasında son testlerinin yapılmasının öneminin kavranması ve bu testlerin yapılması amaçlanmaktadır. Testler tamamlandıktan sonra uygulamanın derlenmesi hedeflenmektedir. Derleme sonrasında uygulamanın kullanıcılar tarafından ulaşılabilir olması için Google Play uygulama mağazasında yayınlama konusunda bilgi sahibi olması beklenmektedir.

Kullanılacak Malzemeler:

Bilgisayar

Haftanın İşlenişi:

Gözele: Proje test etme, derleme (build), APK, AAB kavramları hakkında bilgi sahibi olur, Google Play uygulama mağazası hakkında var olan bilgilerini ortaya koyar ve burada nasıl uygulama yayınlanabileceğini kavrar

Uygula: Daha önceki haftalarda tamamlanan bir uygulamanın AI Companion ile test edilmesini sağlar, uygulamanın derleme işlemini tamamlar ve Google Play uygulama mağazasına yüklemek için gerekli hazırlıkları yapar

Tasarla: Önceki haftalarda hazırlanan projelerden birini Google Play uygulama mağazasına yüklenmek üzere seçer ve derleme öncesi yapılacak testleri hazırlar

Üret: Google Play uygulama mağazasına yüklemek için bir uygulamanın hem APK'sını hem de AAB'sini oluşturur

Değerlendir: Üret bölümünde hazırlanan uygulama çıktılarının değerlendirilmesi ve proje geliştirme süreci ile ilgili ara değerlendirme yapılması

1. Gözle ve Uygula

Bir önceki haftalarda kullanmaya başladığımız App Inventor ile uygulama geliştirme sürecinde öğrencilere aşağıdaki sorular yöneltilir.

- Tamamlanan bir uygulamanın test edilmesi gerekli midir?
- Uygulama test edilirken nelere dikkat edilmelidir?
- Tamamlanan bir uygulama mobil cihazlara nasıl kurulur?
- Android işletim sistemi çalışan mobil cihazlarda uygulama yüklemek için neler yapılabilir?
- APK ve AAB dosyalarını hiç duydunuz mu?
- Google Play uygulama mağazasına uygulamalar nasıl eklenir?

Sorular hakkında öğrencilerin tartışmaları istenir ve tartışmanın ilerlemediği durumlarda öğretmen konuyu yönlendirir. Öğitmen, konuyu yönlendirmede aşağıdaki bilgilerden yararlanabilir.

1.1. Test Etme ve Derleme

Uygulamanızı test etmek ve derlemek, oluşturduğunuz uygulamanın doğru çalıştığından ve hedef kitleniz için uygun olduğundan emin olmak için önemlidir. App Inventor, uygulamanızı test etmek ve derlemek için kullanıcı dostu araçlar sunar.

1.1.1. Test Etme

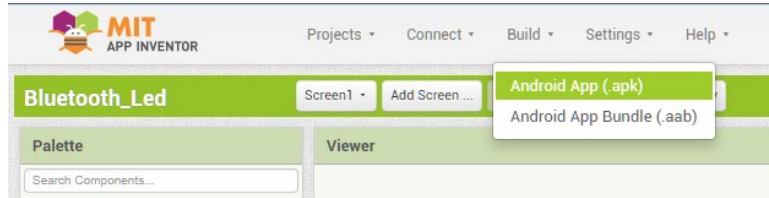
Uygulamanızı test etmek için gereken araçlarla ilgili bilgi, birinci bölümde ve ilgili videolarda (https://youtu.be/trJ6lx6jS_I - <https://youtu.be/rJRmopb9sQg>) verilmişti. Bu araçlar ile uygulamamızın son hali mobil cihazda çalıştırılarak öncelikle açılması ile ilgili herhangi bir sorun olup olmadığı gözlemlenir.

Uygulama mobil cihazımızda sorunsuz açıldıktan sonra projemizde uygulamamızın hangi amaçla yapıldığı göz önünde bulundurularak yapmasını beklediğimiz her işlemi yapıp yapmadığı, hatalı bir işlem yürütüp yürütmediği gibi teknik süreçler test edilir.

Uygulamanın test edilmesi gereken bir diğer boyutu ise tasarımsal boyuttur. Tasarım öğeleri açısından projemiz istediğimiz gibi çalışıp çalışmadığını test etmeliyiz. Ayrıca mümkün olduğunca farklı özellikteki mobil cihazlarda (çözünürlük, işlemci gücü, Android sürümü, v.b) uygulamamızı test etmemiz yararlı olacaktır.

1.1.2. Derleme (Build)

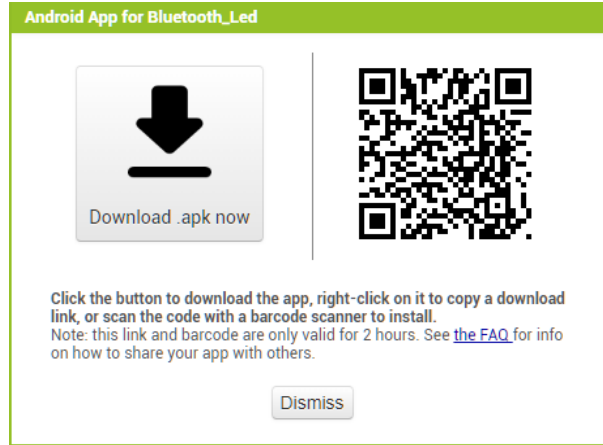
Uygulamanızı test ettikten sonra, "Build" menüsünden "Android App (.apk)" ya da "Android App Bundle (.aab)" seçeneklerinden birini tıklayarak uygulamanızı derleyebilirsiniz. Derleme işlemi, uygulamanın kaynak kodlarını, grafiklerini, ses dosyalarını ve diğer bileşenlerini tek bir APK ya da AAB dosyasına dönüştürür.



Ekran Görüntüsü 55. Build (Derleme)

1.1.2.1. Android App (.apk)

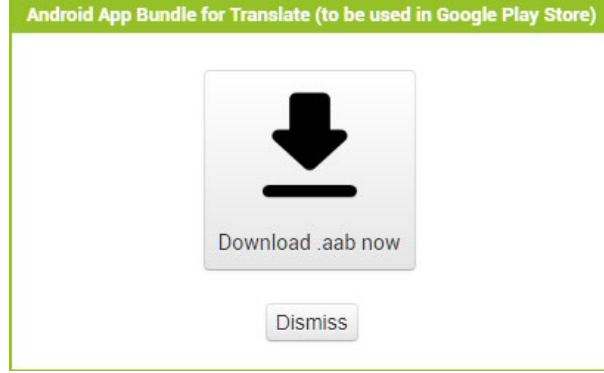
Uygulamanızın derlenmesi sonucunda Android cihazlarda kurulabilir şekilde bir dosya çıktısı istiyorsanız bu seçeneği seçmeniz gerekmektedir. Bu derleme sonucunda MIT App Inventor size uzantısı ".apk" olan bir dosya indirme bağlantısı ve bir QR kod verecektir. Bu QR kodu okutarak derleme işleminden sonraki 2 saat boyunca derlenmiş dosyanızı indirebileceğiniz bir bağlantıya ulaşabilirsiniz. Bu bağlantıdan dosyanızı direk olarak bir mobil cihaz aracılığı ile indirerek kurulumunu gerçekleştirebilirsiniz.



Ekran Görüntüsü 56. Android App (.apk)

1.1.2.2. Android App Bundle (.aab)

Uygulamanızın derlenmesi sonucunda Google Play uygulama mağazasına yüklemek için bir dosya çıktısı istiyorsanız bu seçeneği seçmeniz gerekmektedir. Bu derleme sonucunda MIT App Inventor size uzantısı “.aab” olan bir dosya indirme bağlantısı verecektir.



Ekran Görüntüsü 57. Android App Bundle (.aab)

Android App Bundle, Google tarafından sunulan bir format olarak, Android uygulamalarının daha verimli ve optimize edilmiş şekilde dağıtılmasını sağlamak için kullanılan bir paketleme yöntemidir. Geleneksel APK dosyalarının yerini alabilen Android App Bundle, uygulamanın sadece kullanıcının cihazına ihtiyacı olan parçalarını içeren özelleştirilmiş bir paket oluşturarak uygulama boyutunu azaltır.

Android App Bundle, uygulamaların boyutunu azaltırken aynı zamanda uygulama performansını da artırır. Bu sayede uygulama yükleme süresi hızlanır ve kullanıcı deneyimi iyileştirilir. Ayrıca, Android App Bundle kullanılarak oluşturulan uygulamalar, Google Play uygulama mağazasında dinamik olarak optimize edilebilir. Bu, kullanıcının cihazındaki donanıma, ekran boyutuna, API sürümüne ve diğer faktörlere göre en uygun sürümün indirilmesini sağlar.

Android App Bundle'in bir diğer önemli özelliği, uygulamaların modüler hale getirilmesini sağlamasıdır. Bu sayede uygulama, farklı özellikleri farklı modüllere ayırarak uygulamanın boyutunu daha da azaltabilir. Bu özellik, uygulamanın sadece belirli bir özelliğine ihtiyacı olan kullanıcıların yalnızca bu özelliği indirmesine olanak tanır.

Sonuç olarak, Android App Bundle, uygulama boyutunu azaltmak, uygulama performansını artırmak ve uygulama dağıtımını optimize etmek için kullanışlı bir yöntemdir. Bu özellik, geliştiricilere, uygulamalarını daha iyi bir şekilde optimize etme ve kullanıcıların deneyimini iyileştirme fırsatı verir.

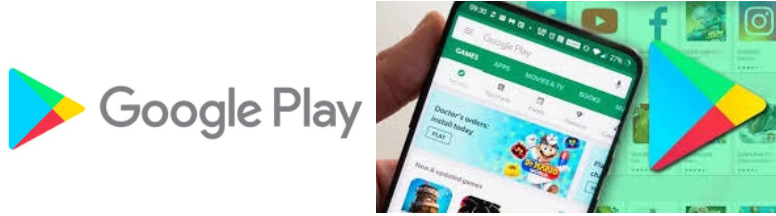
1.2. Google Play Uygulama Mağazası

Google Play, dünya çapında milyarlarca kullanıcının Android işletim sistemli cihazlarından uygulama, oyun, müzik, kitap, film ve TV şovu indirmesine olanak tanıyan bir uygulama mağazasıdır. Google Play uygulama mağazası, Google tarafından yönetilmektedir ve 2008 yılında Android Market olarak piyasaya sürülmüştür.

Google Play uygulama mağazası, geliştiricilerin uygulamalarını ve oyunlarını Android kullanıcılarına sunmalarını sağlamak için bir platform olarak hizmet verir. Geliştiriciler, Google Play Console adı verilen bir web tabanlı arayüz aracılığıyla uygulamalarını yönetebilir, inceleyebilir ve yayınlatabilirler. Google Play Console aynı zamanda geliştiricilerin uygulamalarının performansını takip etmeleri, satışlarını izlemeleri ve gelirlerini yönetmeleri için de kullanılır.

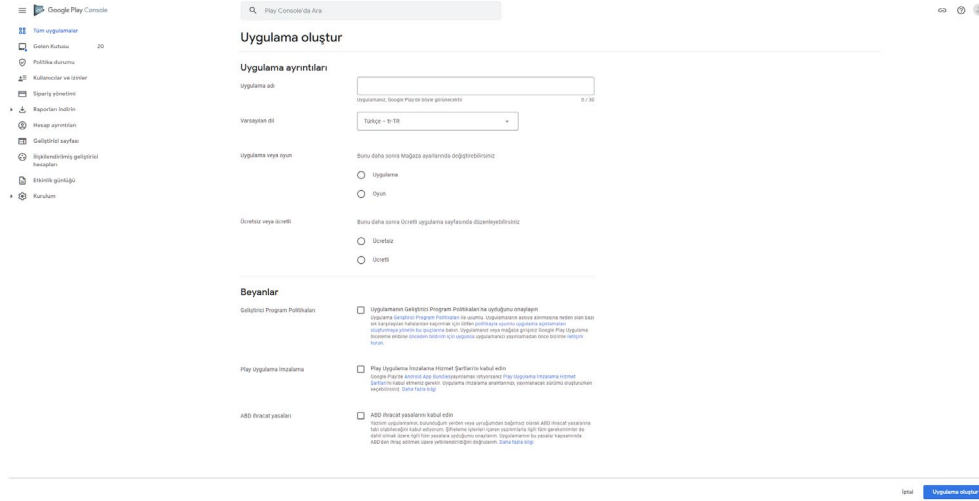
Google Play uygulama mağazasında yayınlanan uygulama ve oyunlar, kullanıcılara ücretsiz veya ücretli olarak sunulabilir. Kullanıcılar, Google Play uygulama mağazasında istedikleri uygulama veya oyunu arayabilir, inceleyebilir, derecelendirebilir ve yorum yapabilirler. Ayrıca Google Play uygulama mağazası, özellikle uygulama ve oyunlar için çeşitli ödeme yöntemleri sunar ve geliştiricilerin uygulamalarını farklı ülkelerde farklı fiyatlarda satmalarına olanak tanır.

Özet olarak; Google Play uygulama mağazası, Android kullanıcılarına dünya genelinde en büyük uygulama mağazası seçeneklerinden birini sunmaktadır. Geliştiricilerin uygulamalarını ve oyunlarını yayınlamaları ve gelir elde etmeleri için önemli bir platformdur. Kullanıcılar için de aradıkları her türlü uygulama ve oyunu kolayca bulmalarını ve indirmelerini sağlamak için kullanıcı dostu bir arayüz sunmaktadır.



Ekran Görüntüsü 58. Google Play Uygulama Mağazası

Uygulamanızı Google Play Store'da yayınlamak istiyorsanız, önce bir Google Play geliştirici hesabı oluşturmanız ve uygulamanızı Google Play Console'a yüklemeniz gerekmektedir. Uygulamanızı Google Play Store'a yüklemek için gerekli adımlar, Google Play Console'da ayrıntılı bir şekilde açıklanmaktadır.



Ekran Görüntüsü 59. Google Play Console

2. Tasarla

Öğrencilerden ders kapsamında hazırladıkları uygulamalardan birini seçmeleri istenir. Seçilen uygulama ile ilgili derleme işlemi yapmadan önce testler yapması beklenir. Testleri yapmaya başlamadan önce uygulamanın gerçekleştirmesini bekledikleri bütün durumları kapsayan senaryolar yazması istenir. Bu senaryolar, uygulamanın proje aşamasında açıklanan şekliyle çalışıp çalışmadığını, teknik sorunlar içerip içermediğini, tasarım olarak beklenen istekleri karşılayıp karşılamadığı gibi durumları denetleyebilecek nitelikte olmalıdır.

3. Üret

Tasarla sekmesinde belirlenen ve testleri tamamlanan uygulamanın bölüm içerisinde anlatıldığı şekliyle APK ve AAB olarak derleme işlemleri yapılır ve çıktıları alınmaları beklenir.

4. Değerlendir

Üret aşamasında alınan çıktı dosyaları öğretmenlerle ve diğer öğrencilerle paylaşılır. Böylelikle öğretmen ve diğer öğrenciler de bu uygulamaları Android mobil cihazlarına kurup, deneyebilecektir. Öğrencinin özdeğerlendirme yapabilmesi için Öz Değerlendirme Formu – 1 öğrenciye uygulanır.

İkinci hafta başlanan proje geliştirme süreci ile ilgili Öz Değerlendirme Formu – 2'yi doldurması istenecektir. Bu rapor öğretmen tarafından da incelenip destek verilmesi gereken öğrenciler belirlenecektir.

Ayrıca bir sonraki hafta yapılacak dersin sonunda projelerini sunacakları hakkında öğrencilere bilgi verilir. Proje sunumu sırasından kullanmaları için proje fikirlerini açıkladıkları, tasarım süreçlerinden bahsettikleri, uygulamayı tamamlama süreçlerinden ve kullandıkları kodlardan örnekler verdikleri bir sunum hazırlamaları gerektiği anlatılır.

6. Hafta: Kod Bloklarını Tekrar Kullanma ve Yayınlama - Proje Sunumları

Ön Bilgi:

- Temel bilgisayar bilgisi
- Temel yazılım bilgisi

Haftanın Kazanımları:

- Hazırlanan kod bloklarını kaydeder.
- Kaydedilen kod bloklarını tekrar kullanır.
- Hazırladığı projeyi App Inventor proje galerisine kaydeder.
- App Inventor proje galerisini kullanabilir.
- Proje sunumu gerçekleştirir.

Haftanın Amacı:

App Inventor ile hazırlanan projelerde kullanılan kod bloklarını kaydeder ve sonra farklı projelerde tekrar aynı kod bloklarını kullanır. App Inventor proje galerisini tanır, bu galeriye proje yüklemeyi öğrenir. App Inventor proje galerisini araştırmayı, bu projelerin tasarım ve kod blok ekranlarını incelemeyi keşfeder. Geliştirdiği özgün projenin sunumunu gerçekleştirir. Sunumu gerçekleştirilen projeleri inceler ve yeni bakış açıları kazanır.

Kullanılacak Malzemeler:

Bilgisayar, Mobil cihaz (cep telefonu ya da tablet), kalem, kâğıt

Haftanın İşlenişi:

Gözle: Proje geliştirme sürecine oluşturulan kod bloklarını kaydeder, kaydedilen kod bloklarını tekrar kullanır, kod bloklarını farklı projelere uyarlayabilir, geliştirilen projeyi farklı insanların da inceleyebilmesi için App Inventor proje galerisine kaydeder, proje galerisindeki farklı projeleri inceler ve yorum yapar

Uygula: Daha önce geliştirdiği projelerdeki kod bloklarını kaydederek yeni oluşturacağı bir projede bu kod bloklarını kullanabilir

Tasarla: App Inventor proje galerisinde yayınlamak için bir proje fikri oluşturur ve bu projeyi tasarlar

Üret: Tasarla bölümünde tasarımı hazırlanan uygulamanın üretilmesi ve App Inventor proje galerisine yüklenmesi

Değerlendir: Üret bölümünde hazırlanan uygulamaların değerlendirilmesi ve proje geliştirme sürecinin sonunda ortaya çıkan projenin sunumunun gerçekleştirilmesi

1. Gözle ve Uygula

Bir önceki haftalarda kullanmaya başladığımız App Inventor ile uygulama geliştirme sürecinde öğrencilere aşağıdaki sorular yöneltilir.

- Etrafınızda gördüğünüz nesneler sadece bir amaca mı hizmet eder, yoksa farklı amaçlar için de kullanılabilir mi? Örnek verebilir misiniz?
- Bir sorunun çözümü için izlenen yollar (algoritma) farklı sorunların çözümü için de kullanılabilir mi?
- Oluşturulan bir kod bloğu sadece belli bir işlemi yapmak için mi kullanılabilir?
- Proje geliştirirken karşılaştığınız bir sorunla daha önce karşılaştığınız ve çözüm ürettiğiniz durumlarla karşılaştığınız oldu mu?
- Geliştirdiğiniz projenin farklı kişiler tarafından da incelenebilir, kullanılabilir ve değiştirilebilir olması hakkında ne düşünüyorsunuz?
- Farklı kişilerin oluşturduğu projeleri incelemek, kullanmak ve değiştirmek ister misiniz? Bu size nasıl bir katkı sağlar?

Sorular hakkında öğrencilerin tartışmaları istenir ve tartışmanın ilerlemediği durumlarda öğretmen konuyu yönlendirir. Öğitmen, konuyu yönlendirmede aşağıdaki bilgilerden yararlanabilir.

1.1. Kod Bloklarını Tekrar Kullanmak

Bir uygulama geliştirme sürecinde yapılacak işlem adımları farklı bir uygulama geliştirme sürecinde de kullanılabilir. Uygulama geliştirme sürecinde bir sorunun çok fazla çözüm yöntemi bulunabileceği gibi bir çözüm yöntemi farklı sorunların çözümü için de kullanılabilir.

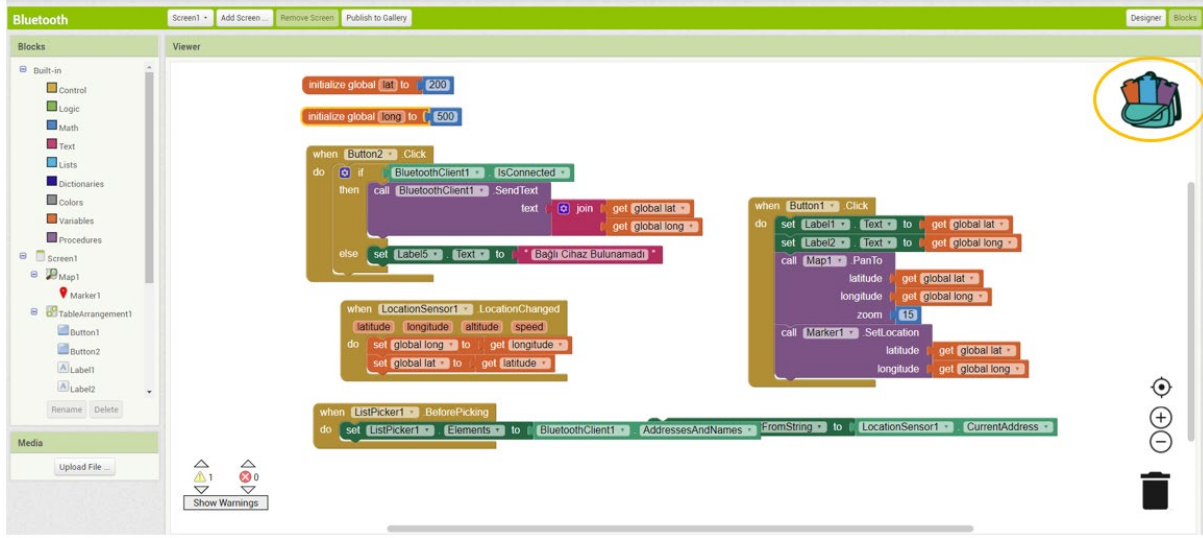
App Inventor ile uygulama geliştirirken oluşturulan kod blokları bazen bir iki satırdan oluşurken algoritmanın karmaşıklığına, yapılacak işlemin işlem adımlarının sayısına göre bazen çok fazla satırdan da oluşabilmektedir. Bu oluşturulan kod bloklarını kaydetmek ve oluşturduğumuz farklı projelerde de kullanmak mümkündür.

Kod bloklarını tekrar kullanabilmek bize hem daha hızlı proje geliştirme olanağı tanırken aynı zamanda da özellikle uzun kod blokları kullanılan durumlarda hata yapma ihtimalimizi de azaltır.

1.1.1. Kod Bloklarını Kaydetmek

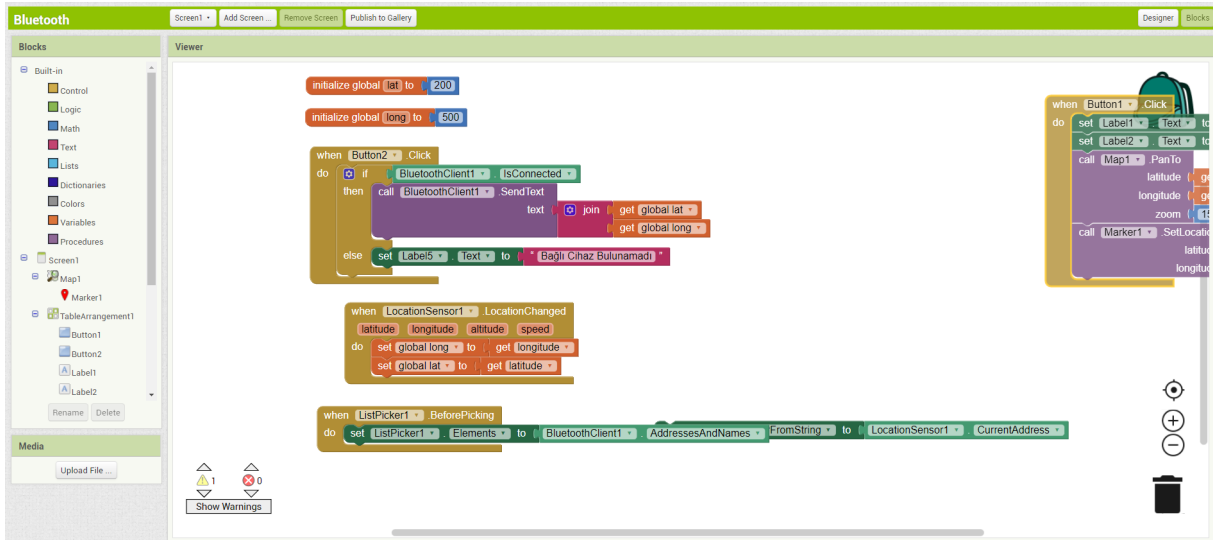
App Inventor'da uygulama geliştirirken yapılacak işlemlerin "Blocks" sekmesinde kod blokları ile yapıldığı önceki haftalarda öğrenmiştik.

"Blocks" sekmesinde ekranın sağ tarafında bulunan sayfa tasarımında oluşturduğumuz blokları sürükleyip bırakarak ekranının herhangi bir yerine yerleştirebiliriz. Sayfada sağ üst köşede bulunan sırt çantası ikonu bizim kod bloklarımızı kaydetmemize yardımcı olmaktadır.



Ekran Görüntüsü 60. Sırt Çantası İkonu

Kaydetmek istediğimiz kod bloğunu sürükleyip bırakarak Sırt Çantası İkonu'nun üstüne bırakırsak kod bloğumuzun çantamıza eklendiğini görebiliriz. Ve böylelikle farklı projelerimizde de bu kod bloğunu projemize ekleyip, gerekli düzenlemeleri yapabiliriz.

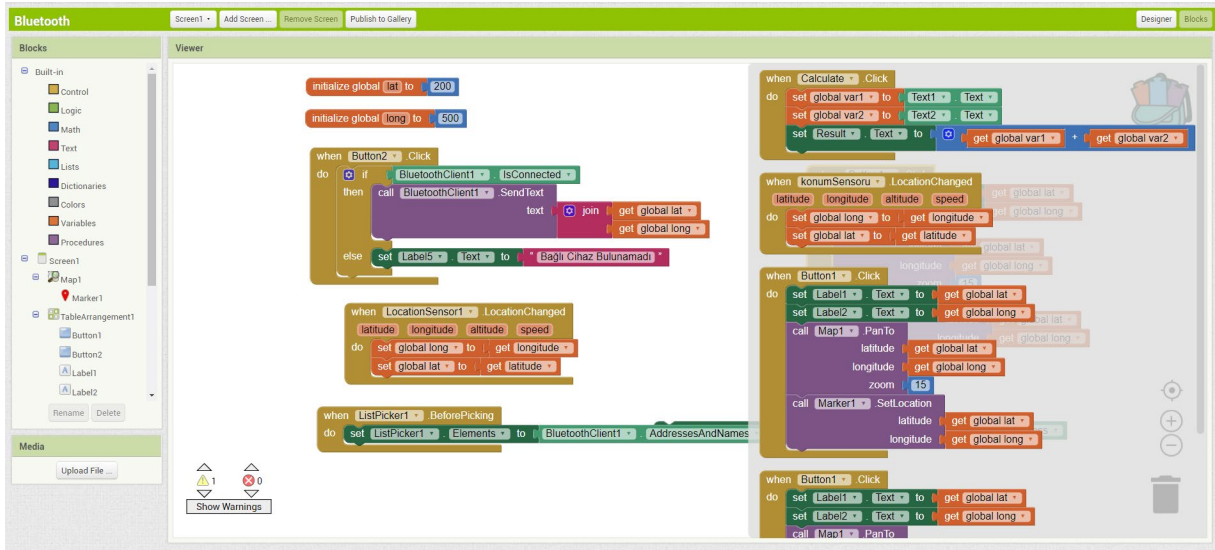


Ekran Görüntüsü 61. Kod Bloğunu Sırt Çantası Üzerine Götmek

1.1.2. Kayıtlı Kod Bloklarını Görüntülemek ve Kullanmak

Kod bloklarımızı kaydetmek için kullandığımız Sırt Çantası İkonu, bir nevi bizim kitaplarımızı taşıdığımız gibi kod bloklarımızı da taşımamıza yardımcı olan bir okul çantası gibi görev yapmaktadır.

Sırt Çantası İkonu'nu tıkladığımızda bize daha önce çantamıza eklediğimiz kod bloklarının bir listesini göstermektedir. Bu liste içerisinde kullanmak istediğimiz kod bloğunu sürükleyip bırakarak projemize ekleyebiliriz.



Ekran Görüntüsü 62. Kaydedilen Kod Bloklarını Görüntüleme

1.2. App Inventor Proje Galerisi

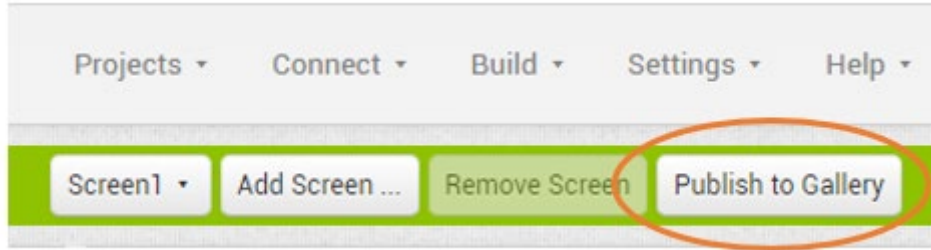
App Inventor Proje Galerisi, kullanıcıların geliştirdikleri projeleri diğer kullanıcılar ile paylaşabilecekleri bir ortamdır. Bu ortamda geliştirilen projeler ile ilgili detayları görebilir, ekran görüntülerini inceleyebilir, projeyi raporlayabilir, projeyi geliştiren kullanıcının farklı projelerini görüntüleyebilir ve en önemlisi projeyi kendi projeleriniz arasına kopyalayabilir ve böylelikle hem bütün kod bloklarını hem de tasarım öğelerini görüntüleyebilir hatta düzenleyebilirsiniz.

App Inventor Proje Galerisi, kullanıcıların paylaşımları ile ortaya çıkmış ve App Inventor kullanan herkese sınırsız kaynak sağlaması özelliğiyle çok iyi bir öğrenme kütüphanesi haline gelmiştir.

Kullanıcılar yükledikleri projelerin herkes tarafından görülmesini sağlayabilir ya da sadece arkadaşları ile paylaşmak üzere yalnızca bağlantıyı paylaştığı kişilerin ulaşabileceği şekilde projesini galeriye yükleyebilir.

1.2.1. App Inventor Proje Galerisine Proje Kaydetme

App Inventor'da proje geliştiren herkes çalışmalarını proje galerisine rahatlıkla ekleyebilmektedir. Projenizi geliştirdiğiniz ekranda sadece bir tuşa basarak (Publish To Gallery) projenizi galeriye eklemeniz için gerekli bilgileri dolduracağınız sayfaya yönlendirilirsiniz.



Ekran Görüntüsü 63. Publish To Gallery - Galeriye Ekle Butonu

Galeriye ekle butonuna tıkladıktan sonra yönlendireceğiniz sayfada projenizle ilgili açıklamalarınızı, projeyi geliştirenler ilgili bilgileri, detaylı açıklamalarınızı ve ekran görüntüsü ekleyebilirsiniz.

[Create Apps!](#)[About](#)[Educators](#)[News](#)[Resources](#)[Blogs](#)[Donate](#)

[Join](#)[MIT App Inventor](#)[New](#)[Gallery!](#)

Update Project

Name:

Description:

B I | | | | | | | ?

Credit:

B I | | | | | | | ?

More info:

B I | | | | | | | ?

Imagefile: Dosya seçilmedi

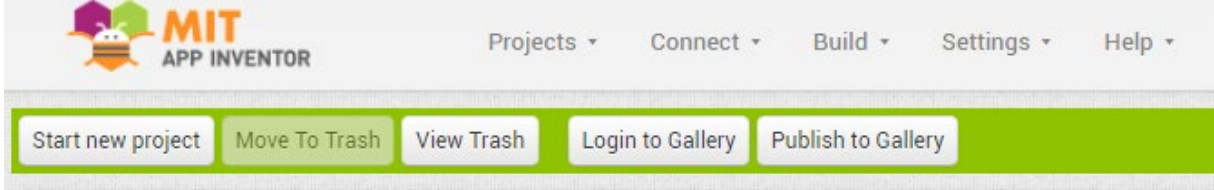
Check if you want your app to be findable with a search: ☐

Ekran Görüntüsü 64. Galeriyeye Proje Yükleme Ekranı

Proje yükleme ekranının en altında bulunan “Check if you want your app to be findable with a search” ifadesi, projenizin aramalarda listelenmesini isteyip istemediğinizi sormaktadır. Eğer bu kutucuğu tıklayarak onay vermezseniz yüklediğiniz uygulama herhangi bir moderasyona girmeden galeriye eklenecek fakat sadece sizin bağlantıyı paylaştığınız kişiler tarafından görüntülenebilecektir. Eğer kutucuğu tıklayarak onay verirsiniz, projeniz moderasyon incelemesi sonrasında herkesin görebileceği şekilde proje galerisine eklenecektir.

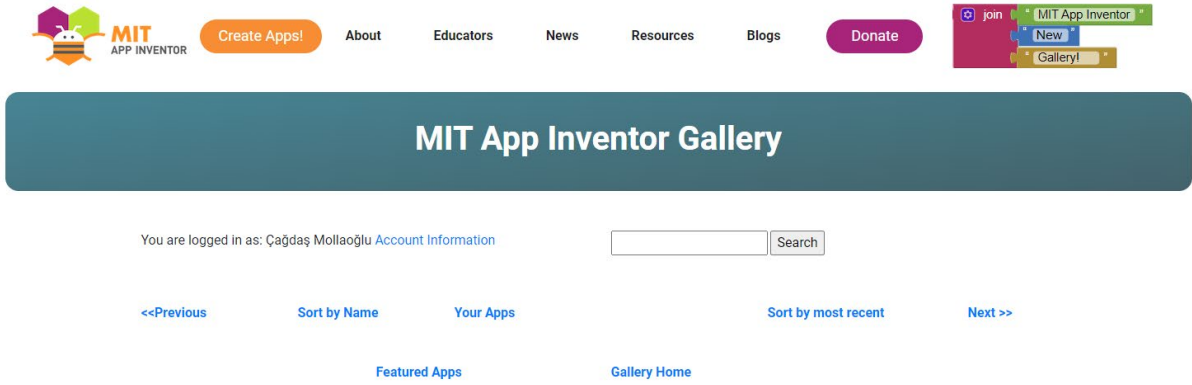
1.2.2. App Inventor Proje Galerisindeki Projeleri Görüntüleme

App Inventor proje galerisine yüklenmiş ve de yüklenirken aramalarda herkesin bulabilmesi için izin verilmiş bütün uygulamalar herkesin erişimine açık halde bulunmaktadır. App Inventor'a giriş yaptıktan sonra Benim Projelerim (My Project) sayfasındayken ekrandaki "Login to Gallery" (Galeriye giriş yap) butonuna tıklayarak proje galerisine giriş yapabiliriz.



Ekran Görüntüsü 65. Proje Galerisine Giriş Butonu

Proje galerisine giriş yaptığınızda projeleri daha rahat listeleyebilmeniz için bazı kategoriler oluşturulmuştur. Bunlar; Sort by Name (isme göre sırala), Your Apps (sizin uygulamalarınız), Sort by most recent (en son yüklenen uygulama en başta olacak şekilde sırala), Featured Apps (öne çıkan uygulamalar)'dır. Ayrıca uygulamalar arasında rahatça gezebilmeniz için de Next (sonraki sayfa), Previous (önceki sayfa) ve Gallery Home (galeri ana sayfası) gibi butonlar da ekranda bulunmaktadır.

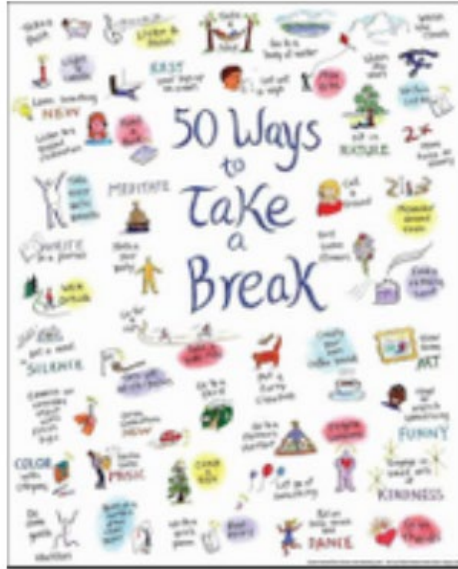


Ekran Görüntüsü 66. Proje Galerisi

Galerideki bir projenin yüklenme tarihini ve saatini görüntüleyebilir, kaç beğeni aldığına bakabilir ve projenin sahibi tarafından proje için yapılmış açıklamayı okuyabilirsiniz. Galeriye daha önce yüklenmiş bir projeyi kendi projelerinize ekleyebilir ve böylelikle hem kodlarını ve tasarımını görebilir hem de değiştirebilirsiniz. Bunun için "Load App into MIT App Inventor" (projeyi App Inventor'da yükle) bağlantısı tıklanabilir. Ayrıca bu yüklenen projenin yapımcısının farklı projelerini görmek için de "Other projects by same author" (Aynı üreticinin farklı projeleri) bağlantısı tıklanabilir. "Report Project" (projeyi raporla) bağlantısıyla da projeyi moderasyon birimine şikâyet edebilirsiniz. En altta bulunan "Permanent link:" (kalıcı bağlantı adresi) üzerinden ise projenin sayfasına direk olarak ulaşabileceğiniz bir bağlantıyı bulabilirsiniz.

CreativeProject

June 8, 2016, 3:45 p.m. Likes: 41 ♥



Our brains reach capacity at some point; if we were to engage in mindfulness, rather than allowing our minds to be too full, we may achieve the widely sought-after balance of work and play.

[Load App Into MIT App Inventor](#)

[Other projects by same author](#)

[Report Project](#)

Permanent link: <https://gallery.appinventor.mit.edu/?galleryid=5250904435720192>

Ekran Görüntüsü 67. Örnek Bir Proje Galerisi Uygulaması

2. Tasarla

Öğrencilerin daha önceki haftalarda ürettikleri bir projeyi ya da yeni bir fikirleri varsa bunu proje olarak oluşturarak galeriye yüklenecek şekilde ayrıntılarını, açıklamalarını yazmaları istenir. Projeyi galeriye eklemek için bir ekran görüntüsü de hazırlamaları beklenir.

3. Üret

App Inventor proje galerisine bir proje yükleyerek bu projenin kalıcı bağlantı adresini öğretmenler ile paylaşması beklenir. Böylelikle App Inventor'da geliştirdiği bir projeyi herhangi birisi ile App Inventor üzerinden nasıl paylaşacağını öğrenmiş olur.

4. Değerlendir

Dersin son haftası olarak bu haftanın sonunda ikinci hafta başlanan projelerin öğrenciler tarafından tamamlanmış olması beklenir. Bu projelerini bir sunum ile derste öğretmenlere ve diğer öğrencilere sunmaları istenir.

Öncelikle proje sunumunda proje fikirlerini açıklamaları, tasarım süreçlerinden bahsetmeleri, uygulamayı tamamlama süreçleri ve kullandıkları kodlardan örnekler vermeleri istenir. Sunum sonunda ise projelerinin çalışmasını emülatör üzerinden diğer öğrencilere gösterirler. Ders sonrasında ise öğrencilerin projelerini App Inventor proje galerisine yüklemesi ve kalıcı bağlantı adresini hem öğretmenle hem de diğer öğrencilerle paylaşması söylenir.

Dersin başında uygulanan Programlamaya Yönelik Tutum Ölçeği öğrencilere tekrar uygulanır. Ölçeğin uygulanması için internet bağlantısı (<https://forms.gle/s8zWSWTd8ssxQKvD9>) bütün öğrencilere gönderilir ve doldurmaları istenir. Bu form öğrencilerin e-posta bilgilerini toplamaktadır. Eğitim sonunda aynı form tekrar uygulanacaktır. Verilerin eşleştirilmesi amacıyla, öğrencilerinize her iki uygulamada da aynı e-posta adresini kullanmaları gerektiğini bildiriniz.

MOBİL UYGULAMA

ORTAOKUL

