

# App Inventor ve Alternatif Blok Tabanlı Mobil Uygulama Geliştirme Platformlarının Karşılaştırmalı İncelenmesi

## Comparative Review of App Inventor and Alternative Block Based Mobile Application Development Platforms

\*Özgür YILMAZ<sup>1</sup>

Ahmet Berk ÜSTÜN<sup>2</sup>

### Öz

Programlama öğretimi, her ne kadar öğrencilere analitik düşünme ve mantıksal çözüm yolları üretme becerisi kazandırmasına yardımcı olsa da zor ve sıkıcı bir süreç olarak görülmektedir. Öğrencilerin programlama eğitiminde yaşanan zorluklarla başa çıkabilmesi ve öğrenme sürecinin eğlenceli bir hale getirilmesi için blok tabanlı programlama platformları geliştirilmektedir. Programlama eğitiminde blok tabanlı mobil uygulama geliştirme platformlarının kullanılması ile öğrencilerin programlamaya olan ilgilerini arttıracakı söylenebilir. Bu çalışmada, blok tabanlı mobil uygulama geliştirme platformları olan MIT App Inventor, Thinkable ve Kodular karşılaştırmalı olarak iki aşamada incelenmiştir. İlk aşamada mobil uygulama geliştirme platformlarının genel özellikleri karşılaştırılmıştır. İkinci aşamada, aynı oyun uygulaması her bir mobil uygulama geliştirme platformu ile geliştirilmiş ve geliştirilen uygulama üzerinden platformlar karşılaştırılmıştır. Sonuç olarak, mobil uygulama geliştirme platformlarının çeşitli ölçütler açısından birbirlerine kıyasla sundukları avantajlar ve dezavantajlar sunulmuştur. Bu karşılaştırmalar sonucunda elde edilen bilgilerin eğitimcilere mobil programlama eğitimi için platform seçiminde yol göstereceği düşünülmektedir.

*Anahtar Kelimeler:* App Inventor, thinkable, kodular, blok tabanlı mobil uygulama geliştirme, kodlama.

### Abstract

Teaching programming has been seen as a difficult and boring process although it helps students improve their ability to think analytically and helps them produce logical solutions. Block-based programming platforms have been developed to cope with the difficulties students experience in programming education and to make the learning process entertaining. The use of block-based mobile application development platforms in programming education possibly increases students' interest in programming. In this study, MIT App Inventor, Thinkable and Kodular, which are block-based mobile application development platforms, were comparatively investigated in two stages. In the first phase, the general features of mobile application development platforms were compared. In the second phase, the same game application was developed with each mobile application development platform and the platforms were compared over the application. As a result, in terms of various criteria, the advantages and disadvantages offered by mobile application development platforms were compared and presented. It is believed that the findings obtained through these comparisons will guide educators in choosing a platform for teaching mobile programming.

*Keywords:* App Inventor, thinkable, kodular, block based mobile application development, coding

<sup>1</sup> Yüksek Lisans, Bartın Üniversitesi, Bartın-Türkiye, ozguryilmaz90@hotmail.com

<sup>2</sup> Dr. Öğr. Üyesi, Bartın Üniversitesi, Bartın-Türkiye, abustun@bartin.edu.tr

## GİRİŞ

Bilgi ve iletişim teknolojileri son yıllarda hızla gelişmektedir. Teknolojideki bu gelişmeler sonucunda bireylerin ihtiyaç duyacağı becerilerde değişmektedir. 21.yüzyıl bireyleri gibi adlandırmaların yapıldığı bu dönemde mantıksal çözüm yolları üretebilen, problem çözebilen ve yaratıcılık becerisi gelişmiş bireyler yetiştirilmeye çalışılmaktadır (Pinto & Escudeiro, 2014). Lai ve Yang (2011)'e göre öğretim süreçlerine programlama ve problem çözme stratejilerini entegre etmek öğrencilerin problem çözme becerilerini geliştirebilir. Programlama, herhangi bir problemin çözümü için gerekli komutlar dizisinin bilgisayarın anlayabileceği şekilde komutlara çevrilmesi, derlenmesi ve çalıştırılması sürecidir (Kesici & Kocabaş, 2001). Bu süreç, problemin çözümünde gerekli olan işlemleri, bilgisayara adım adım ve mantıklı bir sırada tanıtmayı gerektirir. Bu yönüyle programlama öğrenimi karmaşık bir süreçtir (Kert ve Uğraş, 2009; Helminen & Malmi, 2010). 21. yüzyıl becerilerini göz önünde bulundurarak son yıllarda programlama eğitime verilen önemin arttığı söylenebilir. Bunun nedeni olarak literatürde, bilgisayar programlama ve tasarım araçları öğretilirse, bireylerin problem çözme ve analitik düşünme becerileri, ürüne dönük büyük projeler yapma, küçük projelerin entegrasyonu ile karmaşık problemlere çözüm üretme alışkanlığı, yaparak öğrenme ve bilgisayara öğretmek öğrenme alışkanlıkları geliştirilebilir gibi birçok becerinin gelişmesine yardımcı olacağı düşünülmektedir (Akpınar & Altun, 2014).

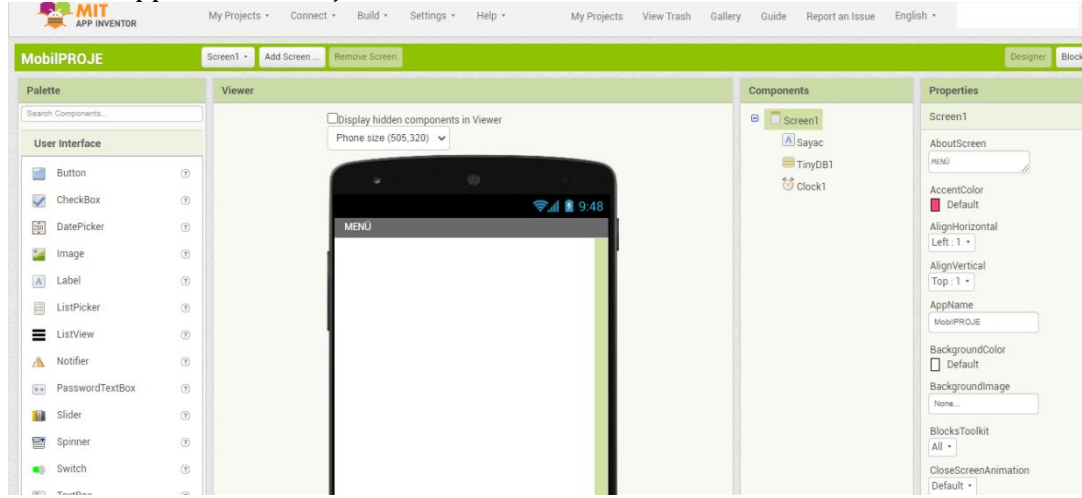
Birçok ülkede üniversite öğrencileri temel programlama kurslarını bitirdikten sonra kendilerinden beklenen bilgi ve becerilere sahip olamadıkları görülmektedir (Helminen & Malmi, 2010). Programlama eğitiminde öğrenme-öğretme yöntemleri, programlama dillerinin yapısı, kodların yabancı dilde olması, öğrencilerin programlamaya yönelik olumsuz tutum ve düşük motivasyonları gibi zorluklar yaşanmaktadır. Programlama öğretiminde yaşanan zorlukları aşmak için kullanılan ortamlardan birisi de kullanıcıların soyut programlama kodlarını somutlaştırarak öğrendiği ve anlık dönütler alabildiği blok tabanlı programlama ortamlarıdır (Saygıner & Tüzün, 2017).

Blok tabanlı görsel programlama ortamları öğrenenlerin kodu ezberlemek yerine hazır blokları sürükleyip bırak yöntemi ile program yazabilmelerini ve kodlama mantığının daha kolay bir şekilde anlaşılmasına olanak sağladığı söylenebilir. Blok tabanlı programlamada geleneksel sözdizimleri yerine görsel blokların kullanıldığı Scratch, App inventor ve Code.org ortamları ile ilgili alan yazında çok sayıda yayın bulunmaktadır (Baz, 2018). Baz (2018)'in çalışması göz önünde bulundurulduğunda Scratch, App inventor ve Code.org ortamlarının programlama eğitiminde yaygın şekilde kullanıldığı söylenebilir. Mobil cihazların yaygınlaşması ile programlama eğitiminde mobil programlama önem kazanmaktadır. Mobil uygulama geliştirmek için en çok kullanılan blok tabanlı programlama ortamının App inventor olduğu söylenebilir.

App inventor, Google tarafından geliştirilip MIT Mobil Öğrenme Merkezine devredilmiş olan programlama bilgisine ihtiyaç duymadan görsel bloklar ile mobil uygulama geliştirmek için kullanılan bir platformdur (Özdiñç, 2015). App inventor platformu Scratch ve Alice gibi öğrencilerin kod bloklarını sürükleyip birbirine bağlayarak uygulamalar oluşturabilecekleri görsel bir programlama ortamı sunar ve bu nedenle programlama eğitiminin zorluklarını azaltır (Roy, 2012). App inventor'un sezgisel programlama metaforu ve artımlı geliştirme yetenekleri, geliştiricinin herkes için dijital okuryazarlığı teşvik ederek kodlama dilinin sözdiziminden ziyade bir uygulamayı programlama mantığına odaklanmasını sağlar (Pokres & Veiga, 2013). MIT App Inventor (2020), herkesin özellikle de öğrencilerin teknoloji üretmeye teşvik etmeyi amaçlayan bir projedir. App Inventor'de uygulama geliştirirken kullanıcılar uygulama tasarımcısı, blok editörü, Android telefon ve Android emülatörü araçlarını kullanırlar (Smutný, 2011). App Inventor, özellikle öğrenci ve öğretmenlerin Android işletim sisteminde uygulama geliştirmeyi öğrenmede kolaylık sağlamayı amaçlamaktadır (Robianto, 2020). Mobil uygulama geliştirmeye

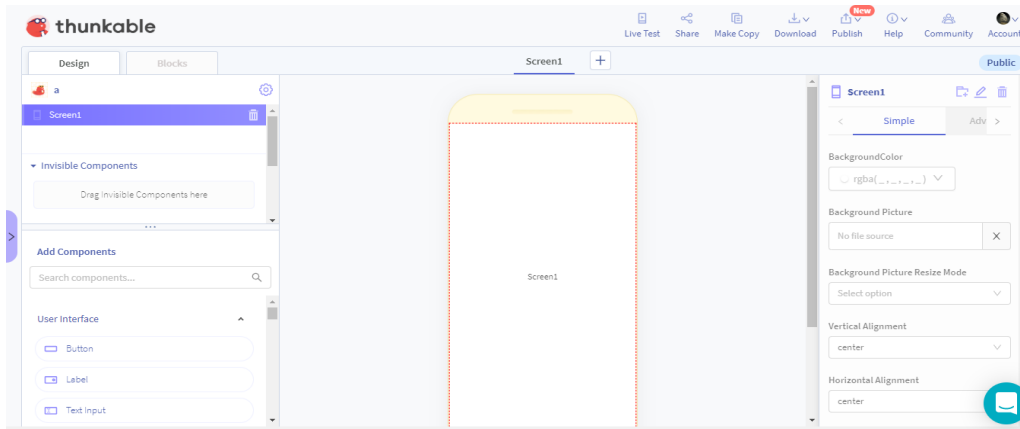
başlamak günümüz öğrencilerinin ilgisini çekeceğinden App Inventor programlama eğitimi için ideal bir araçtır (Roy, 2012).

Görsel 1. App Inventor Proje Ekranı



Thunkable, kod yazmadan mobil uygulama geliştirebileceğiniz App Inventor benzeri bir platformdur (Bales, 2018). Thunkable, (2020), platformlarını tek bir satır kod yazmadan IOS ve Android uygulamaları geliştirebileceğiniz görsel programlama dili olarak tanımlamışlardır. Thunkable'ın sağladığı hizmetler blok fonksiyonu, uygulamanın arayüzü, telefondaki sensörler, telefonun sosyal medyası, depolama hizmetleri, görselleştirme hizmetleri, bağlantı hizmetleri vb. gibi birkaç bölüme ayrılabilir (Fai & Audah, 2017). Thunkable ücretsiz sürümünde projeler diğer kullanıcılara ön izleme ve kaynak kodların kopyalanıp geliştirebilme için Thunkable genel galeride yayınlanır (Thunkable, 2020). Öğrenciler için başka kullanıcıların oluşturdukları projelerin kaynak kodlarının görüntülenebilmesi diğer kullanıcıların bileşenleri nasıl organize ettiğini analiz edebilmeleri ve başkalarının kodlarını kendi ihtiyaçlarına göre değiştirebilme avantajına sahiptir (Siegle, 2020). Ichsan, Sigit ve Miarsyah (2019) yaptıkları çalışmada Thunkable platformunu kullanarak doğa bilimleri konularında çevresel öğrenme için Android tabanlı bir yeşil tüketim kitapçığı geliştirmişlerdir. Bhavani ve Panjala (2017) çalışmalarında nesnelerin interneti kullanarak aile üyelerinin konumunu algılayacak bir yapı geliştirmişler ve Thunkable platformunda algılanan konumların görüntülenebileceği bir mobil uygulama geliştirmişlerdir.

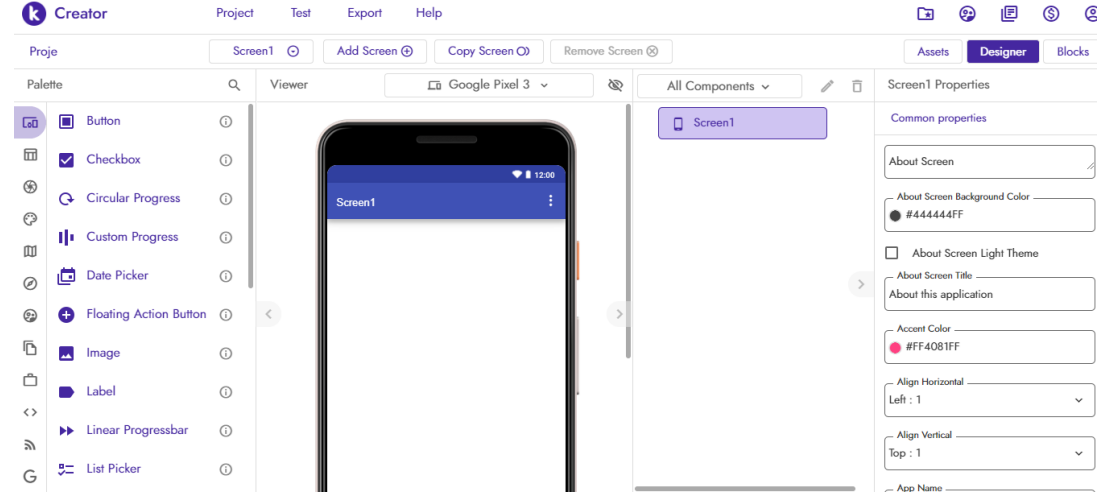
Görsel 2. Thunkable Proje Ekranı



Kodular, Mobil uygulamalar oluşturabileceğiniz App Inventor tabanlı bir programlama ortamıdır (João, Nuno, Fábio, & Ana, 2019). Kodular, (2020), Kodlama becerisine gerek duymadan blok

tabanlı bir düzenleyici ile kolayca android uygulamaları geliştirmeye olanak tanıdığını belirtmektedir. Kodular, bulut da dahil olmak üzere çeşitli veritabanlarıyla doğrudan entegrasyona sahiptir ve bağlantıyı ve bilgi alışverişini etkili bir şekilde kolaylaştırır (de Souza & da Silva, 2019). Zikri (2020) çalışmasında Android telefon ile uzaktan kontrol edilebilen otomatik bir bitki sulama sistemi geliştirmiş ve mobil uygulamanın geliştirilmesinde Kodular platformunu kullanmıştır.

Görsel 3. Kodular Proje Ekranı



Yapılan çalışmalar incelendiğinde blok tabanlı programlama ve metin tabanlı programlama karşılaştırmaları ile ilgili yapılan araştırmalar ve blok tabanlı programlamanın uygulanması üzerine araştırmalar yapıldığı görülmüştür. Mihçı (2014), yaptığı çalışmada programlamaya yeni başlayan öğrenci alt gruplarının blok tabanlı mobil uygulama geliştirme platformu kullananların, metin tabanlı programlama dili kullananlara göre akademik başarı açısından anlamlı ölçüde daha başarılı oldukları sonucuna ulaşmıştır. Bu makalede programlama eğitiminde blok tabanlı mobil uygulama geliştirme platformunu kullanmak isteyen eğitimciler için bulunan alternatifler arasından seçim yapmalarına yardımcı olmak adına App Inventor, Kodular.io ve Thunkable platformlarının geliştirilen uygulamanın performans değerleri üzerinden karşılaştırılması amaçlanmıştır. Uygun yazılımın seçilmesi verilecek olan eğitimin kalitesini arttırmada büyük önem taşır (Özmen & Varol). Fakat özellikle ulusal alanyazın incelendiğinde blok tabanlı mobil uygulama geliştirme ortamlarının karşılaştırıldığı bir çalışma ile karşılaşılmamıştır. Bu çalışmada blok tabanlı mobil uygulama geliştirme ortamlarından App Inventor, Kodular.io ve Thunkable üzerine karşılaştırmalı bir inceleme yapılacaktır. Bu amaç doğrultusunda aşağıdaki sorulara cevap aranacaktır.

Blok tabanlı mobil uygulama platformları genel özellikleri bakımından farklılaşmakta mıdır?

Blok tabanlı mobil uygulama platformları sundukları avantajlar yönünden farklılaşmakta mıdır?

Blok tabanlı mobil uygulama platformlarında kodlama geliştirme süreci farklılaşmakta mıdır?

## YÖNTEM

Bu çalışmada, örneklemin derinlemesine araştırabilmesine imkân sağlayan amaçlı örnekleme yolu izlenmiştir (Büyüköztürk, Akgün, Demirel, Karadeniz, & Çakmak, 2008). Amaçlı örneklemede, çalışma için belirlenen ölçütler seçilerek bu ölçütler doğrultusunda araştırma evrenini temsil edebilecek örneklem seçilir (Tavşancıl & Aslan, 2001). Bu doğrultuda, App Inventor ve alternatif blok tabanlı mobil uygulama ortamları arasında karşılaştırmalı bir analiz

yapılacağı için, App inventor'a alternatif niteliğinde olan blok tabanlı mobil uygulamalar seçilmiştir. Bu bağlamda App inventor, Thunkable ve Kodular.io çalışmanın örneğini oluşturmaktadır. İlk olarak incelenen platformların özellikleri karşılaştırılmıştır. Daha sonra üç ortamda aynı uygulama geliştirilmiş ve geliştirilen uygulamalar üzerinden platformlar değerlendirilmiştir.

### **Ölçütlerin Belirlenmesi**

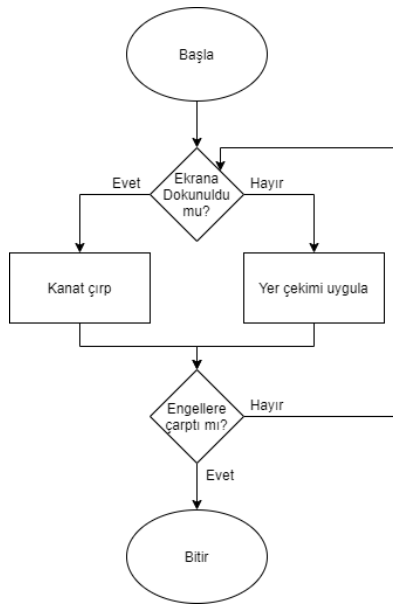
Belirlenen platformların genel özelliklerinin karşılaştırılması için daha önce blok tabanlı programlama ortamlarının karşılaştırma çalışmalarında belirlenmiş kriterler arasından (Baz, 2018; Papadakis, Kalojiannakis, Orfanakis & Zaranis, 2014) platform adı, erişim adresi, uygulama geliştirilen işletim sistemi, ücret durumu, Türkçe dil desteği, emülatör, giriş eğitimi ve topluluk desteği olmak üzere yedi başlık belirlenmiştir.

Performans değerlendirmesi için ise literatürdeki çalışmalar incelenmiş ve mobil cihazlar için önemli olduğu belirtilen (Spataru, 2010; Ryan & Rossi, 2005; Biørn-Hansen, Grønli, & Ghinea, 2019) fiziksel kaynaklardan ölçütler belirlenmiştir. Ryan ve Rossi (2005), çalışmalarında uygulamaların kaynak kullanımlarını da karşılaştırmış, deneysel çalışmalarında merkezi işlem birimi, bellek ve ağ kullanımını ölçüt olarak kullanmışlardır. Ayrıca, her gün binlerce akıllı telefon ve tabletin aktif hale getirildiği günümüzde oluşturulan uygulamalar güç için optimize edilmelidir (Dalmasso, Datta, Bonnet & Nikaen, 2013). Mobil cihazlar, masaüstü cihazlara göre depolama alanı, bellek kapasitesi ve işlem gücü gibi fiziksel kaynaklar açısından sınırlıdır ve bu kaynakların kullanımı dikkate alınmalıdır (Spataru, 2010). Geliştirilecek uygulamaların mobil cihazlarda kapladığı depolama alanı ve kullandığı maksimum bellek boyutunun önemli bir kriter olduğu söylenebilir. Kod bloğu sayısı, iyi bir yazılım geliştirmek için dikkat edilmesi gereken kriterlerden biri olan basitlik ilkesi için belirleyicidir (Alakuş, Daş & Türkoğlu, 2017). Literatürdeki çalışmalar dikkate alındığında incelenen platformlarda geliştirilen uygulamaların kod bloğu sayısı, kapladığı depolama alanı boyutu, mobil cihazda kullandığı maksimum bellek boyutu değerleri açısından karşılaştırılmasına karar verilmiştir.

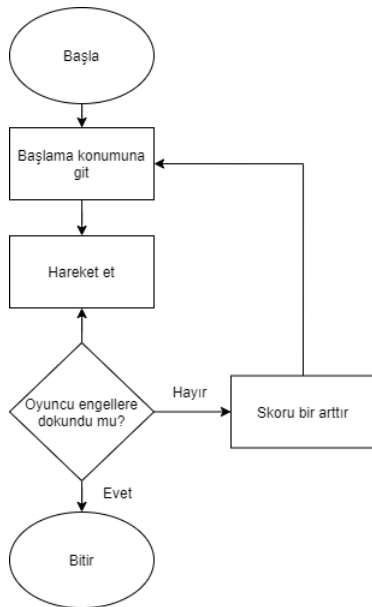
### **Mobil Uygulama**

Daha sonra platformların verdikleri çıktı üzerinden değerlendirebilmek adına 3 platformda da aynı uygulama geliştirilmiştir. Karşılaştırma için flappy bird benzeri basit bir oyun yapılmıştır. Flappy bird oyununun yayınlandığı tarihlerde 10 milyonun üstünde kullanıcı tarafından kullanılıyor olması sebebiyle ilgi çekici bir uygulama olduğu düşünülmüş ve kodlanması sırasında koşullar, döngüler ve değişkenler gibi programlama temel konularını basit düzeyde içermesi açısından programlama eğitimi için uygun bir örnek olacağı düşünülmüştür. Uygulama oyuncu ve engel akış diyagramı şekil 1 ve şekil 2'de gösterilmiştir.

řekil 1. Oyuncu Akıř Diyagramı



řekil 2. Engel Akıř Diyagramı



Belirlenen oyun üç ortamda da geliřtirilmiřtir. Geliřtirilen uygulamalar sanal bir Android cihaz üzerine kurularak denenmiřtir.

## BULGULAR

Programlama eğitiminde kullanılabilecek blok tabanlı mobil uygulama geliştirme platformları olan App Inventor, Thunkable ve Kodular iki aşamalı olarak karşılaştırılmıştır. İlk aşama olarak incelenen platformlar belirlenen kriterler göz önünde bulundurularak incelenmiş ve elde edilen bulgular Tablo 1’de gösterilmiştir.

Tablo 1. Mobil Uygulama Geliştirme Ortamları Genel Özellikler Tablosu

	App inventor	Thunkable	Kodular
Erişim adresi	appinventor.mit.edu	thunkable.com	kodular.io
Uygulama geliştirilen işletim sistemi	Android	IOS ve Android	Android
Ücret durumu	Ücretsiz	Ücretsiz ve Ücretli	Şarta bağlı ücretsiz
Türkçe Dil desteği	Yok	Yok	Var
Emülatör	Var	Var	Yok
Giriş eğitimi	Var	Var	Var
Topluluk desteği	Var	Var	Var

Belirlenen mobil uygulama ortamları uygulama geliştirilen işletim sistemi bakımından incelendiğinde üç uygulamanın da Android işletim sistemi için uygulama geliştirilebildiği fakat IOS işletim sistemi için uygulama geliştirilebilen tek ortamın Thunkable olduğu görülmüştür.

Mobil uygulama geliştirme ortamları için belirlenen diğer bir özellik olan ücret durumudur. App inventor tamamen ücretsiz iken Thunkable ortamında ücretsiz sürüme ek olarak mobil web uygulaması geliştirme, sınırsız özel proje, 1 gb saklama alanı gibi özellikler sunan ücretli versiyonu bulunmaktadır. Kodular ise uygulamalara reklam eklenmediği sürece ücretsiz kullanım sunmaktadır.

İncelenen ortamların hepsinde giriş eğitimi ve topluluk desteği bulunduğu görülmüştür. İncelenen mobil uygulama ortamlarından dil desteği açısından incelenmiş Kodular platformu Türkçe dil desteği ile toplam 19 farklı dil desteği sağladığı görülmüştür. Kodular, bazı bileşen ve bloklarda eksiklikler olmakla birlikte uygulama bileşenlerini ve kod bloklarını Türkçe olarak görüntüleyebilmemizi sağlamaktadır. Platformu tanıtım videosunda Türkçe altyazı bulunmakta fakat uygulama bileşenleri ve kod blokları ile ilgili detaylı bilgi edinebileceğimiz dokümanlar İngilizce olarak sunulmaktadır. Bileşen ve kod blokları anlatımı resimlerle desteklenmiştir. App Inventor platformu 15 farklı dil desteği sunarken Türkçe dil desteği bulunmamaktadır. App Inventor platformunun sunduğu bileşenler ve bloklar ile ilgili öğretici dokümanlar ise İngilizce olarak yayımlandığı görülmektedir. Platformda örnek projelerin yapılış aşamaları ve uygulamalarına erişilebilmekte ve galeri üzerinden diğer kullanıcılar tarafından paylaşılan uygulamalara erişilip kod blokları ve bileşenleri incelenebilmektedir. Öğretici dokümanlar resim, hareketli resim ve videolarla desteklenmiştir. Ek kaynaklar kısmında ise App inventor online kursları, tez çalışmaları gibi içeriklere erişilebilmektedir. Thunkable platformu için İngilizce dışında dil desteği bulunmadığı görülmüştür. Thunkable dokümantasyon bölümünde yeni başlayanlar için örnek projeler ve örnek oturum açma ekranları dışında genel proje galerisi ile diğer kullanıcıların yaptığı projelerde görüntülenebilmektedir. Kullanıcı arayüzü, bileşenler ve kod blokları özellikleri ve kullanımı hareketli resim ve videolarla desteklenmiştir. App inventor ve Thunkable platformlarında uygulamaların test edilebileceği emülatör bulunmakta iken Kodular ortamında olmadığı görülmüştür.

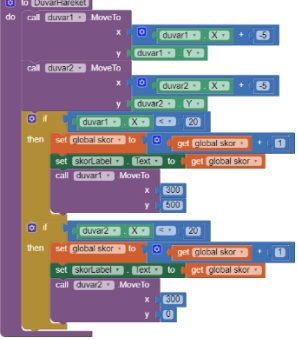
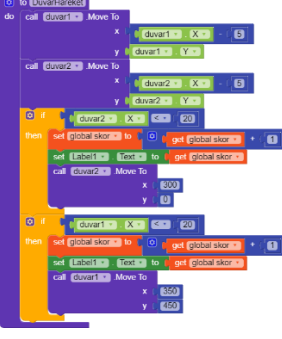
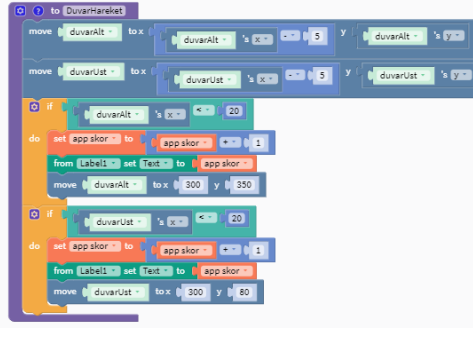
İkinci aşama olarak geliştirilen oyun App inventor, Thunkable ve Kodular platformlarında kodlanmıştır. Geliştirilen uygulamalar sanal bir Android cihaza kurularak test edilmiş ve test sonucunda elde edilen bulgular tablo 3’de gösterilmiştir.

Tablo 2. Uygulama Bilgileri Tablosu

	App inventor	Thunkable	Kodular
Kod bloğu sayısı	70	74	70
Depolama alanı kullanımı	10,36 mb	80,65 mb	15,99 mb
Maksimum Bellek Kullanımı	50 mb	195 mb	52 mb

Mobil uygulama geliştirme platformları kod bloğu yapısı ve kullanımı açısından oldukça benzer olmalarına rağmen Thunkable kod bloğu sayısı olarak diğer platformlara yakın olmakla birlikte en yüksek kod bloğu sayısına sahiptir. App inventor ve Kodular 70'er blok kullanılmışken Thunkable platformunda örnek uygulamayı geliştirebilmek için 74 blok kullanıldığı görülmektedir. App inventor ile Kodular platformlarının blok yapıları oldukça benzer iken Thunkable platformunda kod bloklarının biçimsel olarak farklılaştığı ve arayüz tasarım ekranı kullanımı açısından farklılaştığı görülmüştür. Tablo 4'de örnek uygulamada kullanılan örnek bir fonksiyon her bir platform için örnek olarak gösterilmiştir.

Tablo3. Örnek Kod Blokları

App inventor	Kodular	Thunkable
		

Geliştirilen uygulamaların mobil cihazda kapladığı depolama alanları incelendiğinde Thunkable platformunda geliştirilen uygulamanın en yüksek depolama alanı kaplayan uygulama olduğu görülmektedir. Thunkable platformunda geliştirilen uygulama, en düşük depolama alanı kaplayan App inventor platformunda geliştirilen uygulamadan 7,78 kat daha fazla yer kaplamaktadır. Kodular platformu ise 15,99 mb boyut ile App inventor platformunda geliştirilen uygulamadan 1,54 kat daha fazla alan kapladığı görülmektedir.

Geliştirilen uygulamaların maksimum bellek kullanımı incelendiğinde Thunkable ile geliştirilen uygulamanın diğer platformlarda geliştirilen uygulamalardan yaklaşık olarak 4 kat fazla bellek kullanımına sahip olduğu görülmektedir. App inventor ve Kodular platformlarında geliştirilen uygulamaların bellek kullanımı birbirine yakın olmakla birlikte en az bellek kullanımına sahip olan uygulamanın App inventor platformunda geliştirildiği görülmüştür.

## TARTIŞMA ve SONUÇ

Mobil cihazların yaygınlaşması ile mobil programlama eğitimi ihtiyacının da arttığı söylenebilir. Aslan ve Aslan (2013) yaptıkları çalışmada mobil yazılımlara olan talebin gün geçtikçe artması ile ülkemizde de mobil yazılım uzmanı ihtiyacının arttığını bu sebeple de üniversite bilgisayar programcılığı bölümlerine mobil programlama derslerinin eklenmesi gerektiğini belirtmişlerdir. Bu çalışmada mobil programlama eğitimi için kullanılacak blok tabanlı mobil uygulama geliştirme platformlarının karşılaştırılması yapılmıştır.



Belirlenen platformları uygulama geliştirilen işletim sistemi açısından değerlendirdiğimizde Thunkable platformu geliştirilen uygulamanın hem ios hem de android işletim sistemleri için çıktı sunmaktadır. İşletim sistemlerinin farklılıklarından dolayı farklı işletim sistemleri için uygulama geliştirmek isteyen yazılım geliştiricilerin her bir işletim sistemi için harcadıkları çabayı tekrarlamaları gibi zorluklar bulunmaktadır (Çiçek, Garousi & Tarhan). Thunkable platformunun iki işletim sistemi için çıktı vermesinin verimlilik açısından önemli olduğu söylenebilir.

Kodular platformunda emülatör desteğinin olmadığı görülmüştür. Emülatör, kullanıcıların istedikleri telefonları sanal cihaz olarak kullanıp bu cihazlar üzerinde uygulamalarını test etme imkânı sunar (Dandıl, Öztürk, Ekşi & Çakıroğlu, 2015). Geliştirilen uygulamaların telefona yüklenmeden emülatör ile denenebilmesi zaman ve yapılan işten tasarruf sağladığı söylenebilir. Kodular platformunda kullanıcıların uygulamaları denemek için üçüncü parti bir yazılım kurması ya da telefonlarının olması gerekmektedir. Bu durumun bir dezavantaj olduğu söylenebilir.

Belirlenen platformların hepsinde giriş eğitimi ve topluluk desteği bulunmaktadır. App inventor ve Thunkable platformlarında giriş eğitimi bölümlerinde hazırlanan dokümanlar resim, hareketli resim ve videolar ile desteklenmektedir. Video anlatımlar, göze ve kulağa aynı anda hitap eden iletişim araçları olması dolayısıyla öğrenenlerin ilgi ve dikkatlerini çekme düzeyleri yüksektir (Orhan ve Akkoyunlu, 1999). Rüzgâr (2005) yaptığı çalışmada video ile desteklenen öğretimlerin öğrenci başarılarını olumlu etkilediği sonucuna varmıştır. Yapılan çalışmalar göz önünde bulundurulduğunda App inventor ve Thunkable platformlarının eğitim dokümanlarını videolar ile desteklemiş olmaları öğrenenlere olumlu etki edeceği yönünde yorumlanabilir. Örnek projelerin ve kullanıcılar tarafından yapılan projelerin diğer kullanıcılar ile paylaşılıyor olması diğer kullanıcıların sorunları çözme yöntemlerini görmelerini ve kendi sorunlarının çözümlerine uyarlayabilmeleri açısından yararlı olacaktır.

Kod sayısı açısından geliştirilen oyunlar incelendiğinde en az kod bloğu sayısının App inventor ve Kodular platformlarında kullanılmış olduğu görülmektedir. Teknik anlamda geliştirilen yazılımdaki kod sayısı ve fonksiyon sayısı gibi özelliklere bağlı olan basitlik uygulamanın iyi olarak kabul edilebilmesi için en önemli amaçların başında gelir (Alakuş et al., 2017). En az kod bloğu sayısının App inventor ve Kodular platformlarında kullanılmış olması bu platformların daha basit uygulama geliştirilmesine uygun olduğu şeklinde yorumlanabilir. Kert ve Uğraş (2009)'a göre öğrencileri programlamanın karmaşıklığından uzaklaştırarak problemi anlama, analiz yapma, değerlendirme ve yaratıcılık boyutlarına odaklanabilmelerini sağlayacak yazılımların kullanılması önemlidir. Programlamaya yeni başlayan kişiler ile çalışacak eğitimciler için basit kod yapıları oluşturulmasına olanak sağlayan platformların tercih edilmesi eğitimin daha etkili olması açısından önemli olduğu söylenebilir.

Belirlenen platformların geliştirilen uygulamaların bellek kullanımı ve depolama alanı açısından değerlendirildiğinde App inventor platformunda geliştirilen uygulamanın en düşük depolama alanı kullanımına sahip iken Thunkable platformunda geliştirilen uygulamanın App inventor'a göre 7,78 kat daha fazla depolama alanı kullanmaktadır. Willocx, Vossaert ve Naessens (2015) çalışmalarında mobil uygulamaların değerlendirilmesinde ilk olarak, yüklü uygulamanın cihazda aldığı alan ölçüleceği ve uygulamanın kapladığı alan özellikle sınırlı kaynaklara sahip düşük kaliteli cihazlar için önemli olduğu görüşünü bildirmişlerdir. Pekişerek, Sağlam ve Ustun (2020) tarafından yapılan çalışmada da, geliştirilen bir mobil uygulamanın kullandığı bellek düzeyinin önemi vurgulanmıştır. Bu bağlamda, Thunkable platformunda geliştirilen uygulamanın yaklaşık 8 kat fazla alan kaplaması, App inventor platformunun tercih edilme olasılığını arttırmaktadır. Maksimum bellek kullanımı değerlerinde en az bellek kullanımı App inventor platformunda geliştirilmiş uygulamada iken en yüksek bellek kullanımı Thunkable platformundadır. Thunkable platformunda geliştirilen uygulamanın maksimum bellek kullanımı App inventor platformunda geliştirilen uygulamadan 3,9 kat daha fazladır. Küçük çaplı uygulamalar için göz ardı edilebilecek bir fark olarak düşünülse de Thunkable'da daha kapsamlı uygulamalar geliştirmek isteyen programcılar için sorun teşkil edeceği yorumunda bulunulabilir. Mobil cihazlar, sınırlı bellek kapasitesi, işlem gücü ve güç rezervi gibi kısıtlılıklara sahiptir (Spataru, 2010). Bu kısıtlılıklar göz önünde bulundurulduğunda Thunkable platformunda geliştirilen uygulamaların düşük kaliteli

cihaz kullanıcıları tarafından tercih edilme olasılığının düşük olduğu yorumunda bulunulabilir. Bu durum Thinkable platformunun en büyük dezavantajı olarak görünmektedir. Eğitimsel açıdan düşünüldüğünde öğrencilerin geliştirdikleri uygulamaların daha fazla kullanıcı tarafından tercih edilmesi öğrencilerin mobil uygulama geliştirme motivasyonlarını arttıracaktır.

Çalışmada App inventor, Thinkable ve Kodular platformları özellikleri açısından ve geliştirilen uygulamaların fiziksel kaynak kullanımı açısından karşılaştırılmıştır. Uygun yazılımın seçilmesi verilecek olan eğitimin kalitesini arttırmada büyük önem taşır (Özmen & Varol). Bu karşılaştırma sonucunda elde edilen bilgilerin eğitimcilere mobil programlama eğitimi için platform seçiminde yol göstereceği düşünülmektedir.

## KAYNAKÇA

- Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *Elementary Education Online*, 13(1), 1-4.
- Alakuş, T. B., Daş, R., & Türkoğlu, İ. (2017, Eylül). Yazılım geliştirme süreçlerinin analizi: Zorluklar, tasarım prensipleri ve tekniksel yaklaşımlar. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-10). IEEE.
- Bales, S. (2018). *Build Android apps without Coding: Get started with Android apps using Thinkable-MIT app Inventor*. Independently published. ACM, New York.
- Baz, F. Ç. (2018). Çocuklar için kodlama yazılımları üzerine karşılaştırmalı bir inceleme. *Current Research in Education*, 4(1), 36-47.
- Bhavani, D. D., & Panjala, N. K. (2017). Smart Frame-A Location Sensing Picture Frame using IOT. *International Journal on Recent and Innovation Trends in Computing and Communication*, 5(8), 235-239.
- Büyüköztürk, Ş., Akgün, Ö. E., Demirel, F., Karadeniz, Ş., & Çakmak, E. K. (2008). *Bilimsel araştırma yöntemleri*. Ankara: Pegem Yayınları.
- Dalmaso, I., Datta, S. K., Bonnet, C., & Nikaein, N. (2013, Temmuz). Survey, comparison and evaluation of cross platform mobile application development tools. In *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)* (pp. 323-328). IEEE.
- Dandil, E., Öztürk, İ., Ekşi, Z., & Çakıroğlu, M. (2015). Mobil Cihazlar İçin İlk Yardım Paketi Uygulaması. *Akademik Platform Mühendislik ve Fen Bilimleri Dergisi*, 3(1), 16-20.
- De Souza, A. D. M., & da Silva, E. d. O. (2019). Desenvolvimento de um Aplicativo Móvel para Gestão de Pedidos do Banco Alimentario de La Plata. *Caderno de Estudos em Engenharia de Software*, 1(1). 1-15.
- Fai, L. W., & Audah, L. (2017). In campus location finder using mobile application services. In *AIP Conference Proceedings* (Vol. 1883, No. 1, p. 020022). AIP Publishing LLC.
- Ichsan, I. Z., Sigit, D. V., Miarsyah, M., Nuri, A. R. U., Sajidan, S., Oetomo, D., ... & Yasin, M. (2019). Supplementary book of green consumerism: an innovation of environmental learning based on hots. *Management*, 145, 135-144.
- João, P., Nuno, D., Fábio, S. F., & Ana, P. (2019). A Cross-analysis of Block-based and Visual Programming Apps with Computer Science Student-Teachers. *Education Sciences*, 9(3), 181.
- Kert, S. B., & Uğraş, T. (2009, Ağustos). *Programlama eğitiminde sadelik ve eğlence: Scratch örneği*. In The First International Congress of Educational Research, Çanakkale, Türkiye. [https://www.academia.edu/download/34971233/Kert\\_Ugras\\_Scratch.pdf](https://www.academia.edu/download/34971233/Kert_Ugras_Scratch.pdf)
- Kodular (2020, Mart). *Kodular docs*.<https://docs.kodular.io>
- Mıhçı, C. (2014). *Programlama eğitiminde görsel blok programlama ve mobil uygulama geliştirme araçlarının karşılaştırılması* (Yayınlanmamış yüksek lisans tezi). Marmara Üniversitesi, İstanbul.
- MIT App Inventor (2020, Haziran). *About us*. <https://appinventor.mit.edu/about-us>
- Orhan, F., & Akkoyunlu, B. (1999). Uzaktan eğitim yaklaşımında temel eğitim 1. kademe öğretmenleri'nin video destekli hizmetiçi eğitimi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 17(17), 134-141.
- Özdiñç, F. (2015). Mobil Programlama Eğitiminde Yeni Yaklaşımlar: App Inventor Örneği. *Akademik Bilişim Konferansı-17*, Eskişehir.
- Özmen, B., & Varol, F. (2012). Uzman, aile ve öğretmen gözü ile eğitim yazılımları: *EYADES. Education Sciences*, 7(1), 322-330.

- Pekyürek, M., Sağlam, Z., Ustun, A. (2020). MIT App Inventor ve Android Studio Kullanılarak Tasarlanmış Mobil Uygulamanın Performans Karşılaştırması. *Bilgi ve İletişim Teknolojileri Dergisi*, 2 (2), 161-181.
- Pinto, A. and Escudeiro, P. (2014, Haziran). The Use Of Scratch For The Development Of 21 St Century Learning Skills İn ICT. In *information Systems and Technologies (CISTI), 2014 9th Iberian Conference on, IEEE*, Barcelona, Spain, 18-21 June, 1-4.
- Pokress, S. C., & Veiga, J. J. D. (2013). MIT App Inventor: Enabling personal mobile computing. *arXiv preprint arXiv:1310.2830*.
- Robianto, R. (2020). Analysis Of Android Based Mobile Blocking Application Design Using Mit App Inventor. *Jurnal Ipteks Terapan*, 14(1), 1-11.
- Roy, K. (2012). App inventor for android: report from a summer camp. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 283-288).
- Rüzgar, B. (2005). Bilginin Eğitim Teknolojilerinden Yararlanarak Eğitimde Paylaşımı. *TOJET: The Turkish Online Journal of Educational Technology*, 4(3), 114-119.
- Saygıner, Ş., & Tüzün, H. (2017, Mayıs). Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri. *I. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, Mayıs, 78-90.
- Siegle, D. (2020). There's an App for That, and I Made It. *Gifted Child Today*, 43(1), 64-71.
- Smutný, P. (2011). Visual programming for smartphones. In *2011 12th International Carpathian Control Conference (ICCC)* (pp. 358-361). IEEE.
- Sönmez Çiçek, S. (2018). *Uygulamaların Mobil Ve Masaüstü Sürümlerinin Kod Tabanlı Karşılaştırması: Keşifsel Bir Çalışma* (Yayımlanmamış yüksek lisans tezi), Hacettepe Üniversitesi, Ankara.
- Spataru, A. C. (2010). *Agile development methods for mobile applications* (Yayımlanmamış yüksek lisans tezi). University of Edinburgh, Edinburgh.
- Tavşancıl, E. ve Aslan, E. (2001). *İçerik analizi ve uygulama örnekleri*. İstanbul: Epsilon Yayıncılık.
- Thunkable (2020, Mayıs). *Thunkable About Us*.<https://thinkable.com/#/careers>
- Willox, M., Vossaert, J., & Naessens, V. (2015, Haziran). A quantitative assessment of performance in mobile app development tools. In *2015 IEEE International Conference on Mobile Services* (pp. 454-461). IEEE.
- Zikri, A. (2020). *Rancang bangun sistem penyiraman tanaman otomatis berbasis Raspberry Pi 3 dengan memanfaatkan thingspeak dan interface android sebagai kendali* (Yayımlanmamış lisans tezi) Bilim ve Teknoloji Fakültesi, Syarif Hidayatullah Devlet İslam Üniversitesi, Cakarta.