

T.C.
ANAkkALE ONSEKİZ MART ÜNİVERSİTESİ
EĐİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĐRETİM TEKNOLOJİLERİ EĐİTİMİ ANABİLİM DALI

SCRATCH PROGRAMLAMA DİLİ EĐİTİMİNE YÖNELİK BİR MOBİL
UYGULAMANIN GELİŐTİRİLMESİ

YÜKSEK LİSANS TEZİ

Emre YILDIRIM

ANAkkALE

TEMMUZ, 2017

T.C.
Çanakkale Onsekiz Mart Üniversitesi
Eğitim Bilimleri Enstitüsü
Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı
Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı

Scratch Programlama Dili Eğitimine Yönelik Bir Mobil Uygulamanın Geliştirilmesi

Emre YILDIRIM
(Yüksek Lisans Tezi)

Danışman
Prof. Dr. Mehmet Ali SALAHLI

Çanakkale
Temmuz, 2017

Taahhütname

Yüksek Lisans Bitirme Tezi olarak sunduğum **“Scratch Programlama Dili Eğitimine Yönelik Bir Mobil Uygulamanın Geliştirilmesi”** adlı çalışmanın, tarafımdan, bilimsel ahlak ve değerlere aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yaparak yararlanmış olduğumu belirtir ve bunu onurumla doğrularım.

12/07/2017

Emre YILDIRIM

Çanakkale Onsekiz Mart Üniversitesi

Eğitim Bilimleri Enstitüsü

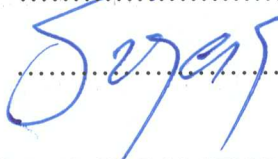
Onay

Emre YILDIRIM tarafından hazırlanan çalışma, 12/07/2017 tarihinde yapılan tez savunma sınavı sonucunda jüri tarafından başarılı bulunmuş ve Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Referans No : 10150721

	Akademik Unvan	Adı SOYADI	İmza
Danışman	Prof. Dr.	Mehmet Ali SALAHLI	
Üye	Doç. Dr.	Nilgün TOSUN	
Üye	Yrd. Doç. Dr.	Serkan İZMİRLİ	

Tarih: 10.08.2017

İmza: 

Prof. Dr. Salih Zeki GENÇ

Enstitü Müdürü

Önsöz

Tez çalışmam boyunca kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan, değerli ve kıymetli danışman hocam sayın Prof. Dr. Mehmet Ali SALAHLI' e teşekkür ve saygılarımı sunarım.

Çalışmalarım boyunca yardımlarını hiç esirgemeyen başta Ömer KIRMACI ve diğer değerli arkadaşlarıma ayrı ayrı teşekkür ederim.

Benim bugünlere gelmemde büyük pay sahibi olan, bugüne kadar maddi ve manevi olarak desteklerini eksik etmeyen, haklarını asla ödeyemeyeceğim kıymetli aileme minnetlerimi sunuyorum.

Son olarak, tez çalışmasının uygulama aşamasında önemli katkıları bulunan, yaptığı büyük fedakarlık, sabır ve anlayışla bana her zaman destek olan sevgili öğretmen eşim Tuba SAYGILI YILDIRIM'a sonsuz teşekkür ederim.

Özet

Scratch Programlama Dili Eğitime Yönelik Bir Mobil Uygulamanın Geliştirilmesi

Bu araştırmanın amacı, ortaokul öğrencilerinin programlama mantığını daha iyi anlamaları ve programlama becerilerinin geliştirilmesi için Scratch programlama diline yönelik bir mobil uygulaması geliştirmektir.

Mobil uygulama Scratch programlama dilinin öğretimi ile ilgili iki sorunun çözümüne yönelik geliştirilmiştir. Birinci sorun, okullarda yaygın olarak kullanılan Scratch uygulamasının eğitsel yönden bazı eksiklerinden kaynaklanmaktadır. İkinci sorun ise Scratch uygulamasının mobil ortamlarda kullanılmaması ile ilgilidir.

Bu çalışma ADDIE tasarım modeli aşamaları kullanılarak gerçekleştirilmiştir. Modelin analiz aşamasında, Scratch programlama dili eğitiminde kullanılan programlama arayüzünün eksiklikleri analiz edilmiştir. Aynı zamanda araştırmanın çalışma grubu belirlenmiştir. Tasarım aşamasında ise yapılan analizler sonucunda mobil öğrenme uygulamasının ekran tasarımları ve Scratch programlama dili eğitiminin içeriği oluşturulmuştur. Geliştirme aşamasında belirlenen tasarımlar ışığında uygulamanın kodlaması yapılmıştır. Uygulama aşamasında, geliştirilen mobil uygulama, ilk önce pilot uygulama olarak Scratch programlama dili eğitimi alan 7. Sınıfta öğrenim gören 10 öğrenci tarafından 1 hafta boyunca test edilmiştir. Ardından gerçek uygulamaya geçilmiştir.

Araştırmada ön test son test kontrol gruplu yarı deneysel model uygulanmıştır. Araştırmanın gerçek uygulamasında çalışma grubunu Kırklareli İstiklal Ortaokulunda öğrenim gören 6. sınıf düzeyinde toplam 94 öğrenci oluşturmaktadır. Mobil öğrenme uygulamasını kullanan deney grubunda 41 öğrenci, kontrol grubunda ise 53 öğrenci bulunmaktadır. Öğrenciler mobil öğrenme uygulamasını 5 hafta boyunca kullanmışlardır. Değerlendirme aşamasında, araştırmada veri toplama aracı olarak araştırmacının geliştirdiği Programlama Becerileri Başarı Testi kullanılmıştır. Araştırmanın verilerini analiz etmek için Bağımsız Örneklem t-testi kullanılmıştır.

Araştırmanın sonunda elde edilen bulgulara göre Scratch programlama dili mobil öğrenme uygulamasının öğrencilerin programlama becerileri üzerinde anlamlı bir farklılık oluşturduğu ve programlama eğitime katkı sağladığı görülmüştür.

Anahtar Kelimeler: mobil öğrenme, programlama eğitimi, scratch.

Abstract

Development of a Mobile Application for Scratch Programming Education

The aim of this research is to develop a mobile application for the Scratch programming language in order to better understand the programming logic of secondary school students and to improve their programming skills.

The mobile application has been developed to solve two problems related to teaching Scratch programming language. The first problem stems from some shortcomings in the educational direction of the Scratch application commonly used in schools. The second problem is that Scratch is not used in mobile environments.

This work was carried out using the ADDIE design model stages. In the analysis phase of the model, the shortcomings of the Scratch application used to do programming using the Scratch programming language have been analyzed. At the same time, the study group of the study was determined. In the design stage, the screenings of mobile learning application and the content of the language training of Scratch programming were created as a result of the analyzes made. During the development phase, the design of the application was coded in the light of the design. In the implementation phase, the developed mobile application was first tested by 10 7th grade students who were trained in Scratch programming language as pilot application for one week. Then the real application was implemented.

In the study pretest posttest quasi experimental model was applied. In the real implementation of the study, the study group constitutes a total of 94 students at the 6th grade level who are studying at Kırklareli İstiklal Ortaokulu. There are 41 students in the experiment group using mobile learning application and 53 students in the control group. The students used the mobile learning application for 5 weeks. During the evaluation stage, the Programmer's Skills Achievement Test developed by the researcher was used as the data collection tool in the research. The independent samples t-test was used to analyze the data of the study.

As a result of the findings obtained at the end of the research, it was seen that Scratch programming language mobile learning application made a significant difference on programming skills and contributed to programming education.

Keywords: mobile learning, programming education, scratch.

İçindekiler

Onay	i
Önsöz.....	ii
Özet	iii
Abstract	iv
İçindekiler.....	v
Tablolar Listesi.....	vii
Şekiller Listesi.....	viii
Kısaltmalar Listesi.....	ix
Bölüm I: Giriş.....	1
Problem Durumu	1
Araştırmanın Amacı	5
Araştırma alt amaçları.	5
Araştırmanın Önemi	6
Araştırmanın Varsayımları	6
Araştırmanın Sınırlılıkları.....	7
Bölüm II: Kavramsal Çerçeve.....	8
Programlama Eğitimi.....	8
Programlama Öğretimi	9
Programlama Eğitiminde Kullanılan Grafik Tabanlı Programlama Dilleri	11
App inventor.....	11
Alice.	13
Scratch	13
Scratch’ın genel yapısı ve kullanımı.	13
Scratch ile ilgili yapılan çalışmalar.	19
Mobil Öğrenme.....	22
Mobil öğrenme ortamının özellikleri.	22
Mobil öğrenme ile ilgili yapılan çalışmalar.	23
Bölüm III: Yöntem	28
Araştırma Modeli.....	28
Çalışma Grubu.....	28

Veri Toplama Araçları.....	29
Programlama becerisi başarı testi.....	29
Verilerin Analizi	32
Bölüm IV: Bulgular.....	33
Scratch Programlama Diline Yönelik Mobil Öğrenme Uygulamasının Geliştirilmesi	33
Analiz aşaması.....	33
Tasarım aşaması.	35
Geliştirme aşaması.	46
Uygulama aşaması.....	48
Değerlendirme aşaması.	50
Öğrencilerin Programlama Becerileri Ön Test ve Son Test Başarı Puanlarına İlişkin Bulgular	50
Ön test programlama becerileri başarı puanlarına ilişkin bulgular.	50
Son test programlama becerileri başarı puanlarına ilişkin bulgular.	52
Bölüm V: Sonuç, Tartışma ve Öneriler.....	54
Sonuç ve Tartışma	54
Öneriler.....	55
Kaynakça.....	57
Ekler	65
Ek 1. Programlama Becerisi Başarı Testi.....	65
Ek 2. Belirtke Tablosu	67
Ek 3. Scratch Eğitim Videoları İçin İzin Yazışması.....	68
Ek 4. Araştırma İzni.....	69
Özgeçmiş.....	70

Tablolar Listesi

Tablo 1. Başarı Testi Madde Güçlük ve Madde Ayırt Edicilik İndeksleri.....	31
Tablo 2. Scratch ile Kodlama Eğitimi Kazanımlarına Göre Etkinliklerin Sınıflandırılması ...	37
Tablo 3. Etkinliklere Göre Kazanılan Rütbe Seviyeleri.....	45
Tablo 5. Ön Test Programlama Becerileri Başarı Puanlarına İlişkin Betimsel İstatistik ve Shapiro-Wilk Normal Dağılım Testi Sonuçları.....	51
Tablo 6. Ön Test Programlama Becerileri Başarı Puanlarına İlişkin Bağımsız Örneklem t-Testi Sonuçları.....	51
Tablo 7. Son Test Programlama Becerileri Başarı Puanlarına İlişkin Betimsel İstatistik ve Shapiro-Wilk Normal Dağılım Testi Sonuçları.....	52
Tablo 8. Son Test Programlama Becerileri Başarı Puanlarına İlişkin Bağımsız Örneklem t-Testi Sonuçları.....	53

Şekiller Listesi

Şekil 1. App Inventor Ortamının Tasarım Ekranı	12
Şekil 2. App Inventor Ortamının Kodlama Ekranı	12
Şekil 3. Alice Grafik Tabanlı Programlama Ortamı	13
Şekil 4. Scratch Çevrimdışı Programlama Uygulaması	15
Şekil 5. Scratch Çevrimiçi Programlama Ortamı	15
Şekil 6. Scratch Çevrimiçi Paylaşım Ortamı	16
Şekil 7. Scratch Çevrimdışı Programlama Uygulamasının Bölümleri	17
Şekil 8. Scratch Kod Bloklarının Sürüklenmesi	18
Şekil 9. Yeni Karakterin Eklenmesi	18
Şekil 10. Başarı Testinin Geliştirilme Aşamaları	32
Şekil 11. ADDIE Tasarım Modeline Göre Mobil Öğrenme Uygulamasının Geliştirilmesi	33
Şekil 12. Tasarım Aşaması	35
Şekil 13. Mobil Öğrenme Uygulamasının Ana Ekran Görüntüsü	36
Şekil 14. Mobil Öğrenme Uygulamasının Bileşenlerinin Oluşturması	36
Şekil 15. Etkinlikler Bileşeninin Ana Ekranı	37
Şekil 16. Başlatılan Bir Etkinliğin İlk Giriş Ekranı: Hedef ve Eylem	39
Şekil 17. Etkinlik Ekranı	39
Şekil 18. Öğrenciyi Doğru Algoritma Kurması İçin Verilen İpucu Ekran Örneği	40
Şekil 19. Etkinlikte Başarılı Olunması Durumunda Öğrenciye Verilen Dönüt Ekranı	40
Şekil 23. Kod Blokları Kategorileri Menüsü	43
Şekil 26. Yardım Menüsü İçeriği	46
Şekil 27. Android Studio Yazılım Geliştirme Platformu	47
Şekil 28. Kod Blokları Ders Videosu Örneği	47
Şekil 29. Uygulama Aşamasının Şekilsel Gösterimi	49

Kısaltmalar Listesi

ADDIE	: Analysis, Design, Development, Implementation, Evaluation
ALGOL	: Algorithmic Language
BT	: Bilişim Teknolojileri
COBOL	: Common Business Oriented Language
CSS	: Cascading Style Sheets
DVR	: Digital Video Recorder
FORTRAN	: Formula Translation
HTML	: Hypertext Markup Language
IOS	: iPhone Operating System
ITU	: International Telecommunication Union
LMS	: Learning Management System
MEB	: Milli Eğitim Bakanlığı
MEGEP	: Mesleki Eğitim ve Öğretim Sistemini Geliştirme Projesi
MIT	: Massachusetts Institute of Technology
OECD	: Organisation for Economic Co-operation and Development
PC	: Personal Computer
PDA	: Personal Digital Assistant
PISA	: Programme for International Student Assessment
SPSS	: Statistical Package for Social Sciences
SQL	: Structured Query Language

Bölüm I: Giriş

Problem Durumu

Programlama eğitimi öğrencilerin problem çözme, eleştirel düşünme, mantıksal düşünme, kritik düşünme ve yaratıcı düşünme gibi üst bilişsel becerilerini geliştirmektedir (Calder, 2010; Clements ve Gullo, 1984; Fessakis, Gouli ve Mavroudi, 2013; Papert, 1980; Park, 2013; Siegle, 2009). Bu nedenle, programlama eğitimini gereken seviyede alan ve programlama mantığını anlayan öğrenciler diğer derslerde de başarılı olabilmektedirler. Valanides (1997) yaptığı araştırmada, mantıksal düşünme becerisinin öğrencilerin fen ve matematik derslerinde başarılı olması için önemli bir etken olduğunu belirtmektedir. Seferoğlu ve Akbıyık (2006), eleştirel düşünme becerileri yüksek olan öğrencilerin fen ve matematik gibi derslerde daha başarılı olduğunu belirtmişlerdir. Bir başka çalışmada ise, problem çözme becerisinin öğrencilerin matematik başarıları üzerinde önemli rol oynadığı sonucuna varılmıştır (Özsoy, 2005). Ancak, ülkemizde eğitim gören öğrencilerin uluslararası sınavlardaki başarıları göz önüne alındığında, öğrencilerimizin üst düzey beceri bakımından yeterli seviyede olmadığı görülmektedir. Ekonomik Kalkınma İşbirliği Örgütü (OECD) tarafından yapılan ve ülkelerin eğitim kalitesini ortaya koyan PISA sınavı sonuçlarına göre öğrencilerimiz fen, matematik ve okuma becerilerindeki başarıları bakımından alt sıralarda yer almaktadır (MEB, 2017; OECD, 2017). Bu nedenle, ülkemizdeki öğrencilerin üst düzey becerilerini geliştirmek için farklı eğitim yöntem ve araçlarının kullanılması gerekmektedir. Bu bağlamda programlama eğitiminde de yeniliklere ihtiyaç duyulmaktadır.

Dünyanın çeşitli ülkelerinde öğrencilerin programlama becerilerinin geliştirmesine yönelik ulusal ve bölgesel projeler yapılmaktadır. Bu projelerin sonuçları okul müfredatlarına yansımaktadır (Balanskat ve Engelhardt, 2014). Örneğin, Estonya’da programlama eğitiminin 2012 yılından itibaren ilkokuldan başlanması ile ilgili proje gerçekleştirilmiştir (Demirer ve Sak, 2016). Bulgaristan’da 9. sınıflar için bilişim derslerinde haftada 2 saat zorunlu olarak

programlama eğitimi verilmektedir. Burada öğrencilerin matematiksel becerilerini geliştirmek esas hedef olarak belirlenmiştir. Kıbrıs Rum Kesiminde, ilköğretim ve ortaöğretimindeki bilişim derslerinde algoritma ve programlama zorunlu konular olarak öğretilmektedir. Ayrıca, bu alanlarda kendilerini geliştirmek isteyen öğrencilere ders dışında kurslar verilmektedir. Finlandiya’da da öğrencilerin programlama ve bilişim teknolojilerindeki becerilerinin geliştirilmesi desteklenmektedir. Bu ülkede 2016 yılından başlayarak 1-9. sınıflarda zorunlu olarak programlama eğitimi verilmektedir. Danimarka’da programlama eğitiminin 1-10. sınıflarda matematik ve fen derslerinde verilmesi ile ilgili girişimlerde bulunulmuştur (Balanskat ve Engelhardt, 2014). İngiltere, 2014 yılının Eylül ayından itibaren ilk ve ortaöğretim seviyesindeki devlet okullarında programlama eğitimini veren ilk Avrupa ülkesidir. Buna göre, öğrenciler 5 yaşında okula geldiklerinden itibaren programlama eğitimi almaya başlar ve 16 yaşına gelene kadar bu eğitimi almaya devam ederler. Örneğin, 7 yaşındaki öğrenciler yapılan algoritmaları adım adım izleyerek anlayabilir, basit programlar oluşturabilir ve hataları ayıklayabilir. Belçika ve İspanya da programlama eğitimini ders müfredatlarına ekleyen ülkeler arasındadır (Balanskat ve Engelhardt, 2014).

Ülkemizde de programlama eğitiminin küçük yaşlardaki öğrencilere verilmesine yönelik çalışmalar yapılmaya başlanmıştır (Kodlama Atölyesi, 2017; MEB, 2017). Ortaokul düzeyinde seçmeli olarak verilen “Bilişim Teknolojileri” dersi Milli Eğitim Bakanlığının (MEB) yapmış olduğu çalışmayla “Bilişim Teknolojileri ve Yazılım” dersi adıyla 5. ve 6. sınıflarda zorunlu 7. ve 8. sınıflarda seçmeli olarak güncellenmiş ve dersin öğretim programı içerisine de programlama eğitimi dahil edilmiştir. Bu eğitimle, öğrencilere temel düzeyde programlama becerileri kazandırmak amaçlanmaktadır (MEB, 2017).

Programlama eğitimi, yapısı itibariyle soyut kavramlar ve terimler içerdiğinden öğrenilmesi zordur (Eckerdal, 2009; Gomes ve Mendes, 2007; Hadjerrouit, 2008; Mannila ve diğerleri, 2006; Westcott, 2008). Bu nedenle, programlama eğitimini alan öğrenciler konuları

tam kavrayamıyor, bunun sonucunda ise motivasyonları düşüyor ve dersten başarısız oluyorlar (Erdoğan, Aydın ve Kabaca, 2008; Korkmaz, 2016). Bunun nedeni programlamaya yeni başlayan öğrencilerin karşılaştıkları karmaşık mantıksal yapılar içerisindeki algoritma hatalarıyla uğraşmak zorunda kalmalarıdır. Programlamaya giriş derslerinde başarısızlık veya programlamayı bırakma oranlarının % 15 ila % 30 olduğu tahmin edilmektedir (Fesakis ve Serafeim, 2009). Çünkü programlama yapmak için geliştirilen birçok ortam genellikle metin tabanlı sistemlerdir. Öğrenciler bu çeşit sistemlerde program geliştirmek adına binlerce kod öğrenmek zorunda kalabilmektedirler. Dolayısıyla metin tabanlı sistemlerde yazılan kodları okumak, yazmak, anlamak ve hatırlamak zor olabilmektedir (Ortiz, 1997). Öğrencilerin bu sorunlarını ortadan kaldırmak için programlama daha basit şekilde öğrenilebilir hale getirilmeye çalışılmalıdır.

Programlamanın öğrenilmesinde en önemli ve temel unsur öğrenciye programlamanın mantığını veya algoritma becerisini kazandırmaktır (Bennedsen ve Caspersen, 2008). Bu nedenle programlama eğitimini ilk defa alacak öğrenciler için grafik tabanlı programlama dillerine dayanan yazılımlar geliştirilmiştir. Grafik tabanlı programlama dilleri öncelikle bir öğrenme aracı olarak tasarlanmışlardır. Bunlar, öğrencilere programlamanın mantığını öğretebilecek yazılımlardır. Yeni başlayanlara eğlenceli ortamlar sunarlar ve gelecekte ileri seviye programlama becerisi kazandırmada yardımcı olurlar (Papert, 1993). Bu programlama dilleri aynı zamanda öğrencilerin C#, Visual Basic ve Java gibi modern programlama dillerini öğrenmeleri için bir temel oluşturmaktadır (Schwartz, Stagner ve Morrison, 2006). Ancak, ileri seviye kod yazdıracak kadar becerilere sahip değildirler. Logo, App Inventor, Small Basic ve Scratch günümüzde etkin olarak kullanılan ve iyi bilinen grafik programlama dilleridir. Bu programlama dillerinin içerisinde Scratch daha popüler olarak kullanıldığı söylenebilir.

Scratch, öğrencilerin grafiksel kod bloklarını sürükleyip bırak yöntemiyle kullanarak basit programlar oluşturmalarını sağlayan ve programlamanın mantığını öğretmek için kullanılan programlama ortamıdır. Bu ortam içerisinde nesneler, koşul ifadeleri, döngüler, değişkenler ve olaylar gibi kodlama konuları öğretilir. Scratch ücretsiz olarak indirilip kullanılabilir. Aynı zamanda Scratch, kendi web sitesinde öğrencilere videolar, forumlar ve bitmiş Scratch projeleri gibi öğrencilerin örnek alabileceği ve kendi projelerini paylaşabileceği bir ortam sunmaktadır (Scratch Web Sitesi, 2017). Böylece, Scratch programlamaya yeni başlayanlar öğrencilerin daha çok ilgilerini çeken ve onlar için daha kolay ve eğlenceli bir ortam sunmaktadır (Yükseltürk ve Altıok, 2016). Ayrıca ileri seviye kod yazmaları içinde kullanıcıları cesaretlendirmektedir. Kobsiripat (2014) çalışmasında Scratch kullanarak ilköğretim öğrencilerinin yaratıcılıklarını incelemiştir. Çalışma sonunda Scratch'ın öğrencilerin yaratıcılıklarını olumlu yönde etkilediği sonucuna varılmıştır. Bir başka çalışmada (Taylor, Harlow ve Forret, 2010), Scratch kullanımının öğrencilerin işbirliği, mantıksal düşüncelerini, algoritmik düşüncelerini ve problem çözmelerini geliştirdiğini belirtmişlerdir. Böylece düşük seviyede matematik becerisine sahip öğrencilerin matematik becerilerini geliştirdiğini ifade etmişlerdir. Araştırmalarda, Scratch kullanımının öğrencilerin üst bilişsel becerilerini geliştirerek matematik ve fen gibi derslerinde olumlu yönde katkı sağladığı görülmektedir.

Scratch, programlama eğitimine yeni başlayan öğrenciler için yararlı olsa da bazı eksik olduğu yönleri de görülmektedir. Bu eksikler araştırmada iki şekilde ele alınmaktadır.

- 1- Scratch, sadece masaüstü ve dizüstü bilgisayarlarda çalışmaktadır. Bu nedenle, mobil teknolojilerin eğitim ortamlarında popüler olarak kullanıldığı günümüzde Scratch'ın mobil cihazlarda kullanımının olmaması araştırmada bir eksiklik olarak görülmektedir.

- 2- Öğretmenin olmadığı ortamlarda öğrenciler problem çözümünde, özellikle algoritma kurma sürecinde karşılaştıkları hatalar veya yetersizlikleri giderebilmek, doğru çözüm yoluna yönlendirmek için bir desteğe ihtiyaç duymaktadır. Fakat Scratch böyle bir destek öğrencilere sağlamamaktadır.

Problem durumunun analizi gösterdi ki, Scratch programlama dili okullarda programlama eğitiminde yaygın olarak kullanılmaktadır. Bu dil öğrencilere basit düzeyde programlama temellerini öğretme imkanı sağlasa da, öğrencilerin programlama mantığını ve programlama becerilerini daha üst seviyede geliştirmeleri için kısıtlıdır.

Tez çalışması da bu kısıtlamayı ortadan kaldırma yönünde yapılmıştır. Çalışmanın temelini algoritma oluşturmada öğrencilere yön göstermek ve öğrencilerin sınıf dışı ortamda mobil araçlar kullanılmakla da Scratch programlama becerilerini geliştirmek imkanı veren bir mobil uygulama yazılımının geliştirilmesini oluşturmaktadır.

Araştırmanın Amacı

Bu araştırmanın genel amacı, Scratch programlama dili eğitiminin, mobil uygulama ortamını ve gerekli yazılım eklentilerini kullanarak daha etkili bir şekilde yapılmasını sağlamaktır. Bu amaca ulaşabilmek için çalışmada aşağıdaki alt amaçlar belirlenmiştir:

Araştırma alt amaçları.

1. Programlama mantığının öğrencilere daha iyi bir şekilde anlatımına katkı sağlayacak Scratch programlama dili eğitime yönelik bir mobil uygulamanın geliştirilmesi.
2. Mobil uygulamalı Scratch programlama eğitimi alan öğrenciler ile bu eğitimi yalnızca ders ortamında (yüz yüze) alan öğrencilerin programlama becerileri arasında anlamlı bir farklılığın olup olmadığının belirlenmesi.

Araştırmanın Önemi

Son yıllarda küçük yaştaki öğrencilerin programlama eğitimi alması yönünde birçok çalışma bulunmaktadır. Ülkemizde de ortaokul 6., 7. ve 8. sınıfların öğretim programlarına programlama eğitimi entegre edilmiştir. Buna rağmen programlama eğitimi öğrencilerin yeterince dikkatini çekmemektedir. Bu nedenle öğrencilere programlama becerilerinin kazandırılmasında yeterli düzeye ulaşamamıştır.

Öğrencilere programlama becerilerini kazandırmak için yaygın olarak kullanılan yöntemlerden birisi grafik tabanlı programlama dillerinin kullanılmasıdır. Grafik tabanlı programlama dilleri eğlenceli bir ortamda öğrencilere programlamanın basit kavramlarını anlatmayı amaçlamaktadır. Bu diller içerisinde Scratch en yaygın olarak kullanılanlardan biridir. Sürükle bırak yöntemi ile farklı renkte olan kod bloklarını birleştirerek öğrencilere programlamanın mantığını öğretmektedir. Fakat, mobil teknoloji araçlarının ve mobil öğrenmenin yoğun bir şekilde kullanıldığı günümüzde Scratch akıllı telefonlarda kullanılamamaktadır. Ayrıca, Scratch programlama dili eğitiminin mobil cihazlarda verilmesi yönünde mobil öğrenme uygulaması bulunmamaktadır. Bu bağlamda, çalışmada yapılacak olan Scratch programlama dili eğitimine yönelik mobil öğrenme uygulaması ile öğrenciler zaman ve mekandan bağımsız olarak mobil teknolojilerinde avantajlarını kullanarak programlama eğitimi alabileceklerdir. Bu nedenle, mobil öğrenme uygulaması öğrencilerin programlama becerilerinin geliştirmesi açısından önemli olduğu düşünülmektedir.

Araştırmanın Varsayımları

Araştırmada;

1. Deney ve kontrol grubunda olan öğrencilerin derse olan ilgi ve tutumlarının benzer olduğu,
2. Programlama eğitimini sadece yüz yüze olarak alan grubun mobil öğrenme uygulamasına erişiminin olmadığı,

3. Mobil öğrenme uygulaması ile desteklenen grubun uygulamayı yeterli şekilde ve gereken ilgiyi göstererek kullandığı,
4. Deney ve kontrol grubunda olan öğrencilerin yüz yüze olarak aldıkları Scratch programlama dili eğitiminin aynı olduğu varsayılmıştır.

Araştırmanın Sınırlılıkları

1. Araştırmanın deneysel işlem süreci toplam 5 hafta ile,.
2. Araştırma, 2016 – 2017 eğitim öğretim yılında Kırklareli İstiklal Ortaokulunda 6. sınıfta öğrenim gören toplam 94 öğrenci ile,.
3. Araştırmada kullanılan ölçme araçları “Programlama Becerisi Başarı Testi” ile sınırlı tutulmuştur.

Bölüm II: Kavramsal Çerçeve

Bu bölümde çalışma alanı ile ilgili kavramlar, programlama eğitimi ve öğretimi, grafik tabanlı programlama dilleri, Scratch ve mobil öğrenme kavramları açıklanmıştır.

Programlama Eğitimi

Programlama, bilgisayarda gerçekleştirilecek bir problem çözümünü, programlama dillerindeki komutların bir araya getirilerek çalıştırılması sonucunda meydana gelen işlem basamaklarına denir (Kesici ve Kocabaş, 2007). Bir başka tanımda programlama, bir bilgisayarın belirli bir görevi yerine getirmesi, sorunları çözmesi ve insan etkileşimi sağlaması için çeşitli talimat setleri geliştirip uygulayan süreçtir. Bu talimatlar, bir programlama dili kullanılarak geliştirilen bilgisayar programları olarak düşünülür ve bilgisayarın sorunsuz çalışmasına yardımcı olur (Balanskat ve Engelhardt, 2014).

Özünde, programlama kelimesi, bir görevi gerçekleştirmek için bir mekanizmaya yön vermek demektir. İnsanlar, dijital video kaydediciler (DVR), cep telefonları veya kahve makinesi gibi çeşitli mekanizmaları kendi içindeki ara yüzündeki talimatları kullanarak programlamışlardır. Bu aygıtlar gibi bir bilgisayar da programlanabilen bir mekanizmadır. Bir bilgisayara verilen talimatlara, bilgisayar programları veya daha basitçe programlar denir. Program yazanlara programcılar denir. İnsanlar kendi aralarında nasıl İngilizce, İspanyolca, Hintçe ve Çince gibi dillerini kullanarak iletişim kuruyorsa, programcılarda bilgisayarla iletişim kurmak için çeşitli programlama dilleri kullanmaktadır. (Parker, 2016; Zak, 2013).

Programlama dili, programcılarının programları, komut dosyaları veya bilgisayarların yürütmesi için kullanılan talimat setlerini geliştirmek için kullandığı özel bir dildir. Zaman içerisinde yüzlerce programlama dili geliştirilmiştir. Programlama dilleri bilgisayarı

programlamak için ilk olarak makine dilini kullanmıştır. Makine dili, bilgisayarın anladığı tek dildir. Ancak makine dilinin kullanımı karmaşık, maliyetli ve zor olduğundan makine diline temsil edecek programlama dilleri ortaya çıkarılmıştır. Bu programlama dilleri makine dilinin üzerine inşa edilerek geliştirilmiştir (Sebesta ve Mukherjee, 2002). İlk olarak, makine dili talimatlarını temsil etmek için assembly dillerinin kullanılmasıydı. Bunlar daha sonra bir derleyici tarafından makine koduna çevrildi. Bir sonraki adım ise FORTRAN ve COBOL gibi üst düzey programlama dillerinin geliştirilmesiydi. Bunlar, assembly dillerine ve makine koduna göre daha kolaydı. Bu diller, programlamanın kalitesini ve verimliliğini artırmaya yardımcı oldu (Regan, 2008).

Programlama Öğretimi

Programlama becerisine sahip olmak genel olarak önemli bir yeterlilik olarak kabul edilir. Programlamanın eğitimsel ilgisi yalnızca ekonomik öneminden değil aynı zamanda bir öğrenme ortamı olarak değerinden kaynaklanmaktadır. Programlama, bilgisayar bilimlerindeki diğer alanların anlaşılması için önemli bir beceri oluşturmaktadır. Ayrıca, bazı özel programlama sistemleri yapılandırmacı öğrenme ile uyumlu öğrenme ortamları olarak düşünülebilir (Papert, 1980). Programlama, aynı zamanda öğrencilerin mantıksal düşünme, yaratıcı düşünme, problem çözme ve sistematik düşünme gibi üst bilişsel düşünme becerilerinin geliştirilmesinde yardımcı olmaktadır (Clements ve Gullo, 1984; Çetin, 2012; Fesakis ve Serafeim, 2009; Jenkins, 2002).

Programlama öğreten en ünlü ve çoğu zaman alıntılanan yazar Massachusetts Institute of Technology (MIT)'den Seymour Papert'dir. Papert, zamanında üç yaşındaki çocuklar için okullarda yaygın olarak kullanılan ünlü Logo programlama dilinin yazarıdır. Logo dili sadece kapsamlı programlama yetenekleri için kullanılan bir dil değil aynı zamanda üst düzey düşünmeyi geliştiren bir dil olduğu iddia edilir (Tetenbaum ve Mulkeen, 1984). Uygulamanın

satışı için yapılan reklamlar, planlama ve problem çözme gibi bilişsel becerileri geliştirdiğini göstermektedir. Papert (1980), "bilgisayar varlığının zihinsel süreçlere sadece araçsal değil, kavramsal yollarla katkıda bulunabileceğini, insanların bilgisayarla fiziksel temastan uzakta kaldıklarında nasıl düşündüklerini etkileyebileceğini" ileri sürmüştür.

Clements (1985), Logo'nun mantıksal düşünme, yön bulma ve üstbiliş üzerindeki etkilerini belirlemek için Logo kullanan 18 ilkokul öğrencisini incelemiştir. Deney grubundaki öğrenciler, bilgisayar destekli eğitimde aynı beceriler üzerinde çalışan bir kontrol grubuyla karşılaştırıldı. Çalışma, Logo'nun mantıksal düşünme becerileri veya yön bulma üzerinde anlamlı bir etkisinin olmadığı, ancak üst biliş üzerinde belirgin bir etkisinin olduğunu göstermiştir. Bu sonuç, programlamanın öğrencilerin sahip oldukları bilişsel becerileri daha etkili bir şekilde kullanmalarına yardımcı olabileceğini belirtmiştir.

Programlama önemli bir beceri olmasına rağmen programlamayı öğrenmek ve bu alanda ustalaşmak zor bir süreçtir (Eckerdal, 2009; Genç ve Karakuş, 2012; Gomes ve Mendes, 2007; Hadjerrouit, 2008; Jenkins, 2002). Bunun nedeni programlamaya yeni başlayan öğrencilerin karşılaştıkları karmaşık mantıksal yapılar içerisindeki algoritma hatalarıyla uğraşmak zorunda kalmalarıdır. Programlamaya giriş derslerinde başarısızlık veya programlamayı bırakma oranlarının % 15 ila % 30 olduğu tahmin edilmektedir (Fesakis ve Serafeim, 2009). Çünkü programlama yapmak için geliştirilen birçok ortam genellikle metin tabanlı sistemlerdir. Öğrenciler bu çeşit sistemlerde program geliştirmek adına binlerce kod öğrenmek zorunda kalabilmektedirler. Dolayısıyla metin tabanlı sistemlerde yazılan kodları okumak, yazmak, anlamak ve hatırlamak zor olabilmektedir (Ortiz, 1997). Programlamaya yeni başlayan öğrencilerin yaşadıkları zorlukları azaltmak ve öğrenmelerini kolaylaştırmak için çeşitli yaklaşımlar ortaya konmaktadır. Bu yaklaşımlar arasında grafik tabanlı programlama dili uygulamalarının kullanımı olduğu belirtilmektedir (Guzdial, 2004).

Programlama Eğitiminde Kullanılan Grafik Tabanlı Programlama Dilleri

Bilgisayar bilimi eğitimine artan ilgi, programlama öğrenen çocukların programlama dili ve editörü seçeneklerinde bir artışa neden olmuştur. Bu programlama dillerinin çoğu grafik tabanlı olup, öğrencilerin sözdizimini görmezden gelmelerine ve programlar oluşturmaya odaklanmasına olanak tanır. Grafik tabanlı programlama dilleri, kullanıcıların komut dosyaları oluşturmak için puzzle parçaları gibi sürükleyecekleri Lego benzeri bloklarda bir dizi programlama komutları sağlar. Bu komut dosyaları genellikle bir oyun oluşturmak, bir hikaye anlatmak veya daha fazlasını yapmak için kullanılabilecek karakterleri veya diğer görüntüleri kontrol eder. Grafik tabanlı programlama dilleri, muhtemel sözdizimi hatalarını azaltarak ve çoğu metinsel programlama dilinden daha küçük bir komut seti sağlayarak programlamaya yönelik bilişsel engelleri azaltmaktadır (Hill, 2015). Grafik tabanlı programlama dilleri içerisinde en bilinenler arasında App Inventor, Alice ve Scratch olduğu söylenebilir (Demirer ve Sak, 2016).

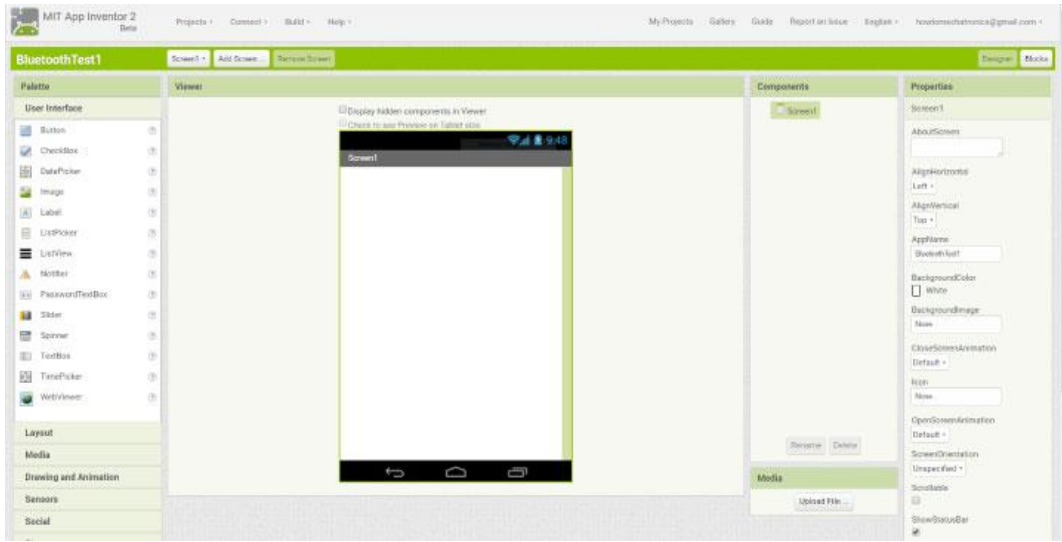
App inventor.

App Inventor programlama ortamında, Android işletim sistemli telefonlar için uygulamalar oluşturmak için sürükle bırak arayüzü kullanılır. Kullanıcılar, grafik arayüzünü tasarlar, kodlarını temsil etmek için birbirine uyan bulmaca parçalarını sürükleyip bırakır ve kodlarını bir emülatör veya bir Android cihazda test eder. Bu ortam, değişkenler, koşullar, döngüler, prosedürler, mantık, olaylar ve listeler kavramlarını öğretmeye yardımcı olur. App Inventor öğeleri, Java SDK, yazılım geliştirme kiti kullanılarak programlanacak olan uygulamalara dahil edilebilir. Bu, kullanıcıların Android için gerekli incelikleri öğrenmeden Java kodlamayı yapmalarını sağlar.

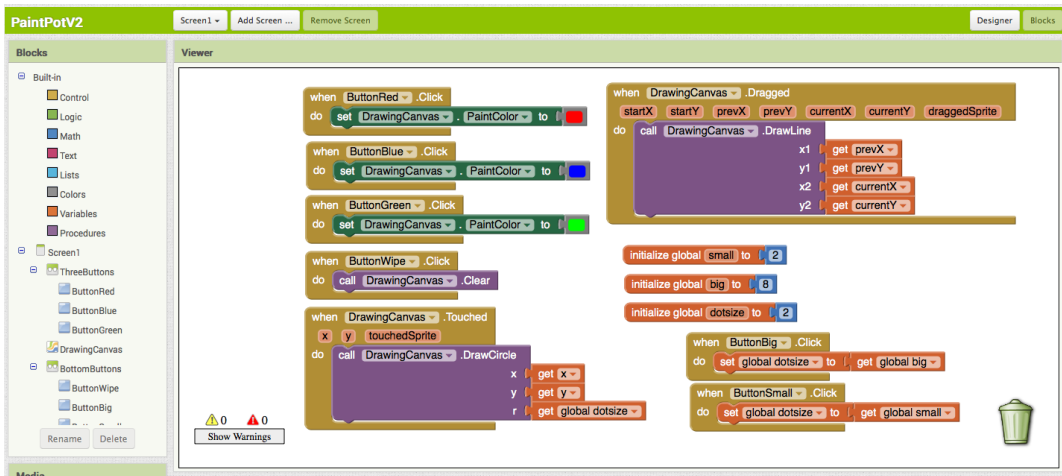
App Inventor uygulaması ücretsizdir ve Windows işletim sisteminde, Apple Macintosh işletim sisteminde veya Linux işletim sisteminde kullanılabilir. Kullanıcılar ortama erişim yapabilmek için bir Gmail hesabına sahip olmalıdırlar. Tüm App Inventor dosyaları

kullanıcının Gmail hesabı altında çevrimiçi olarak saklanır. Uygulamaya ve dosyalara erişmek için internet bağlantısı olması gerekmektedir.

App Inventor web sitesi, değişen zorluk seviyelerine göre geniş bir eğitim yelpazesi sunar. Eğitimciler için mevcut pek çok kaynak vardır. Forumlar, bloglar, ders programları, örnek projeler ve ders kitaplarını içerisinde bulabilirler. App Inventor gerçekte Google tarafından oluşturulmuştur. Uygulama 2010 yılında halka açıldı ve bir yıl sonra feshedildi. MIT, bu projeyi devraldı ve Mart 2012'de tekrar yayınladı. Şekil 1'de App Inventor ortamının tasarım ekranı ve Şekil 2 'de kodlama ekranı göstermektedir.



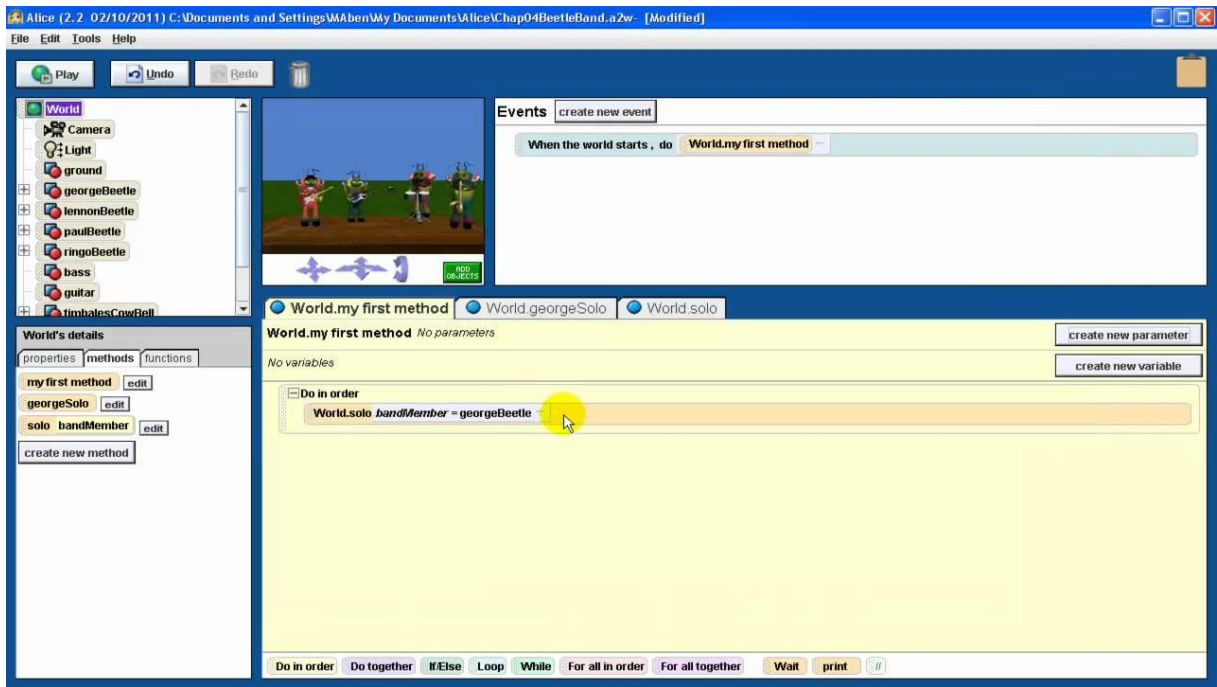
Şekil 1. App Inventor Ortamının Tasarım Ekranı



Şekil 2. App Inventor Ortamının Kodlama Ekranı

Alice.

Programlama eğitimi için kullanılan uygulamalardan biri Carnegie Mellon Üniversitesi'nin geliştirdiği Alice'dir. Alice, kullanıcıya çeşitli komutlar çalıştırmak için sürükle bırak arayüzü sağlayarak programlama yapmayı kolaylaştırmaya çalışır (Şekil 3). sağlayarak programlamayı kolaylaştırmaya çalışır. Alice ile öğrenciler kendi hikayelerini ve sahnelerini oluşturabilir, basit oyunlar ve filmler yaratabilirler. Alice, öğrencilerin sözdizimsel olarak doğru, hatasız kod üretmesine izin vermek için ideal bir ortam sunar (Stenerson, 2012).



Şekil 3. Alice Grafik Tabanlı Programlama Ortamı

Scratch

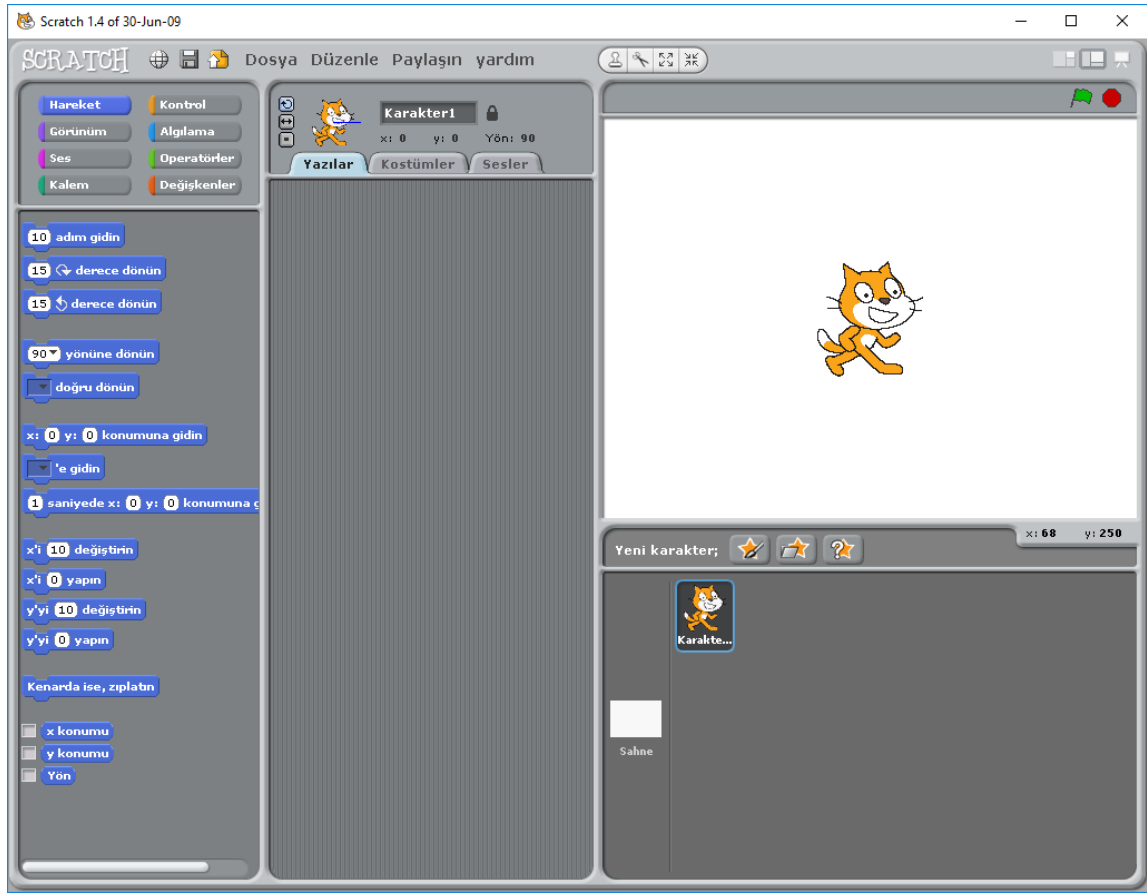
Scratch'ın genel yapısı ve kullanımı.

Scratch, çocuklara ve programlamaya ilk kez başlayan kişilere programlama öğretmek amacıyla MIT Media Lab tarafından geliştirilen bir programlama dilidir. Scratch, Mayıs 2007'de piyasaya sürülmüştür. Scratch, bilgisayar oyunları, etkileşimli öyküler, bilgisayar animasyonu ve diğer her tür multimedya projesinin geliştirilmesini destekler.

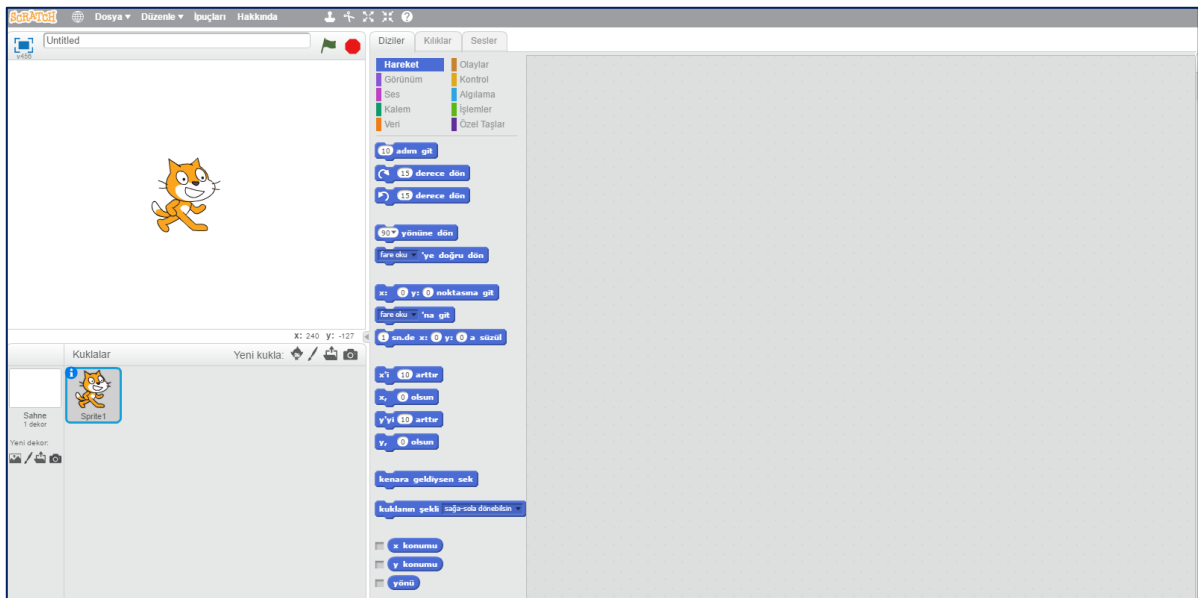
Scratch, yeni programcılarının Lego veya puzzle gibi kod bloklarını kullanarak programlar oluşturmalarına olanak tanır. Scratch, 8 farklı renkte kod bloklarından oluşan bir programlama dilinden ve grafik oluşturmak için bir boya uygulaması ve dahili ses düzenleme özellikleri içeren öğrenmesi kolay bir grafik geliştirme ortamından oluşur. Scratch, aynı zamanda kendi Scratch projelerinizi oluşturmak için kullanabileceğiniz örnek uygulama koleksiyonları, grafik ve ses dosyaları ile birlikte gelir.

Scratch programları, birbirine kenetlenmiş grafiksel kod bloklarından oluşur. Kod blokları puzzle parçalarını birbirine yapışacak şekilde kullanılmasına benzemektedir. Kod blokları, yeni programcılarının geçersiz kombinasyonlarda kullanmalarını engelleyerek, sadece mantıklı yollarla birbirine yapıştırabilirler. Böylece Scratch, programcılarının uygun programlama algoritmasını oluşturmaya zorlar ve programcılarının programlama mantığını geliştirmesine ve algoritmayı doğru bir şekilde öğrenmesine olanak sağlar.

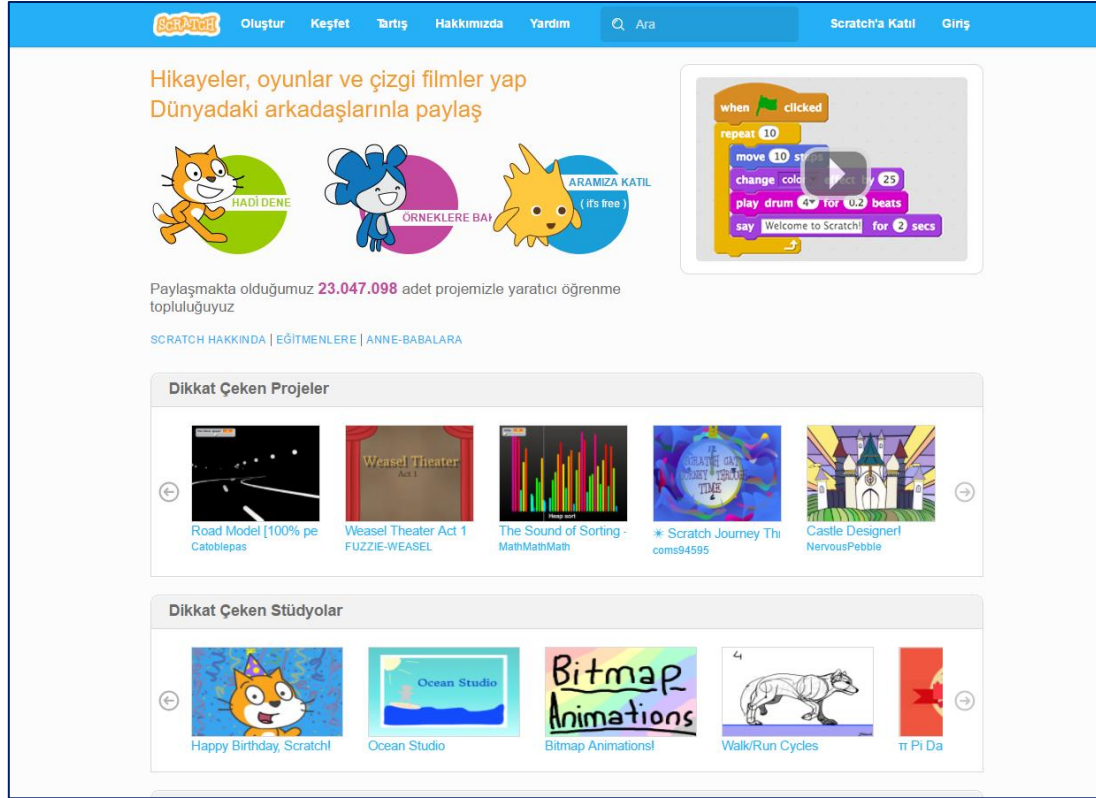
Scratch platformunun iki bileşeni vardır. Birincisi, kullanıcıların çevrimdışı projeler oluşturabilecekleri indirilebilir bir programlama uygulamasıdır (Şekil 4). İkincisi, Scratch çevrimiçi web ortamıdır (scratch.mit.edu). Bu ortamda, kullanıcılar çevrimiçi olarak programlama yaparak projelerini oluşturabilmektedir (Şekil 5). Aynı zamanda kullanıcılar tarafından oluşturulan projelerle oluşan bir paylaşım ortamıdır (Şekil 6). Scratch kullanıcılara 40 dil seçeneği sunmaktadır. Eğitimciler için etkileşimli forumların yanı sıra oyun geliştiricileri için dinamik yorum merkezleri sağlar.



Şekil 4. Scratch Çevrimdışı Programlama Uygulaması



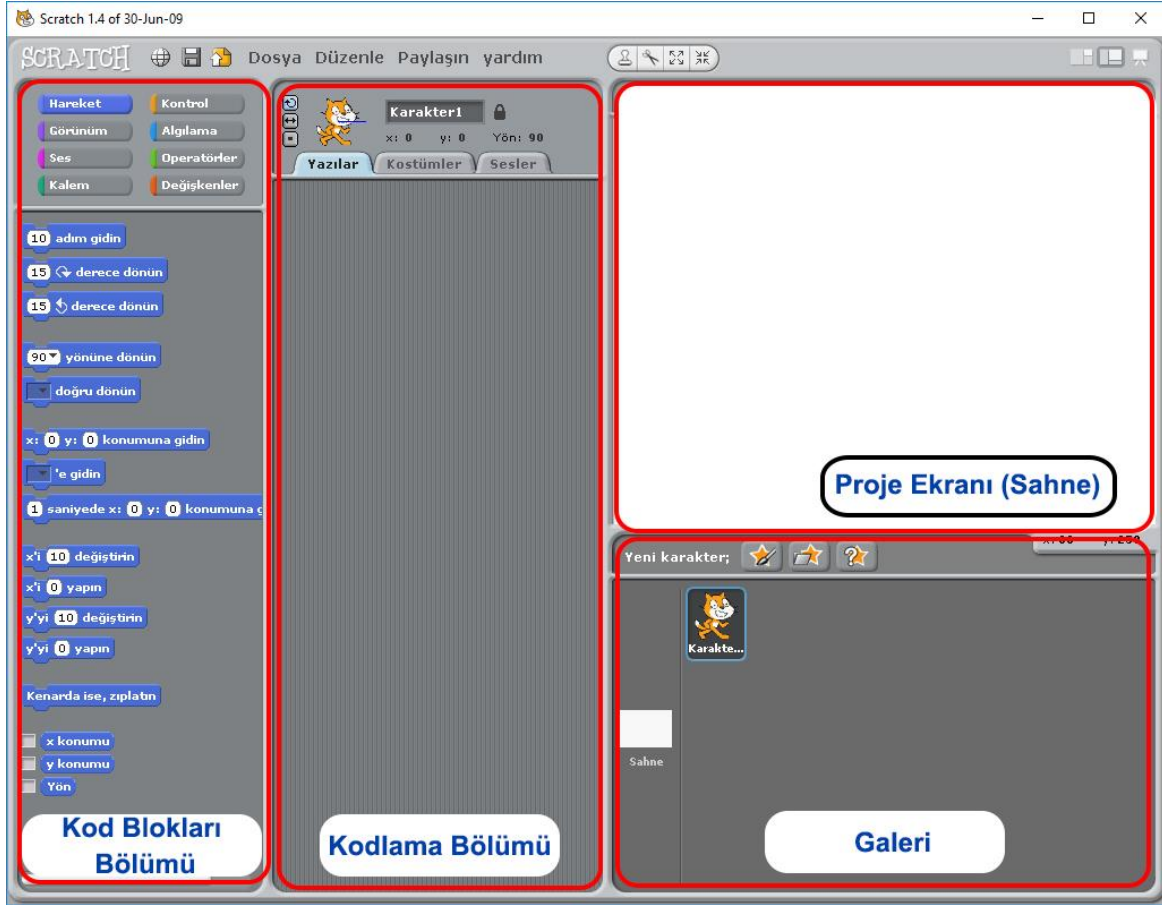
Şekil 5. Scratch Çevrimiçi Programlama Ortamı



Şekil 6. Scratch Çevrimiçi Paylaşım Ortamı

MIT Medya Laboratuvarında Lifelong Kindergarten Project Scratch platform tasarımcıları, kişiselleştirmeye büyük önem vermişlerdir. Kullanıcılara Scratch projelerinde kendi ses kayıtlarını, müziklerini, fotoğraflarını kullanabilecekleri bir ortam sunmaktadır. Aynı zamanda Scratch projesinin içerisinde kişisel grafiklerini çizebilmeye imkan sağlamaktadır (Resnick, 2012).

Scratch, Windows, Mac OS ve Linux işletim sistemlerin altında çalışmaktadır. Uygulama scratch.mit.edu adresinden indirilebilmektedir. Çevrimdışı proje uygulaması, dört ana çalışma alanından oluşmaktadır (Şekil 7). Bunlar; kod blokları bölümü, kodlama bölümü, proje ekranı (sahne) ve galeri'dir.



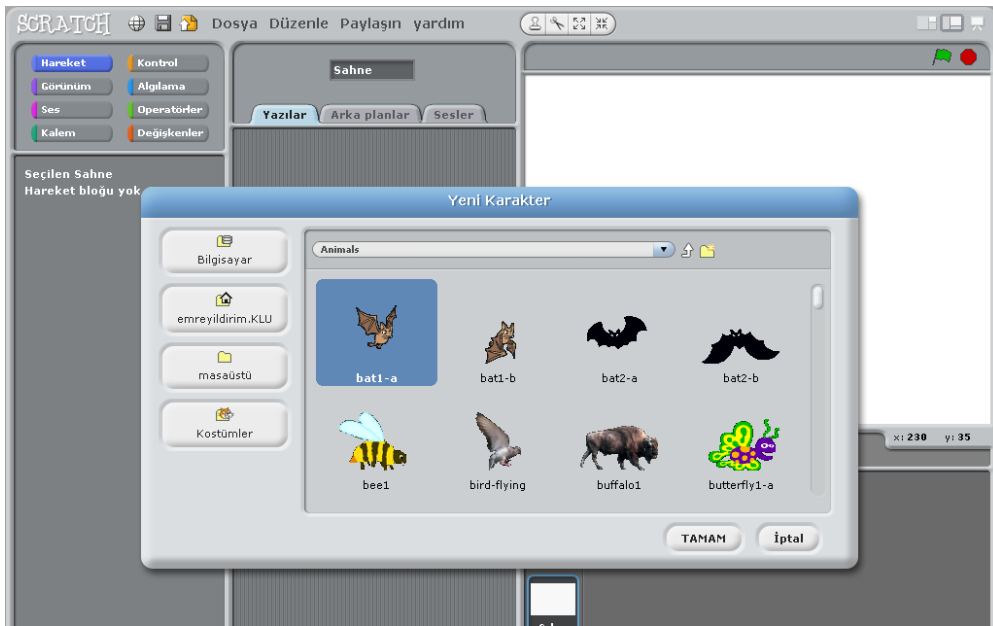
Şekil 7. Scratch Çevrimdışı Programlama Uygulamasının Bölümleri

Kod blokları bölümünde, sekiz kod bloğu kategorisi vardır. Bunlar; Hareket, Görünüm, Ses, Kalem, Kontrol, Algılama, Operatörler ve Değişkenler'dir. Hareket kod blokları, sahnedeki karakterlerin x ve y eksenlerinde hareket etmesini sağlar. Görünüm kod blokları, sahnedeki karakterin rengini, boyutunu ve genel görünümünü kontrol eder. Kod bloklarını düzenlemek için, seçilen kod bloğunu kodlama bölümüne sürükleyip onları Lego gibi birleştirebilirsiniz (Şekil 8).



Şekil 8. Scratch Kod Bloklarının Sürüklenmesi

Galeri bölümünde, çizilen veya kullanıcı tarafından yüklenen resimler ve karakterler seçilebilir. Bu bölüm, kullanıcının karakterleri denetlediği yerdir. Bununla birlikte, eğer kullanıcı istediği karakteri galeri içerisinde bulamazsa kendisi “yeni karakter ekle” butonunu kullanarak istediği karakteri ekleyebilir (Şekil 10.).



Şekil 9. Yeni Karakterin Eklenmesi

Scratch ile ilgili yapılan çalışmalar.

Bu bölümde, Scratch programlama dilinin; farklı disiplinlerde kullanıma, programlamaya dair öğrenci görüşlerine ve öğrencilerin problem çözmelerine, yaratıcılıklarına, algoritma ve programlama becerilerine ve motivasyonlarına etkisini araştıran çalışmalar bulunmaktadır. Bu bağlamda yapılan çalışmalar:

Gülbahar ve Kalelioğlu (2014) çalışmalarında, öğrencilerin problem çözme becerileri üzerinde Scratch programlama uygulamasının etkisini incelemişlerdir. Araştırmaya ilköğretim 5.sınıfta eğitim gören 49 öğrenci katılmıştır. Araştırma sonucunda, öğrencilerin problem çözme becerilerinde anlamlı bir fark olmamıştır. Ancak, öğrencilerin programlama hakkındaki düşünceleri dikkate alındığında, tüm öğrenciler programlamayı sevdiklerini ve kendi yazılımlarını geliştirmek istediklerini ifade etmişlerdir. Bununla birlikte Scratch uygulamasının kullanımını kolay bulmuşlardır.

Ouahbi ve diğerleri (2015) araştırmalarında, Scratch yazılımını kullanarak öğrencilere temel programlamayı öğretmeyi amaçlamışlardır. Ayrıca, öğrencilerin programlamaya karşı motivasyonlarını incelemişlerdir. Araştırma, Fas'ın kuzey doğusunda yer alan Nador kentinde yaşayan 69 lise öğrencisi ile gerçekleşmiştir. Öğrenciler 3 grubu ayrılmıştır. Birinci grup, öğretmenleri ile birlikte Scratch programında kullanılan oyunlaştırma yöntemiyle programlama öğrenmişlerdir. İkinci grup, öğretmenlerinden yüz yüze eğitim alarak programlama öğrenmişlerdir. Üçüncü grup ise, yine yüz yüze eğitim almışlardır. Fakat, derslere farklı öğretmenler girmiştir. Deneysel işlem başlamadan önce öğrencilerin programlama bilgi düzeylerini belirlemek için anket uygulanmıştır. Anket sonucunda, öğrencilerin %95'inin programlama bilgisine sahip olmadığı ve %5'nin ise az miktarda olduğu görülmüştür. Deneysel işlemler gerçekleştikten sonra toplanan verilere göre, Scratch programı kullanılarak yapılan eğitimde, öğrencilerin başarıları ve motivasyonları olumlu

yönde farklılık göstermiştir. Diğer iki grupta ise, programlamanın monoton ve sıkıcı olduğu izlenimi ortaya çıkmıştır.

Shin ve Park (2014) araştırmalarında ilköğretim düzeyindeki öğrencilere Scratch programlama eğitimi verilmiş ve öğrencilerin problem çözme becerilerindeki etkisini incelemişlerdir. Araştırmaya toplam 46 öğrenci katılmıştır. Öğrencilere problem çözme becerilerini ölçen 34 soruluk bir anket uygulamışlardır. Araştırma sonunda Scratch programlama eğitiminin öğrencilerin problem çözme becerilerine olumlu katkı sağladığı sonucuna varmışlardır.

Nam, Kim ve Lee (2010) yapmış oldukları çalışmada Scratch programlama dili ile öğrencilere sıfırdan programlama eğitimi vererek, öğrencilerin programlama başarılarına ve problem çözmelerine etkisini incelemişlerdir. Yapılan çalışmaya 6. sınıfta eğitim gören 60 öğrenci katılmıştır. Öğrenciler deney ve kontrol grubu olarak ayrılmışlardır. Deney grubundaki öğrenciler Scratch programlama dili ile programlama eğitimi almışlardır. Kontrol grubundaki öğrenciler ise klasik yöntemlerle programlama eğitimi almışlardır. Çalışma toplam 4 hafta boyunca devam etmiştir. Çalışmanın sonunda araştırmacılar deney grubundaki öğrencilerin başarılarında ve problem çözme becerilerinde olumlu yönde bir değişim olduğunu belirtmişlerdir.

Sanjanaashree, Anand Kumar ve Soman (2014), Scratch programlama dilinin yabancı dil eğitiminde kullanılabilirliğine dair bir araştırma yapmışlardır. Araştırmacılar, Scratch programlama dili ile yaptıkları programda İngilizce ve Tamil dilini öğretmeye çalışmışlardır. Programın içeriğinde ders ortamı ve test olmak üzere iki bölüm bulunmaktadır. Araştırmanın sonunda araştırmacılar Scratch programlama dili yapılan uygulamalarla İngilizce ve Tamil gibi yabancı dil eğitiminin verilmesine yardımcı olunacağı bulgusuna ulaşılmıştır.

Malan ve Leitner (2007) Bilgisayar bilimleri dersinde Scratch programlama dili kullanılarak üniversite öğrencilerine programlama eğitimi vermişlerdir. Bu çalışmada

öğrencilerin Scratch ile verilen programlama eğitiminde görüşlerini incelenmiştir. İki hafta süren çalışma sonucunda öğrencilere görüşlerini belirlemek için anket uygulanmıştır. Toplanan verilerin analizi sonucunda öğrenciler Scratch eğitiminin Java programlamaya temel olması için kullanılabilir bir ortam olduğunu belirtmişlerdir.

Wang, Huang ve Hwang (2014) yaptıkları araştırmada problem çözme senaryolarının olduğu Scratch ile entegre proje tabanlı bir çalışma yaparak öğrencilerin problem çözme becerisini, öğrenmeye karşı tutumlarını ve motivasyonlarını incelemişlerdir. Araştırmaya 43 normal düzeyde ve 48 matematikte başarılı öğrenci olmak üzere toplam 91 ortaokul öğrencisi katılmıştır. Deneysel sonuçlar, önerilen öğrenme yaklaşımının hem normal düzey ki hem de matematikte iyi olan öğrencilerin başarılarında önemli ilerlemelere sebep olmuştur. Bununla birlikte matematikte iyi olan öğrencilerin problem çözmelerinde, öğrenme tutumlarında ve motivasyonlarında ortalama öğrencilere göre daha iyi sonuçlar elde edilmiştir.

Ruf, Mühling ve Hubwieser (2014) çalışmalarında Scratch ve Karel programlama ortamlarını karşılaştırmışlardır. Çalışmanın grubu, ortaokul seviyesinde farklı iki sınıfta öğrenim gören öğrenciler seçilerek oluşturulmuştur. İki ortamdan her biri bir sınıfta denenerek öğrencilere programlama becerileri kazandırılmaya çalışılmıştır. Her iki sınıfa da deney öncesi ve sonrası programlama başarı testi uygulanmıştır. Deney sonucunda, Scratch kullanan öğrencilerin programlama becerileri ve motivasyonlarının, Karel kullanan öğrencilere göre daha yüksek çıktığı belirtilmiştir.

Osman ve diğerleri (2012) programlama eğitimi veren 3 farklı öğrenme ortamını öğrencilerin programlama becerilerine ve motivasyonlarına bakarak karşılaştırmışlardır. Kullanılan 3 öğrenme ortamı: Visual Basic Express, 2D Programmable PyGame ve Scratch uygulamalarıdır. Çalışmaya 591 ortaokul öğrencisi katılarak bu öğrenme ortamlarını kullanmaları için 3 gruba ayrılmıştır. Öğrencilere deney sonrasında memnuniyetlerini ve programlama başarılarını ölçen testler uygulanmıştır. Çalışmanın veri analizleri sonucunda

Scratch ve 2D Programmable PyGame kullanan öğrencilerin Visual Basic Express kullanan öğrencilere göre daha yüksek motivasyon ve programlama becerisi göstermişlerdir. Sonuç olarak Scratch ve 2D Programmable PyGame öğrenme ortamları katılımcılar için uygun ortam olduğunu araştırmacılara göstermektedir.

Yapılan çalışmalar incelendiğinde; Scratch kullanımının öğrencilerin tutumlarına, motivasyonlarına, üst düzey bilişsel becerilerine ve programlama becerilerine genel olarak olumlu yönde katkı yaptığı görülmektedir. Aynı zamanda Scratch'ın, öğrencilerin programlama eğitimleri için uygun bir ortam olduğu görüşü de ortaya çıkmaktadır.

Mobil Öğrenme

Mobil öğrenme ortamının özellikleri.

Mobil öğrenme ile ilgili alanyazında birçok tanım bulunmaktadır. Bu tanımlara bakıldığında, Traxler (2005) mobil öğrenmeyi, tek veya baskın cihaz olan el ve avuçiçi bilgisayar teknolojileri ile herhangi bir eğitimin sağlanması olarak tanımlamaktadır. Bir başka tanımda ise, öğrencilerin kablosuz cihazlar kullanarak zaman ve mekan gözetmeksizin öğrenme içeriklerine erişebilmelerine ve etkileşimli öğrenme ortamları oluşturabilmelerine olanak sağlayan öğrenme sistemi olduğu belirtilmektedir (Dönmez, Gelibolu ve İnceoğlu, 2006).

Mobil cihazların ve teknolojilerin kullanımında son yıllarda hızlı bir büyüme görülmüştür. Uluslararası Haberleşme Birliği (ITU), ülkelerin 2014 yılına kadar cep telefonu kullanım sayılarını ve oranlarını belirten bir istatistik raporu yayınlamıştır. İstatistik raporuna göre, Türkiye'de yaşayan insanların %94'ünün cep telefonu kullanmakta olduğu belirtilmiştir (ITU, 2016).

Akıllı telefon ve tablet gibi güncel mobil teknolojiler, kablosuz bağlantı özellikleri sayesinde bilgiye zamandan ve mekandan bağımsız olarak erişimini sağlamaya yardımcı

olmaktadır (Chen, Chang ve Wang, 2008; Uden, 2006; Vinu, Sherimon ve Krishnan, 2011). Günümüzde mobil cihazlar her alanda kullanılmaktadır. Mobil cihazların yaygınlaşmasıyla birlikte, araştırmacılara öğrenme ve öğretme ortamlarına bu teknolojiyi dahil etmelerini sağlamış ve mobil öğrenme alanında araştırma olanakları kazandırmıştır (Ako-Nai, Tan ve Pivot, 2012; Goundar, 2011; Park, 2011).

Birçok araştırmacı, bir mobil öğrenme ortamının uygulanması için fırsatlar ve kısıtlamalar hakkında araştırmalar yapmıştır. Bazıları cihaz üzerinde yoğunlaşmış ve bazıları ise öğrenme ortamına odaklanmışlardır. Mobil öğrenme, bir öğrenme destek aracı olarak çok sayıda eğitim görevi gerçekleştirebileceği söylenmektedir (Keegan, 2005).

Mobil öğrenme ile ilgili yapılan çalışmalar.

Bu bölümde, mobil öğrenmenin farklı disiplinlerde kullanımının öğrenci başarısını ve görüşlerini araştıran çalışmalara değinilmiştir;

Chen ve Chung (2008) yaptıkları çalışmada, İngilizce kelime eğitimine yönelik bir mobil öğrenme sistemi geliştirmişlerdir. Mobil öğrenme sistemi, madde tepki kuramı ve öğrenme bellek döngüsüne dayalı kelime yeteneği ve hafıza döngüsüne göre tasarlanmış kişiselleştirilmiş bir yazılımdır. Yazılım, mobil cihaz olarak PDA'ların kullanımına göre geliştirilmiştir. Çalışmanın katılımcılarını, 21 ile 23 yaş arasında değişen iki erkek ve 13 kız olmak üzere 15 öğrenci oluşturmuştur. Deneysel sonuçlar, önerilen sistemin İngilizce kelime öğretimine yönelik öğrencilerin performansını olumlu etkilediğini göstermektedir.

Sun ve diğerleri (2015) yapmış oldukları çalışmalarında, yabancılara Çince öğretimine destek amaçlı içerik tabanlı mobil öğrenme sistemi geliştirmişlerdir. Mobil öğrenme sistemi ontoloji semantik web kural dili kullanılarak geliştirilmiştir. Sistem öğrenciye uygun içerikleri

getirerek öğrenme sürecinde destek sağlıyor. Araştırmanın sonucunda, mobil öğrenme sisteminin büyük ölçüde öğrenme etkisini ve verimliliğini arttırdığı gözlemlenmiştir.

Köse, Koç ve Yücesoy (2013) öğrencilere mobil artırılmış gerçeklik aracı sunarak onların soyut ya da teknik derslerdeki eğitim süreçlerine destek vermeyi amaçlamışlardır. Öğrenciler mobil yazılımın içerisindeki 3D animasyonlar ve özel yapılmış videoları izleyerek konu hakkında daha fazla fikir sahibi olabilmekte ve bilgilerini geliştirmektedirler. Mobil yazılım, işlemleri yapabilmesi için öğrencilere verilen ders kitapları veya öğretim elemanlarının verdiği fiziksel evrakların üzerindeki özel işaretleri kamera yardımıyla okuyarak öğrencilere içerikleri sunmaktadır. Araştırmanın deneysel değerlendirilmesinde bilgisayar mühendisliği, bilgisayar eğitimi ve farklı üniversitelerin bilgisayar teknolojileri bölümünden toplam 200 öğrenci yer almıştır. Sonuç olarak yazılım, mobil cihazların avantajlarını kullanarak öğrencilere etkili bir öğrenme deneyim sunmaktadır.

Fernández-López ve diğerleri (2013) çalışmalarında, özel eğitim öğrencilerinin temel becerilerini (dil, matematik, çevre bilinci, özerklik ve sosyal) geliştirmek amaçlı IOS tabanlı Picaa isimli bir mobil öğrenme yazılımı geliştirmişlerdir. Araştırmanın çalışma grubunu, İspanya'dan özel ve ilköğretim okullarında özel eğitim ihtiyaçları olan 39 öğrenci oluşturmuştur. Uygulama süreci tamamlandıktan sonra öğretmenler, öğrencilere 5 temel beceriyi ölçen anket uygulamışlardır. Uygulanan 51 maddelik anket ve uygulama sonucunda, Picaa mobil yazılımının, özel eğitim ihtiyacı olan öğrencilerin temel becerilerini geliştirmesinde olumlu etkileri olduğu tespit edilmiştir.

Skiada ve diğerleri (2014) araştırmalarında, Disleksi öğrenme güçlüğü bulunan çocuklar ve yetişkinler için mobil uygulama geliştirmişlerdir. Araştırmada mobil öğrenme sistemini kullanarak katılımcıların okuduğunu anlama, kısa süreli bellek ve matematiksel

problem çözme gibi temel becerilerini geliştirmelerine yardımcı olmak amaçlanmaktadır. Araştırma Yunanistan'daki konuşma terapisi merkezinde gerçekleştirilmiştir. Sonuçlara göre zamanla kullanıcıların performansında ilerlemeler gösterdiği ifade edilmiştir ve mobil öğrenmenin temel becerilere katkıda bulunduğu gözlemlenmiştir.

Keskin (2011) çalışmasında akademisyenlerin mesleki gelişimini sağlamalarına yönelik mobil öğrenme sistemi geliştirmiştir ve akademisyenlerin algılarını ve deneyimlerini incelemiştir. Uygulama IOS işletim sistemi tabanlıdır. Çalışmada eylem araştırması ve tasarım tabanlı araştırma kullanılmıştır. Çalışmaya odak grup görüşmesinde 12, ankete 478 ve eylem araştırmasına katılan 15 akademisyen katılmıştır. Toplanan verilerin çözümlemesinde betimsel çözümlemeler kullanılmıştır. Çalışma sonuçlarına göre, akademisyenlerin mobil teknolojileri kullanabileceği gözlemlenmiş ve akademisyenler mobil öğrenmenin uygulamasının mesleki açıdan olumlu katkıda bulunduğunu belirtmişlerdir.

Burmabıyık (2014) tarafından yapılan çalışmada, ADDIE öğretim tasarım modeline göre katı cisimler ve geometri konularına yönelik mobil öğrenme materyali geliştirilmiştir. Araştırmanın deseni eylem araştırmasından biçimlendirici araştırma yöntemidir. Araştırmanın katılımcılarını 9.sınıfta okuyan toplam 8 öğrenci ve 5 öğretmen oluşturmaktadır. Verilerin toplanmasında, öğrencilerin ve öğretmenlerin mobil öğrenmenin kullanılabilirliğine yönelik standartlaştırılmış açık uçlu görüşme soruları kullanılarak görüşleri alınmıştır. Araştırmanın sonucunda, mobil öğrenmenin hem öğretmen hem de öğrenciler tarafından beğenildiği ve teknoloji kullanımına karşı istekli oldukları ifade edilmiştir.

Turgut (2015) yapmış olduğu çalışmada öğretim tasarımı modellerinden ADDIE modeli kullanılarak IOS tabanlı dijital hikaye oluşturma mobil yazılımını geliştirmiştir. Mobil yazılım okul öncesi öğrencilerinin fen ve doğa etkinliklerine yönelik geliştirilmiştir.

Öğrenciler mobil yazılımda resim ekleme, ses kaydetme ve çizimler yaparak kendi hikayelerini oluşturabilmektedirler. Bu uygulama 5 öğrenci tarafından 4 hafta boyunca kullanılmıştır. Uygulamanın kullanımı bittikten sonra öğrenciler ve aileleriyle yarı yapılandırılmış görüşmeler ve mülakatlar yapılarak yazılımın değerlendirilmiştir. Araştırma sonunda toplanan verilere göre geliştirilen yazılımın okul öncesi dönemindeki öğrencilerin sosyal, psikolojik ve bilişsel yönden özelliklerine uygun olduğu ifade edilmiştir.

Dehmenoglu (2015) Programlama Temelleri dersine yönelik Android tabanlı “Kod Her Yerde” isimli mobil öğrenme uygulaması geliştirmiştir. Geliştirdiği uygulamayı kullanan öğrenciler ile kullanmayan öğrencilerin akademik başarıları arasındaki farkları incelemiştir. Araştırmaya meslek lisesi 10. sınıf kademesinde öğrenim gören 31’i kontrol ve 31’i deney grubu olmak üzere 62 öğrenci katılmıştır. Kontrol grubu programlama dersini okullarında yüz yüze olarak almıştır. Deney grubu ise programlama dersini yüz yüze eğitime destek olarak geliştirilen mobil öğrenme yazılımını kullanmışlardır. Araştırmanın sonucunda, mobil öğrenmeyi kullanan öğrencilerin akademik başarılarında anlamlı bir etkisinin olduğu ve sadece yüz yüze öğrenim gören öğrencilerden daha başarılı oldukları ifade edilmiştir.

Kılıç (2015) çalışmasında, Kimya dersine ait Atom ve Periyodik Sistem ünitesine yönelik Android tabanlı mobil öğrenme uygulaması geliştirmiştir ve öğrencilerin akademik başarılarına, kalıcı öğrenmelerine ve motivasyonlarına etkisini incelemiştir. Uygulamaya, Anadolu lisesi 9. sınıf kademesindeki 60 öğrenci katılmıştır. Çalışmada araştırma modeli olarak deneme modellerinden yarı deneysel desen kullanılmıştır. Araştırma sonunda, deney grubundaki öğrencilerin akademik başarıları, tutumları ve kalıcık puanlarının, kontrol grubundaki öğrencilere göre daha yüksek olduğu tespit edilmiştir.

Yokuş (2016), Öğretim İlke ve Yöntemleri dersine yönelik öğrencilerin bilgi ve becerilerinin geliştirmesini sağlayacak mobil bir uygulama geliştirmiştir. Geliştirilen uygulamayı öğrenciler 2 ay boyunca kullanmışlardır. Araştırmada öğrencilerin akademik başarıları, tutumları ve mobil öğrenme uygulamasına yönelik görüşleri incelenmiştir. Araştırma tasarım tabanlı olarak desenlenmiştir. Araştırmaya lisans düzeyinde 209 öğrenci katılmıştır. Araştırma sonucunda, öğrencilerin akademik başarıları ve tutumları olumlu yönde farklılık göstermiştir. Öğrenciler, mobil öğrenme uygulamasına karşı olumlu görüş belirtmişlerdir.

Yukarıda bahsedilen çalışmalar incelendiğinde, mobil öğrenmenin öğrencilerin başarılarına olumlu yönde katkıda bulunduğu görülmektedir. Bununla birlikte mobil öğrenme kullanan öğrencilerin genel olarak mobil öğrenmeye karşı olumlu görüşler belirttikleri görülmektedir. Sonuç olarak, yapılan bu tez çalışmasında mobil öğrenmenin öğrencilerin akademik başarıları üzerinde faydalı olabileceği düşünülmektedir.

Bölüm III: Yöntem

Bu bölümde araştırmanın modeli, çalışma grubu, veri toplama araçları, Scratch programlama dili mobil öğrenme uygulaması ve verilerin analizi açıklanmıştır.

Araştırma Modeli

Bu araştırmada Scratch programlama dili eğitime yönelik Android tabanlı bir mobil uygulama geliştirilmiştir. Mobil uygulama geliştirilirken ADDIE öğretim tasarımı modelinden yararlanılmıştır. Geliştirilen mobil uygulamanın uygulanıp değerlendirilmesinde öntest-sontest kontrol gruplu yarı deneysel desen kullanılmıştır. Bu desende öğrenciler, deney ve kontrol grubu olarak iki gruba ayrılmaktadır Her iki gruptaki öğrenciler, çalışmanın öncesinde ve sonrasında olmak üzere bağımlı değişkenle ilgili ölçüme tabi tutulurlar. Bu araştırma modeli, okul ortamında var olan düzeni bozmadan uygulama yapma olanağı sağlamaktadır (Büyüköztürk, 2004).

Çalışma Grubu

Kırklareli İstiklal Ortaokulu 6. sınıf kademesinde öğrenim gören 48'i erkek ve 46'sı kız olmak üzere 94 öğrenci, araştırmanın çalışma grubunu oluşturmaktadır. Çalışmanın deney grubunu 41, kontrol grubunu ise 53 öğrenci oluşturmaktadır. Çalışma grubu, amaçlı örnekleme yöntemine göre seçilmiştir. Amaçlı örneklemede, araştırmacı tarafından araştırmanın amaçlarına en uygun özelliklerini taşıyan katılımcıları kendi yargısıyla belirlemektedir (Balcı, 2004; Yıldırım ve Şimşek, 2008). Araştırmaya katılan öğrenciler daha önce Scratch programlama dili eğitimi almamışlardır.

Veri Toplama Araçları

Programlama eğitime yönelik geliştirilen mobil öğrenme uygulamasının kullanılacağı bu araştırmada nicel bir veri toplama aracı olan programlama becerisi başarı testi kullanılmıştır. Nicel veri toplama aracı araştırmacı tarafından geliştirilmiştir.

Programlama becerisi başarı testi.

Programlama Becerisi Başarı Testi, Scratch programlama dili eğitime yönelik geliştirilmiştir (Ek 1). Başarı testi, Bilişim Teknolojileri ve Yazılım dersinin herhangi bir yıllık planı olmadığı için dersin öğretmenlerinin belirlediği yıllık plandaki “Scratch Eğitimi” ünitesini kapsamaktadır. Bu dersin, Milli Eğitim Bakanlığı’nın çıkardığı MEGEP modülü veya ayrıca tavsiye ettiği bir kitap olmadığı için öğretmenler, Talim ve Terbiye Kurulu Başkanlığının yayınladığı dersin öğretim programındaki hedeflerine göre kendileri ünitelerini belirlemişlerdir. Bu üniteler arasında dersin öğretim programında yer alan programlama öğrenim hedefine göre belirlenen “Scratch Eğitimi” ünitesi bulunmaktadır. Başarı testindeki sorular bu üniteye göre hazırlanmıştır.

Başarı testi geliştirilirken önce Scratch Eğitimi ünitesini kapsayan kazanımların bulunduğu bir belirtke tablosu oluşturulmuştur. Belirtke tablosundaki kazanımlara göre toplam 27 soru hazırlanmıştır. Belirtke tablosu Ek 2’de yer almaktadır. Kapsam geçerliliğinin ve öğrenciye uygunluğunun belirlenmesi amacıyla, belirtke tablosundaki kazanımlar ve bu kazanımlara göre hazırlanan 27 soru, 4 alan uzmanı (2 bilgisayar ve öğretim teknolojileri eğitimi yüksek lisans mezunu öğretim elemanı, 2 bilişim teknolojileri öğretmeni) tarafından incelenmiştir. Alan uzmanlarının geri dönütlerine göre gerekli düzenlemeler yapılmıştır. Oluşturulan başarı testi, pilot çalışma için daha önce bu dersi aynı öğretmenden alan ortaokul 7. Sınıfta öğrenim gören 98 öğrenciye uygulanmıştır. Pilot çalışma sonunda bazı öğrenciler ile soruların anlaşılabilirliğine dair görüşmeler yapılmıştır. Öğrenciler, başarı testindeki soruların

anlaşılır ve okunabilir olduğunu belirtmişlerdir. Pilot çalışma sonucunda, doğru cevaplar için 1 puan, yanlış cevaplar veya boşluklar için 0 puan verilerek toplam puanlar elde edilmiştir. Toplam puanlar elde edildikten sonra başarı testinin “madde ayırt edicilik indeksi” ve “madde güçlük indeksleri” belirlenmesi için toplam puanlar en yüksek puandan en düşük puana göre sıralanmıştır. Sıralamaya göre, en yüksek puana sahip öğrenciden başlayarak %27’lik üst grup, en düşük puana başlayarak %27’lik alt grup belirlenmiştir. Üst ve alt gruplar belirlendikten sonra madde ayırt edicilik ve güçlük indeksleri hesaplanmıştır. Madde ayırt ediciliği hesaplanırken çıkan değerler 0 ve 1 arasında olmaktadır. Madde ayırt ediciliği .20 değeri altında olan sorular kesinlikle başarı testinde kullanılamaz ve testten çıkarılmalıdır. Madde güçlük indeksi hesaplamasında ise soru eğer .00 – .20 arasında ise kolay soru, .20 – .80 arasında ise normal soru, .80 – .00 arasında ise kolay soru olmaktadır. Hesaplamalar sonucunda değerlendirilen toplam 27 sorunun madde ayırt edicilik indeksi .20’nin altında olan 3 soru başarı testinden çıkarılmıştır. Kalan 24 sorunun madde ayırt edicilik ve madde güçlük indeksleri Tablo 1’de gösterilmiştir.

Tablo 1. Başarı Testi Madde Güçlük ve Madde Ayırt Edicilik İndeksleri

Soru	Madde Güçlüğü (P _j)	Madde Ayırt Edicilik İndeksi (r _j)	Soru	Madde Güçlüğü (P _j)	Madde Ayırt Edicilik İndeksi (r _j)
1	.85	.30	13	.69	.48
2	.81	.30	14	.57	.33
3	.74	.37	15	.59	.52
4	.57	.56	16	.83	.33
5	.61	.56	17	.57	.85
6	.52	.37	18	.85	.30
7	.46	.26	19	.50	.78
8	.31	.41	20	.44	.52
9	.35	.41	21	.70	.59
10	.52	.74	22	.56	.67
11	.65	.56	23	.59	.59
12	.83	.26	24	.54	.63
Ortalama Madde Güçlük İndeksi = .61					
Ortalama Ayırt Edicilik İndeksi = .49					

Başarı testinin son hali verildikten sonra KR-20 (Kuder Richardson-20) güvenilirlik analizi için SPSS 21 kullanılmıştır. SPSS’de yapılan analiz sonucunda testi güvenilirlik katsayısı cronbach's alpha değeri .90 olarak bulunmuştur. Başarı testinin ortalama madde ayırt edicilik indeksi .49 ve madde güçlük indeksi ise .61 olarak bulunmuştur. Kodlama başarı testinin oluşturulma aşamaları Şekil 10’da gösterilmiştir.



Şekil 10. Başarı Testinin Geliştirilme Aşamaları

Verilerin Analizi

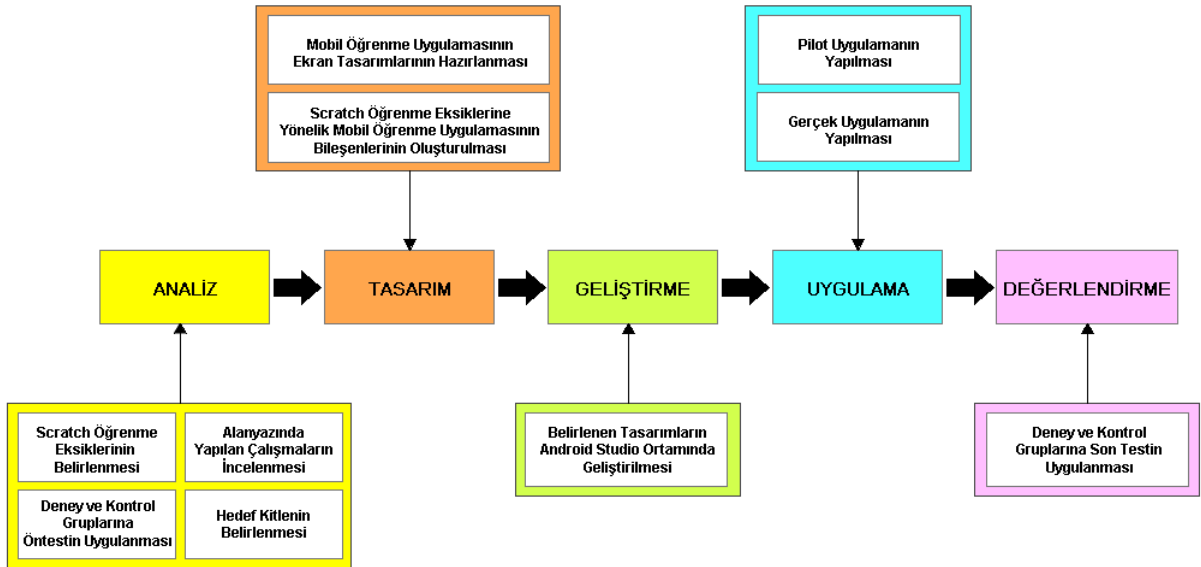
Araştırma sürecinde elde edilen verilerin analizinde SPSS 21 programı kullanılmıştır. Grupların programlama becerileri başarısı ön test ve son test sonuçları bağımsız örneklem t-testi ile analiz edilmiştir.

Bölüm IV: Bulgular

Araştırmanın bu bölümünde araştırmanın alt amaçlarına ait bulgulara yer verilmiştir.

Scratch Programlama Diline Yönelik Mobil Öğrenme Uygulamasının Geliştirilmesi

Bu bölümde öğrencilere Scratch programlama dili eğitimine yönelik bir mobil öğrenme uygulaması geliştirilmiştir. Bu aynı zamanda araştırmanın birinci alt amacına cevap niteliğindedir. Scratch programlama dili mobil öğrenme uygulaması öğretim tasarımı modellerinden ADDIE modeline göre geliştirilmiştir. ADDIE öğretim tasarımı modelinde toplamda 5 aşama bulunmaktadır. Bunlar; Analiz (Analysis), Tasarım (Design), Geliştirme (Design), Uygulama (Implementation) ve Değerlendirme (Evaluation) aşamasıdır. ADDIE modeline uygun olarak mobil öğrenme uygulamasının geliştirme süreci Şekil 11’de gösterilmektedir.



Şekil 11. ADDIE Tasarım Modeline Göre Mobil Öğrenme Uygulamasının Geliştirilmesi

Analiz aşaması.

Bu aşamada ihtiyaçlar ve hedef kitle belirlenmiştir. Bilişim teknolojileri öğretmenleri ile yapılan görüşmeler sonucunda öğrencilerin Scratch programlama dili ile kodlama

yaparken öğretmenin olmadığı durumlarda bazı öğretimsel eksikliklerin olduğu görülmektedir. Bu eksikler;

- Öğrencilere kodlama yaparken herhangi bir amaç veya hedef verilmemesi,
- Bir problemin çözümü için algoritma oluşturma sürecinde ortaya çıkan hatalar ve yanlış kodlama mantığı ile ilgili bir dönütün, bilgilendirmenin ve yönlendirmenin bulunmaması,
- Öğrencilerin hangi konuda bilgi eksiği olduğunu veya hangi konuları çalışması gerektiği ile ilgili kendini değerlendirebilme imkanının bulunmaması,
- Öğrencileri programlama öğrenme sürecinde motive edecek bildirimlerin bulunmaması.

Scratch programlama ortamının belirlenen tüm bu eksiklerinden dolayı öğrencilerin programlama öğrenmelerinde aksaklılara sebep olabileceği düşünülmektedir. Bu sorunlar ders içerisinde öğretmenler tarafından giderilebilmektedir. Ancak, öğrenci tek başına programlama yapmak zorunda kaldığında bu sorunlarla karşılaşabilmektedir. Bu bağlamda, geliştirilen mobil uygulama içerisine belirtilen öğretimsel eksikleri giderecek yazılımlar geliştirilecektir.

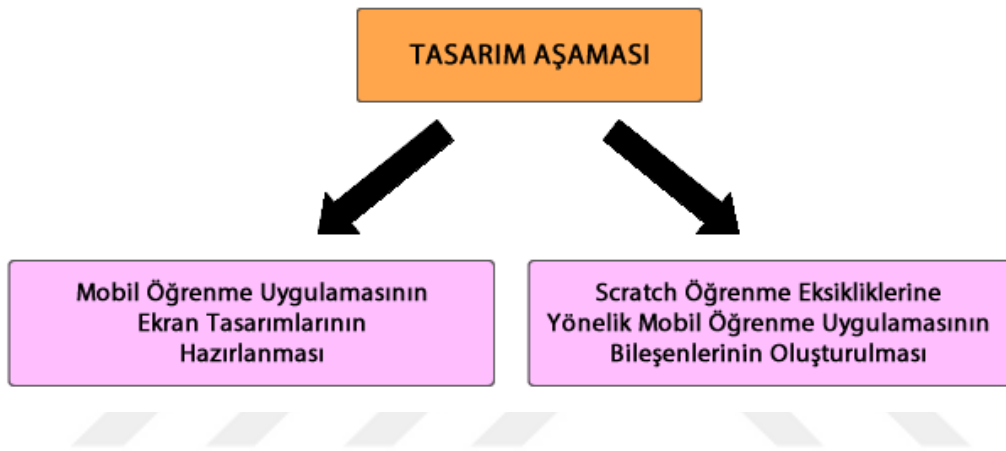
Türkiye’de Scratch ile programlama eğitimi genellikle 6. sınıf düzeyinde verildiğinden araştırmanın çalışma grubunu Kırklareli İstiklal Ortaokulu’nda eğitim gören 6. sınıf öğrencileri oluşturmuştur. Bu seviyedeki öğrenciler toplam 4 şubeye ayrılmıştır. Araştırmaya toplamda 94 öğrenci katılmıştır. Öğrencilerin okul başarı durumları ve ön test programlama becerileri başarı puanları incelenerek 2 sınıf deney grubu, 2 sınıf ise kontrol grubu olarak belirlenmiştir.

Scratch programlama dili eğitimi Bilişim Teknolojileri ve Yazılım dersi içerisinde bu dersin öğretmeni verdiğinden mobil öğrenme uygulamasının her aşamasının geliştirilmesinde

yardımcı olmuşlardır. Öğretmenler ile belirli zaman aralıklarıyla bir araya gelerek görüşleri alınmıştır.

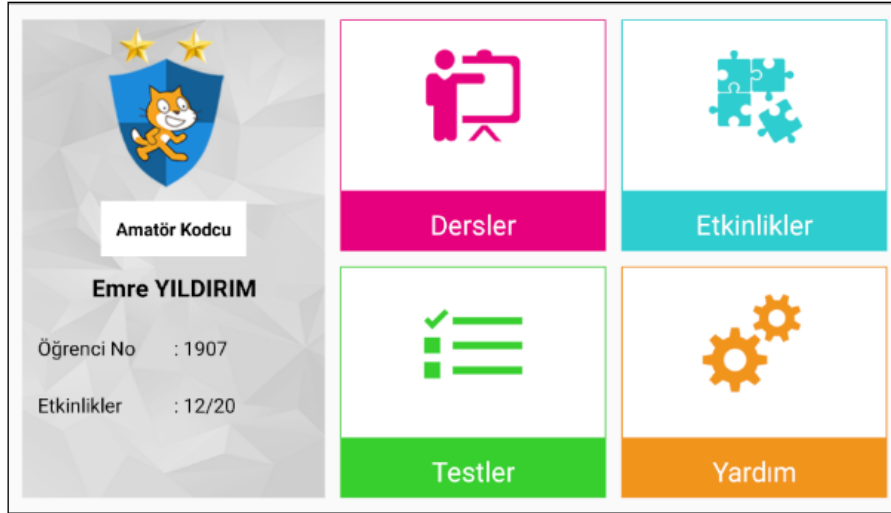
Tasarım aşaması.

Bu aşama 2 bölümden oluşmaktadır (Şekil 12). Birinci bölümde Scratch programlama dilinin mobil ortamda kullanılabilmesi için mobil uygulamanın gerekli tasarımları yapılmıştır. İkinci bölümde ise analiz aşamasında Scratch'ta belirlenen öğretim eksiklerine yönelik mobil uygulamanın öğretim içeriği belirlenmiştir.



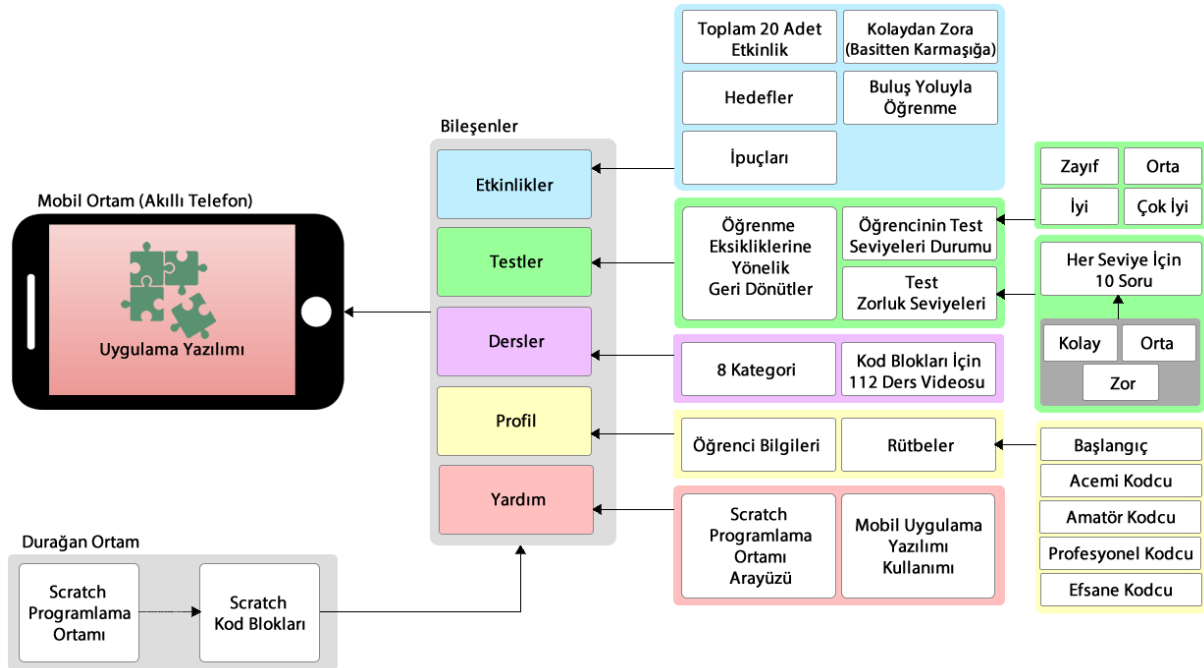
Şekil 12. Tasarım Aşaması

Birinci bölümde mobil öğrenme uygulamasının ekran tasarımları oluşturulmuştur. Ekran tasarımları Adobe Photoshop CS6 grafik programı kullanılarak yapılmıştır. Ekran tasarımları öğrencilerin yaş grubuna uygun olacak şekilde hazırlanmıştır. Bunun nedeni öğrencilerin mobil uygulamayı kullanmalarının cazip hale getirilmesinin düşünülmesidir. Öğretmenler ile görüşmeler sonucunda mobil öğrenme uygulamasının içerisinde 5 bileşen olması yönünde karar verilmiştir. Bu bileşenler; profil, dersler, etkinlikler, testler ve yardımdır. Mobil öğrenme uygulamasında kullanılan bileşenlerin toplu bir şekilde gösterimi ve aynı zamanda mobil uygulamanın ana ekran görüntüsü Şekil 13'te gösterilmektedir.



Şekil 13. Mobil Öğrenme Uygulamasının Ana Ekran Görüntüsü

İkinci bölümde ise Scratch'ta belirlenen öğretim eksiklerine yönelik mobil uygulamanın öğretim içeriği oluşturulmuştur. Bu amaca yönelik birinci bölümde tasarımları belirlenen bileşenlerin içerikleri oluşturulmuştur. Mobil uygulama bileşenleri içeriklerinin nasıl oluşturulduğu Şekil 14'te gösterilmektedir.



Şekil 14. Mobil Öğrenme Uygulamasının Bileşenlerinin Oluşturması

Etkinlikler bileşeni.

Etkinlikler içerisinde öğrencilere belirli bir hedef sunarak onlardan bu hedefler doğrultusunda düzgün algoritmalar kurabilecekleri toplam 20 etkinlik tasarlanmıştır. Etkinlikler bileşeninin giriş ekranı Şekil 15’te gösterilmektedir.



Şekil 15. Etkinlikler Bileşeninin Ana Ekranı

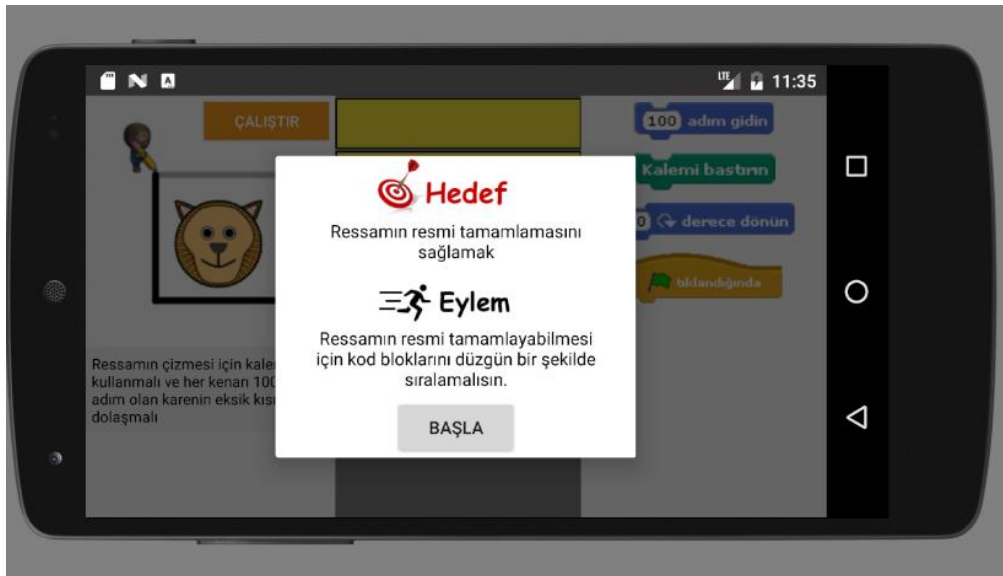
Etkinlikler bileşenindeki etkinlikler belirli kazanımlar dahilinde oluşturulmuştur. Kazanımlara göre etkinliklerin sınıflandırılması Tablo 2’de verilmiştir. Tabloda kazanımların karşısında o kazanıma ait etkinliğin numarası yer almaktadır.

Tablo 2. Scratch ile Kodlama Eğitimi Kazanımlarına Göre Etkinliklerin Sınıflandırılması

Kazanım	Etkinlik Numarası
Eksik algoritmayı doğru çalışacak biçimde düzenler.	1, 2
Basit kod bloklarını problem çözümünde kullanır.	3, 4
Döngü ve koşul yapıları içeren algoritmalar oluşturur.	5, 6, 7, 10, 18, 20
Değişkenleri problem çözümünde kullanır.	8, 10, 12, 14, 20
Farklı kod bloklarını iç içe kullanarak probleme çözüm üretir.	9, 10, 14, 15, 19, 20
Karmaşık problemlerin çözümü için ayrı ayrı algoritmalar oluşturur.	11, 12, 13, 14, 16, 17

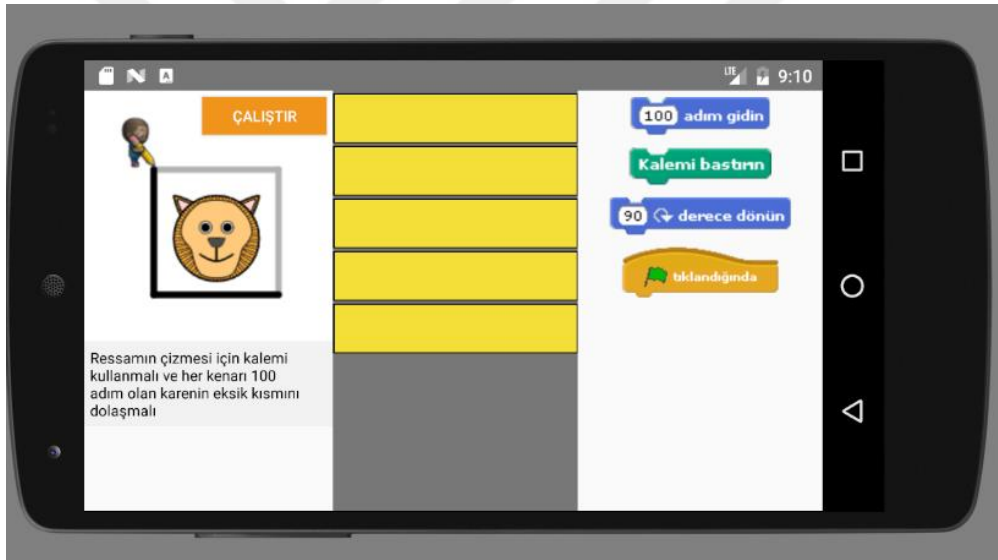
Öğrenmenin daha etkili ve verimli olabilmesi çeşitli öğretim ilkeleri ve öğretim stratejileri bulunmaktadır. Bu nedenle etkinlikler oluşturulurken belirli ilke ve stratejiler kullanılmıştır. Öncelikle etkinlikler basitten karmaşığa (kolaydan zora doğru) ilkesi kapsamında olacak şekilde oluşturulmuştur. Örneğin, 1. etkinlikte öğrenciye sadece sahnedeki karakterin adım atması öğretilmektedir. 2. etkinlikte ise öğrenciye sahnedeki karakterin adım atması aynı zamanda dönmesi öğretilmektedir. Böylece tüm etkinliklerin arasında mantıksal bağlantılar bulunmaktadır. Bu ilkenin kullanılmasının sebebi, öğrencinin özgüvenini ve programlamaya karşı olan tutumunu olumlu yönde etkilemektir. Ardından, etkinlikler keşfetme (buluş) yoluyla öğretme stratejisine göre oluşturulmuştur. Öğrenciler etkinliklerde verilen problemi kendi bilişsel becerilerini kullanarak çözmektedirler.

Belirlenen kazanımlar doğrultusunda etkinliklerin içerikleri tasarlanmıştır. Etkinlikler, Scratch programlama dilinin kod blokları kullanılarak oluşturulmuştur. Öğrencilere her etkinlikte çözmeleri için belirli problemler verilmektedir. Bu problemler, öğrencilerin karşısına etkinliklerde hedefler olarak sunulmaktadır. Verilen hedefler ve nasıl yapılması gerektiği ile eylemler öğrencilerin karşısına her etkinliğin ilk giriş ekranında verilmektedir (Şekil 16).



Şekil 16. Başlatılan Bir Etkinliğin İlk Giriş Ekranı: Hedef ve Eylem

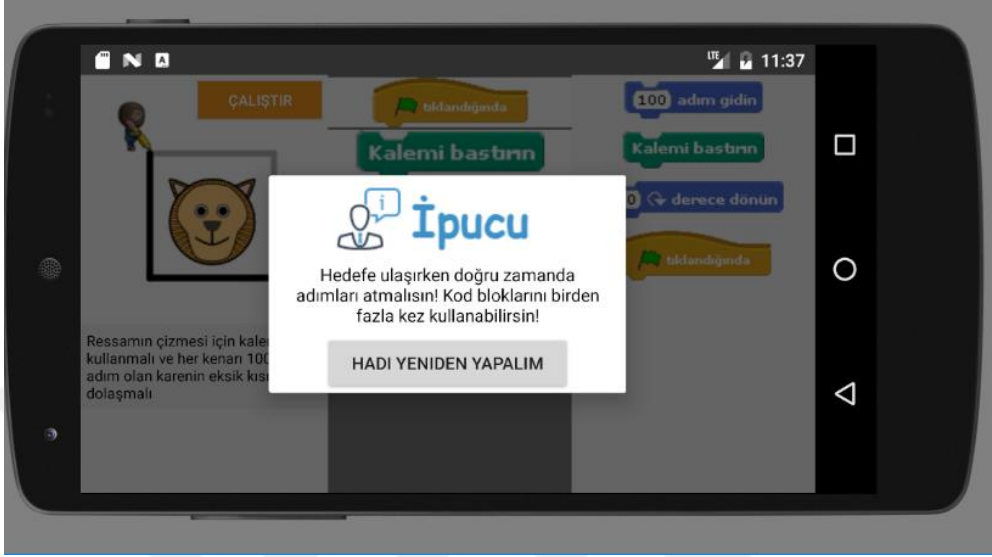
Öğrenci hedefler ve eylemler doğrultusunda etkinliğe başladığında karşısına üç bölümden oluşan bir ekran gelmektedir (Şekil 17). Solda yer alan birinci bölüm, öğrenciye verilen problemin görsel hali ve öğrencinin hedefe ulaşması için yapması gerekenler bölümüdür. Sağda yer alan ikinci bölüm, problemin çözümünde kullanılacak olan Scratch programlama dili kod bloklarının bulunduğu bölümdür. Ortada yer alan üçüncü bölüm ise kod bloklarının kullanılacağı kodlama bölümüdür. Kodlama bölümünde öğrenciye kaç tane kod bloğu kullanılacağı belirtilmektedir. Öğrenci bu doğrultuda, sürükle bırak yöntemini kullanarak doğru alanlara kod bloklarını yerleştirmektedir.



Şekil 17. Etkinlik Ekranı

Öğrenciler kod bloklarını yerleştirdikten sonra doğru algoritmayı oluşturup oluşturmadığını “Çalıştır” düğmesine basarak test edebilirler. Eğer öğrenciler problemin çözümünde doğru algoritmayı kuramazsa yazılım öğrencilerin kendi kurdukları algoritmaları üzerinden çıkarımlar yaparak karşılarına ipuçlarının olduğu bir ekran çıkarmaktadır. Öğrencilerin verilen ipuçlarını okuduktan sonra doğru algoritmayı tekrar kurmaları için “Hadi

Yeniden Yapalım” düğmesine basmaları gerekmektedir. Verilen ipuçlarını değerlendiren öğrenciler doğru algoritmayı kurmaya çalışmaktadırlar. İpuçları ekranı sayesinde öğrencilerin doğru algoritma kurmalarına yardımcı olarak problemi çözmeleri sağlanmaktadır (Şekil 18).



Şekil 18. Öğrenciyi Doğru Algoritma Kurması İçin Verilen İpucu Ekran Örneği

Öğrenciler problemin çözümünde doğru algoritmayı oluşturduklarında tekrar “Çalıştır” düğmesine basarak etkinliği başarılı bir şekilde bitirdiklerine dair karşılırlarına ekran çıkmaktadır (Şekil 19).



Şekil 19. Etkinlikte Başarılı Olunması Durumunda Öğrenciye Verilen Dönüt Ekranı

Öğrenciler etkinliği başarı bir şekilde yaptıktan sonra etkinlikler bileşenin ana ekranında “Sarı” olarak gösterilen etkinlik numarası “Yeşil” olarak gösterilmektedir (Şekil 15).

Testler bileşeni.

Testler bileşeni, öğrencilerin Scratch programlama dili programlama becerisi dahilinde kendini değerlendirdiği bileşendir. Öğrenciler, testler bileşenin içerisine girdiğinde karşısına gelen ekranda kendi test istatistiklerini, test sonucuna göre öğrenme durumunu ve kendisini değerlendirebileceği testler görmektedirler (Şekil 20). Testler, kolay, orta ve zor olmak üzere toplamda 3 seviye olarak sınıflandırılmaktadır. Her seviyede 10’ar soru bulunmaktadır. Test istatistikleri bölümünde öğrencinin seçmiş olduğu testte toplamda kaç doğru ve kaç yanlış yaptığı gösterilmektedir. Test durumunda ise, o testteki yapmış olduğu başarısına göre tavsiyeler sunulmaktadır.



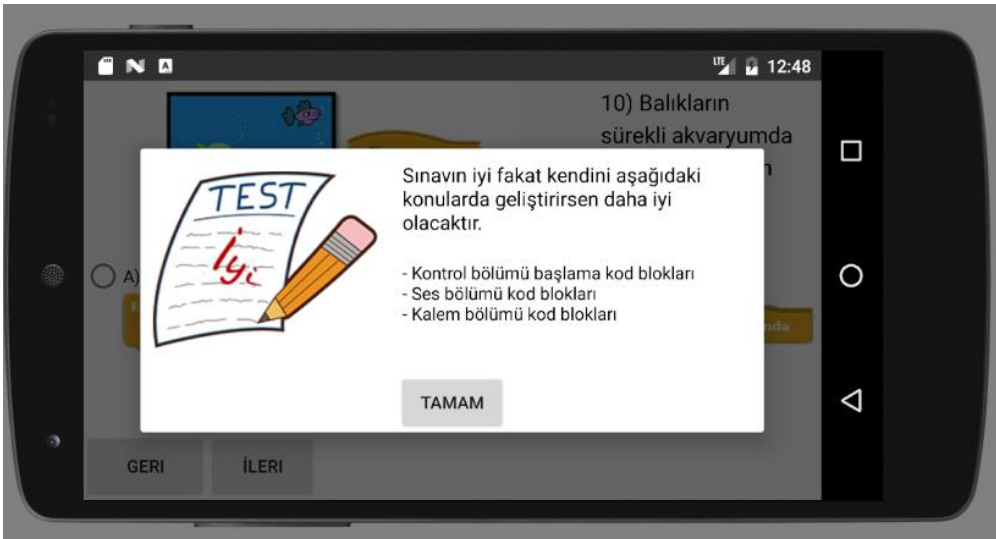
Şekil 20. Testler Bileşenin Ana Ekranı

Öğrenciler farklı seviyelerde geliştirilen testlerden birine giriş yaptıktan sonra o test ile ilgili sorular karşısına birer birer gelmekte ve onları cevaplamaktadır (Şekil 21).



Şekil 21. Testlerden Gelen Soru Örneği

Öğrenciler testteki tüm soruları cevapladıktan sonra karşısına, test sonuçlarına ilişkin test sonu ekranı gelmektedir (Şekil 22). Burada öğrencilerin kaç soruyu doğru ve yanlış cevapladıkları gösterilmekte ve testteki öğrenme durumlarının hangi seviyede olduğu görülebilmektedir. Aynı zamanda hangi konularda eksik olduklarını da görebilmektedirler. Buradaki sonuç ekranını aynı zamanda testler bileşenin ana ekranındaki test seviyelerini seçerek görebilmektedirler.

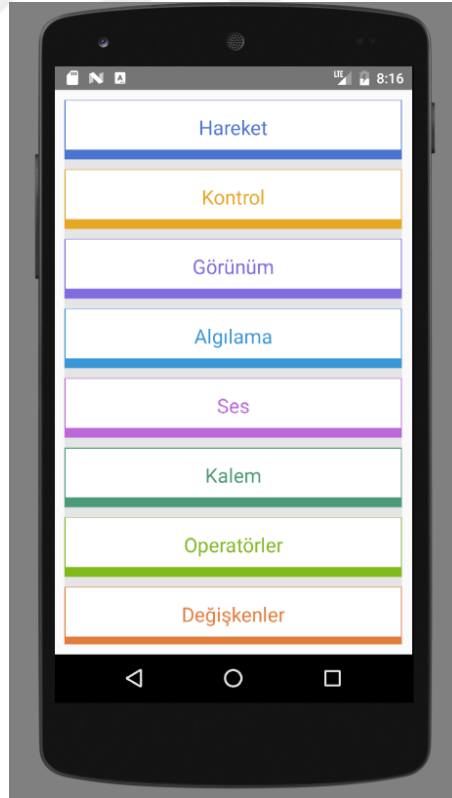


Şekil 22. Test Sonu Ekranı

Öğrenciler testlerde kendilerini değerlendirdikten sonra testte verilen tavsiyeler ışığında o konulara çalışarak kendilerini aynı testlerde tekrardan değerlendirebilmektedirler.

Dersler bileşeni.

Dersler bileşeninde Scratch yazılımında kullanılan 8 kod bloğu kategorileri ve bu kategoride bulunan kod bloklarına ait toplam 112 tane kod bloğunun nasıl kullanılması gerektiğine ait açıklamalı eğitim videoları bulunmaktadır. Bu eğitim videoları, daha önceden yapılmış bir çalışmadan izin alınarak bu çalışmaya eklenmiştir (Ek 3). Öğrenciler dersler bileşeninin içerisine girdiğinde, 8 farklı renkte bulunan kod blokları kategorileri karşılaşmaktadır. Bu kategorilerde kullanılan renkler Scratch programlama ortamında bulunan kod blokları kategorileri ile aynı renktedir. Buradaki amaç, öğrencilerin Scratch programındaki göz alışkanlığını kullanarak onlara kullanım sırasında kolaylık sağlamaktır. Bu menünün ekran tasarımı Şekil 23'te gösterilmektedir.



Şekil 23. Kod Blokları Kategorileri Menüsü

Öğrenciler hangi kategorideki kod bloğunu öğrenmek isterse o kategorinin içerisine girerek öğrenmek istediği kod bloğunun eğitim videolarına erişebilmektedir. Öğrenciler seçmiş oldukları dersin eğitim videosunda o kod bloğunun Scratch programlama ortamında nasıl kullanılması gerektiği ile ilgili örnekleri izleyebilmektedirler (Şekil 24).



Şekil 24. Kod Blokları İle İlgili Örnek Bir Ders

Öğrenciler ders bileşeninde bulunan ders videolarını izleyip nasıl çalıştıklarını öğrendikten sonra etkinlikler bileşenindeki etkinlikleri yapabilmektedirler. Bununla birlikte, testler bileşenindeki testleri de çözerek kendilerini değerlendirebilmektedirler.

Profil bileşeni.

Profil bileşeni, öğrencilerin mobil öğrenme uygulamasına ilk girdiklerinden itibaren görebilecekleri bir bileşendir. Burada öğrencilere kendileri ile ilgili bilgiler sunulmaktadır (Şekil 25). Bu bilgilerde;

- Öğrencilerin sahip oldukları rütbelerin sözel ve görsel olarak gösterimi,
- Öğrencinin adı, soyadı ve numarası,
- Etkinlik bileşeninde yapmış oldukları etkinlik sayıları bulunmaktadır.



Şekil 25. Profil Bileşeni Ekran Görüntüsü

Mobil öğrenme uygulamasında öğrencilerin motivasyonlarını üst seviyede tutmak amacıyla performanslarına göre rütbeler tasarlanmıştır. Rütbeler 5 kademe olarak tasarlanmıştır. Tasarlanan 5 kademe rütbe Tablo 3’de gösterilmektedir. Öğrencilerin belirli kademelerdeki rütbelere sahip olabilmeleri için etkinlikler bileşeni içerisindeki etkinlikleri yapmak zorundadırlar. Başarılı olarak yaptıkları her 5 etkinlik sonunda öğrenciler 1 rütbe kazanmaktadır. Kazandıkları rütbeler ise profil bileşeninde görsel olarak gösterilmektedir.

Tablo 3. Etkinliklere Göre Kazanılan Rütbe Seviyeleri

Başlangıç (0-4 Etkinlik)	Çaylak Kodcu (5-9 Etkinlik)	Amatör Kodcu (10-14 Etkinlik)	Profesyonel Kodcu (15-19 Etkinlik)	Efsane Kodcu (20 Etkinlik)

Profil bileşenindeki etkinlik sayıları ise öğrencilerin etkinlik bileşeninde başarılı oldukları etkinlik sayılarıdır. Buradaki amaç, öğrencinin etkinliklerdeki ilerleme durumlarını kendilerine göstermektir.

Yardım bileşeni.

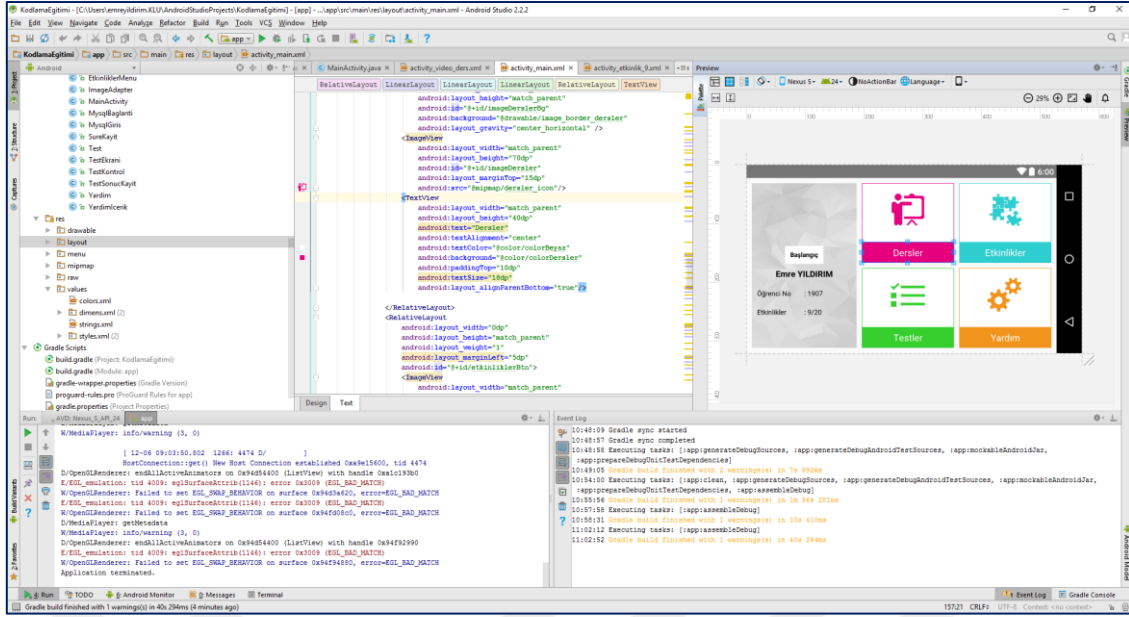
Yardım bileşeni iki bölümden oluşmaktadır. Birinci bölümde, Scratch yazılımının arayüz tanıtımı ile ilgili bilgiler bulunmaktadır. Öğrenciye bu bölümde Scratch arayüzünü kullanımı hakkında bilgiler verilmektedir. İkinci bölümde ise, Mobil öğrenme uygulamasının kullanımı hakkında öğrenci bilgilendirilmektedir. Böylece öğrenci mobil öğrenmeyi kullanmak için dışarıdan destek almak zorunda kalmayabilir. Bu bileşene ait ekran çıktısı Şekil 26’da gösterilmektedir.



Şekil 26. Yardım Menüsü İçeriği

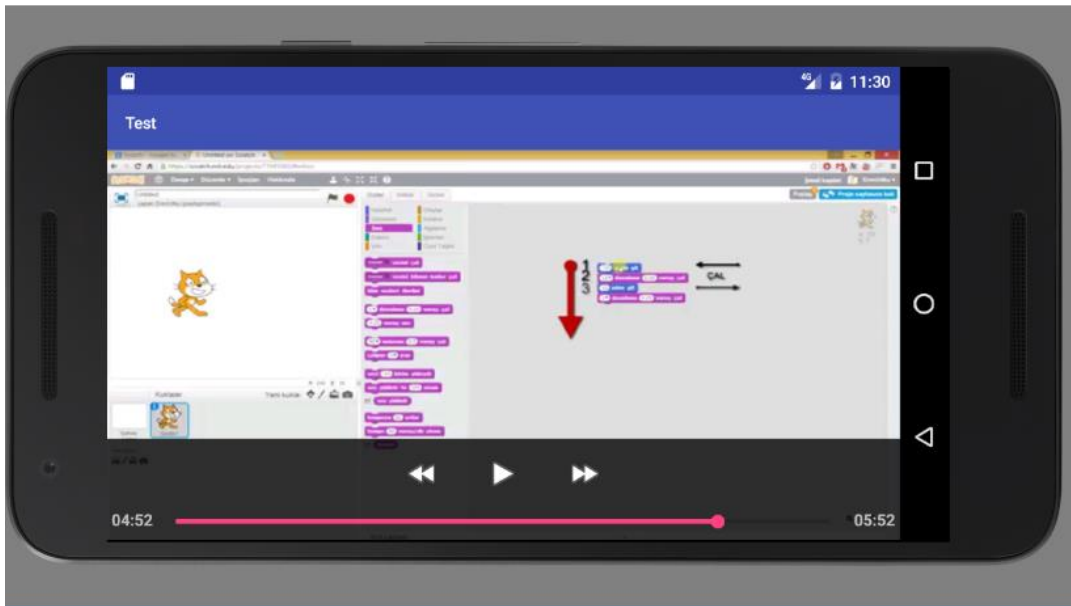
Geliştirme aşaması.

Tasarım aşamasında belirlenen ekran tasarımları ve eğitim içerikleri bu aşamada birleştirilerek mobil öğrenme uygulaması olarak geliştirilmiştir. Mobil öğrenme uygulaması Android işletim sistemi tabanlı bir yazılım olacağından Android Studio program geliştirme yazılımından yararlanılmıştır (Şekil 27).



Şekil 27. Android Studio Yazılım Geliştirme Platformu

Android Studio ile uygulama geliştirilirken içerisinde bulunan Android Emulator'u sayesinde mobil uygulamayı cihazlara atmadan bilgisayar ortamında yazılım gerçek ortamdaki gibi kullanılabilir. (Şekil 28). Mobil öğrenme uygulaması geliştirirken Java programlama dili kullanılmıştır.



Şekil 28. Kod Blokları Ders Videosu Örneği

Uygulama aşaması.

Bu aşamada, geliştirilen mobil öğrenme yazılımı için ilk önce pilot uygulama yapılmıştır. Daha sonra yazılım, gerçek uygulama olarak deney grubundaki öğrenciler tarafından kullanılmıştır. Pilot uygulama için Scratch ile kodlama eğitimi almış 7. sınıf kademesinde öğrenim gören toplam 10 öğrenci seçilmiştir. Öğrencilere uygulamayı kullanmaları için bir hafta süre verilmiştir. Bir hafta sonra mobil uygulamayı kullanan öğrenciler ile sınıf ortamında toplu bir şekilde görüşmeler yapılmıştır. Araştırmacı görüşme sonunda öğrencilerden gelen dönütleri araştırmacı günlüğüne not almıştır. Öğrenciler tarafından verilen dönütler içerisinde tavsiyeler, hatalar ve eksiklikler bulunmaktadır. Bunlar aşağıdaki gibidir;

- Mobil öğrenme uygulamasının çalışmasının yavaş olduğu,
- Etkinlikleri yaparken aniden etkinlikten atması ve programın kapanması,
- Etkinlik bileşenindeki etkinliklerin az olduğu,
- Testlerdeki soru sayısının az olduğu,
- Test çözümünde ileri butonunun olması fakat geri butonunun olmaması,
- Testlerin tekrar tekrar yapılamamasıdır.

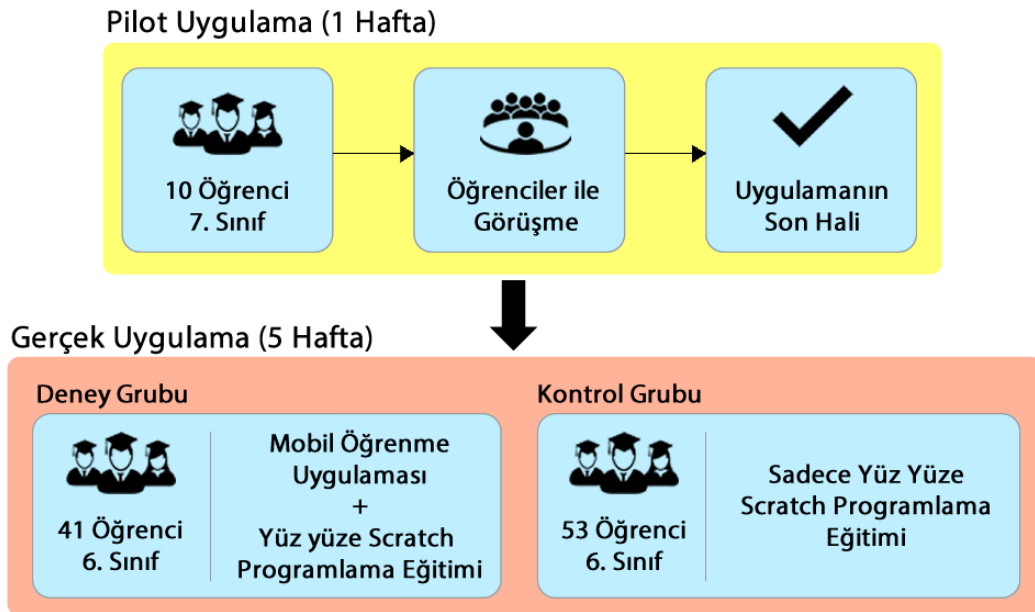
Öğrencilerden alınan yukarıdaki tavsiyeler, hatalar ve eksiklikler değerlendirilerek mobil öğrenme uygulamasının son hali verilmiştir. Ardından gerçek uygulama sürecine geçilmiştir.

Gerçek uygulama süreci, 2016-2017 eğitim öğretim yılının güz döneminde gerçekleştirilmiştir. Uygulama sürecine, Kırklareli İstiklal Ortaokulu'nun 6.sınıf öğrencileri katılmıştır. Çalışmanın bu okulda yapılabilmesi için gerekli izinler Kırklareli Milli Eğitim Müdürlüğü'nden alınmıştır (Ek 4). Bu düzeydeki öğrencileri seçmedeki amaç, öğrencilerin Bilişim Teknoloji ve Yazılım dersinde Programlama eğitimi almalarıdır. Böylelikle öğrencilerin diğer derslerden zamanları alınmayarak derslerden geri kalmamaları sağlanmıştır.

Uygulama sürecine 6. sınıfta öğrenim gören toplam 94 öğrenci katılmıştır. Deney grubunda 41 öğrenci ve kontrol grubunda 53 öğrenci bulunmaktadır. Deney grubunun kontrol grubuna göre daha az olmasının sebebi deney grubu olarak seçilen şubelerdeki bazı öğrencilerin akıllı telefonlarında Android işletim sisteminin bulunmaması veya telefonlarının olmamasıdır.

Deney grubunda bulunan öğrenciler akıllı telefonlarından Scratch programlama dili mobil öğrenme uygulamasını kullanarak programlama eğitimi almaktadırlar. Aynı zaman da sınıf içinde de ders öğretmeninden programlama eğitimi almaktadırlar. Mobil öğrenme uygulaması Google Play market uygulamasına yüklenemediği için öğrencilerin yanında elden verilerek akıllı telefonlarına yüklenmiştir. Mobil uygulamanın kullanımı ile ilgili kendilerine uygulama öncesi eğitim verilmiştir.

Kontrol grubundaki öğrenciler ise sadece ders öğretmeninden programlama eğitim almaktadırlar. Bunun yanında her iki grup da aynı öğretmen tarafından programlama eğitimi almışlardır. Her iki gruptaki öğrencilere uygulama öncesi ve sonrası Programlama Becerisi Başarı Testi uygulanmıştır. Gerçekleştirilen uygulama süreci toplam 5 hafta boyunca devam etmiştir. Uygulama aşaması Şekil 29’da gösterilmektedir.



Şekil 29. Uygulama Aşamasının Şekilsel Gösterimi

Deney ve kontrol grubundaki öğrenciler Scratch programlama eğitimlerine 5 hafta boyunca devam etmişlerdir. Bu süre sonunda her iki grubun programlama becerileri değerlendirilmiştir. Değerlendirme süreci *4.1.5 Değerlendirme Aşaması* başlığı altında ayrıntılı bir şekilde açıklanmıştır.

Değerlendirme aşaması.

Deney ve kontrol grubundaki öğrenciler 5 hafta boyunca Scratch programlama eğitimlerini aldıktan sonra kendilerine araştırmacı tarafından geliştirilen Programlama Becerisi Başarı Testi uygulanmıştır.

Değerlendirme sonunda öğrencilerden toplanan veriler analiz edilmiştir. Analizler ve bulgular, ayrı başlıklar altında açıklanmıştır.

Öğrencilerin Programlama Becerileri Ön Test ve Son Test Başarı Puanlarına İlişkin

Bulgular

Bu bölümde, veri toplama araçlarından elde edilen verilerin analizinde öğrencilerin programlama becerileri ön test ve son test başarı puanlarına yönelik bulgular yer almaktadır. Bu bulgular aynı zamanda, araştırmanın alt amaçlarında belirlenen ikinci alt amaca yönelik cevap niteliğindedir.

Ön test programlama becerileri başarı puanlarına ilişkin bulgular.

Grupların ön test programlama becerileri başarı puanları analizi yapılmadan önce, verilerin normal dağılıma uygunluğu test edilmiştir. Verilerin normal dağılım gösterip göstermediğini test etmek için Shapiro-Wilk testi uygulanmıştır. Bu test sonucuna göre her iki grubun verileri .05 anlamlılık düzeyinden yüksek olduğu için veriler normal dağılım gösterdiği ifade edilebilir ($p > .05$). Aynı zamanda, grupların varyansların homojenliği ise Levene Testi kullanılarak analiz edilmiştir. Analiz sonucunda gruplar arasında anlamlı bir

farklılık görülmemiştir [$F= .121, p=.73>.05$]. Grupların ön test programlama becerileri başarı puanlarına ait betimsel istatistik ve normal dağılım testi sonuçları Tablo 5'te gösterilmektedir.

Tablo 5. Ön Test Programlama Becerileri Başarı Puanlarına İlişkin Betimsel İstatistik ve Shapiro-Wilk Normal Dağılım Testi Sonuçları

Grup	N	$\bar{X}$	Ss	Min	Max	Shapiro-Wilk
Deney	41	13.51	3.35	6	22	.83
Kontrol	53	12.36	3.09	6	19	.29

Deney ve kontrol gruplarının normal dağılım testi sonuçlarına göre veriler normal dağılım gösterdiğinden programlama becerileri başarı puanları arasındaki farklılığın incelenmesine yönelik parametrik test tekniklerinden bağımsız örneklem t-testi kullanılmıştır. Bağımsız örneklem t-testi sonuçları Tablo 6'da gösterilmektedir.

Tablo 6. Ön Test Programlama Becerileri Başarı Puanlarına İlişkin Bağımsız Örneklem t-Testi Sonuçları

Grup	N	$\bar{X}$	Ss	Sd	t	p
Deney	41	13.51	3.35	92	1.729	.87
Kontrol	53	12.36	3.09			

Tablo 6 incelendiğinde deney grubu ile kontrol grubunda bulunan öğrencilerin ön test programlama becerileri başarı puanları arasında anlamlı bir farklılık bulunmamaktadır ($p=.87>.05$). Bu nedenle, çalışma grubundaki öğrencilerin başlangıçtaki programlama becerileri benzerlik gösterdiği söylenebilir. Grupların ön test puanları arasında anlamlı bir farklılık çıkmadığından grupların programlama becerileri son test puanları karşılaştırılarak yapılmıştır.

Son test programlama becerileri başarı puanlarına ilişkin bulgular.

Grupların son test programlama becerileri başarı puanları analizi yapılmadan önce, verilerin normal dağılıma uygunluğu test edilmiştir. Verilerde normal dağılım olup olmadığını test etmek için Shapiro-Wilk testi uygulanmıştır. Bu test sonucuna göre her iki grubun verileri .05 anlamlılık düzeyinden yüksek olduğu için verilerin normal dağılım gösterdiği ifade edilebilir ($p > .05$). Aynı zamanda, grupların varyansların homojenliği Levene Testi kullanılarak analiz edilmiştir. Analiz sonucunda gruplar arasında anlamlı bir farklılık görülmemektedir [$F = 3.90, p = .051 > .05$].

Grupların son test programlama becerileri başarı puanlarına ait betimsel istatistik ve normal dağılım testi sonuçları Tablo 7’de gösterilmektedir.

Tablo 7. Son Test Programlama Becerileri Başarı Puanlarına İlişkin Betimsel İstatistik ve Shapiro-Wilk Normal Dağılım Testi Sonuçları

Grup	N	$\bar{X}$	Ss	Min	Max	Shapiro-Wilk
Deney	41	20.00	2.40	15	24	.21
Kontrol	53	18.68	3.30	9	24	.09

Deney ve kontrol gruplarının son test normal dağılım testi sonuçlarına göre veriler normal dağılım gösterdiğinden, son test programlama becerileri başarı puanları arasındaki farklılığın incelenmesi için parametrik test tekniklerinden bağımsız örneklem t-testi kullanılmıştır. Grupların son test bağımsız örneklem t-testi sonuçları Tablo 8’de gösterilmektedir.

Tablo 8. Son Test Programlama Becerileri Başarı Puanlarına İlişkin Bağımsız Örneklem t-Testi Sonuçları

Grup	N	$\bar{X}$	Ss	Sd	t	p	d
Deney	41	20.00	2.40	92	2.16	.034*	.46
Kontrol	53	18.68	3.30				

* $p < .05$

Tablo 8 incelendiğinde deney grubu ile kontrol grubunda bulunan öğrencilerin son test programlama becerileri başarı puanları arasında anlamlı bir farklılığın olduğu görülmektedir ($t(92)=2.157$, $p=.034<.05$, $d=.46$). Bu farklılığın ise, deney grubundaki öğrencilerin başarıları yönünde olduğu görülmektedir. Deney grubunun programlama becerisi başarı puanlarındaki ($\bar{X}=20.00$) gözlenen artış kontrol grubuna göre ($\bar{X}=18.68$) daha fazla olduğu ifade edilebilir. Ancak, her iki grubun da ön test ve son test puanları incelendiğinde, hem deney hem de kontrol grubunun programlama becerileri başarı puanlarında artış görülmektedir.

Bölüm V: Sonuç, Tartışma ve Öneriler

Bu bölümde, elde edilen bulgular ışığında oluşturulan sonuç, tartışma ve önerilere yer verilmiştir.

Sonuç ve Tartışma

Bu araştırmada ortaokul seviyesinde öğrenim gören öğrencilerin programlama eğitimlerine destek amaçlı Android işletim sistemi tabanlı Scratch programlama dili eğitimine yönelik mobil öğrenme uygulaması geliştirilmiştir. Bu mobil öğrenme uygulamasını kullanan öğrencilerin programlama becerilerine etkisi incelenmiştir.

Programlama becerileri başarı testi sonuçlarının bulguları incelendiğinde, araştırmada, hem deney grubunun hem de kontrol grubunun programlama becerilerinde artış görülmektedir. Deney grubundaki öğrencilerin programlama becerileri ön test $\bar{X}$ başarı ortalamaları $X=13.51$ iken son test başarı puanları $X=20.00$ olduğu görülmektedir. Kontrol grubunun programlama becerileri ön test başarı puanları $X=12.36$ iken son test başarı puanları $X=18.68$ olduğu görülmektedir. Bu bağlamda, mobil öğrenme uygulamasını kullanan ve kullanmayan öğrencilerin ön test sonuçları arasında anlamlı bir farklılık görülmemektedir. Ancak, son test sonuçları arasında anlamlı bir farklılığın olduğu görülmektedir. Bu farklılığın ise mobil öğrenme uygulamasını kullanan deney grubunun lehine olduğu görülmektedir. Son test sonuçlarının bulguları incelendiğinde mobil öğrenme uygulaması kullanan öğrencilerin programlama becerisi bakımından daha başarılı olduğu görülmektedir. Bu farklılığın mobil öğrenme uygulaması kullanımından kaynaklandığı sonucuna varılabilir. Mobil öğrenme uygulamasının içerisindeki derslerin, etkinliklerin ve testlerin öğrencilerin başarılarına etki ettiği düşünülebilir.

Yukarıda belirtilen sonuçlar ışığında, alanyazındaki benzer araştırmalar ile karşılaştırıldığında bu araştırmanın, Scratch programlama dili kullanımının öğrencilerin programlama becerilerini arttırdığı sonucunu desteklediği söylenebilir (Nam, Kim ve Lee, 2010; Osman ve diğerleri, 2012; Ouahbi ve diğerleri ,2015; Ruf, Mühling ve Hubwieser, 2014). Aynı zamanda mobil öğrenme uygulaması kullanımının öğrencilerin akademik performanslarına olumlu yönde etki ettiği sonucunu da desteklemektedir (Burmabıyık, 2014; Chen ve Chung; 2008; Dehmenoğlu, 2015; Fernández-López ve diğerleri, 2013; Kılıç, 2015; Köse, Koç ve Yücesoy, 2013; Skiada ve diğerleri, 2014; Sun ve diğerleri; 2015). Bu bağlamda, bu çalışmada geliştirilen mobil öğrenme uygulamasının ortaokul düzeyinde eğitim gören öğrencilerin programlama becerilerine katkı sağladığı sonucuna ulaşılabilir.

Öneriler

Bu çalışmada yapılan analizler sonucunda gelecekte yapılacak benzer araştırmalar için aşağıdaki önerilerde bulunulmuştur:

- Mobil öğrenme uygulamasının içerisindeki etkinliklerin sayısı artırılarak öğrencinin uygulamada daha fazla vakit geçirmesi sağlanabilir.
- Mobil öğrenme uygulamasının içerisindeki testlerin sayısı artırılabilir. Böylece öğrenci kendini daha doğru bir şekilde değerlendirebilir.
- Mobil öğrenme uygulamasını aynı koşullarda daha fazla öğrenci test edebilir. Aynı sınıf düzeyinde bulunan farklı okullardaki öğrenciler benzer şekilde uygulamayı kullanabilir.
- Scratch programlama dili eğitime yönelik yapılan bu çalışma, farklı grafik tabanlı programlama dilleri için de geliştirilerek öğrenciler tarafından test edilebilir.

- Mobil öğrenme uygulamasına mobil cihazların farklı özellikleri ve farklı eğitim içerikleri eklenerek uygulama geliştirilebilir.
- Mobil öğrenme uygulaması, programlama eğitimi için kullanılan diğer mobil uygulamalar ile karşılaştırılabilir.
- Mobil öğrenme uygulaması diğer mobil işletim sistemleri (IOS, Windows gibi) içinde geliştirilerek test edilebilir.
- Mobil öğrenme uygulamasına tasarım ve kodlama bakımından iyileştirmeler yapılabilir.
- Testlerdeki soru sayıları arttırılabilir. Soru ve yanıtlar yer değiştirilerek rastgele öğrencilere sunulabilir.
- Benzer bir araştırmada öğrenci görüşleri alınarak nitel bir araştırma yapılabilir.

Kaynakça

- Ako-Nai, F., Tan, Q., ve Pivot, F. C. (2012). The 5R Adaptive Learning Content Generation Platform for Mobile Learning. *In Technology for Education (T4E), 2012 IEEE Fourth International Conference.* 132-137.
- Balanskat, A., ve Engelhardt, K. (2014). *Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe.*
- Balcı, A. (2004). *Sosyal bilimlerde araştırma yöntem, teknik ve İlkeleri.* Ankara: Pegem Akademi Yayıncılık.
- Bennedsen, J., ve Caspersen, M. E. (2008). Exposing the programming process. *Reflections on the Teaching of Programming.* 6-16.
- Burmabıyık, A. (2014). Geometrik cisimlerin öğretimi için geliştirilen 3 boyutlu mobil uygulamalar hakkında öğrenci ve öğretmen görüşleri. *Yayımlanmamış Yüksek Lisans Tezi.* Balıkesir Üniversitesi. Balıkesir.
- Büyüköztürk, Ş. (2004). *Sosyal bilimler için veri analizi el kitabı.* (4. Baskı). Ankara: Pegem A Yayıncılık.
- Calder, N. (2010). Using scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom,* 15(4), 9-14.
- Chen, G. D., Chang, C. K., ve Wang, C. Y. (2008). Ubiquitous learning website: Scaffold learners by mobile devices with information-aware techniques. *Computers & Education,* 50(1), 77-90.
- Chen, C. M., ve Chung, C. J. (2008). Personalized mobile English vocabulary learning system based on item response theory and learning memory cycle. *Computers & Education,* 51(2), 624-645.
- Clements, D. H. (1985). Logo programming: Can it change how children think. *Electronic Learning,* 4(4), 28.

- Clements, D. H., ve Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051-1058.
- Çetin, E. (2012). Bilgisayar Programlama Eğitiminin Çocukların Problem Çözme Beceleri Üzerine Etkisi. *Yayınlanmamış Yüksek Lisans Tezi*. Gazi Üniversitesi. Ankara.
- Dehmenoğlu, C. (2015). Programlama temelleri dersine yönelik mobil öğrenme aracının geliştirmesi. *Yayınlanmamış Yüksek Lisans Tezi*. Bahçeşehir Üniversitesi. İstanbul.
- Demirer, V. ve Sak, N. (2016). Programming education and new approaches around the world and in Turkey. *Eğitimde Kuram ve Uygulama*, 12 (3), 521-546.
- Dönmez, O., Gelibolu, M. F., ve İncoğlu, M. M. (2006). Eğitim Teknolojisinin Yeni Yüzü Mobil Öğrenme. *International Educational Technology Conference. Famagusta, North Cyprus*.
- Eckerdal, A. (2009). Novice programming students' learning of concepts and practise. *Doctoral dissertation, Acta Universitatis Upsaliensis*.
- Erdoğan, Y., Aydın, E. ve Kabaca, Y.T. (2008). Exploring the psychological predictors of programming achievement. *Journal of Instructional Psychology*, 35(3), 264-270.
- Fernández-López, Á., Rodríguez-Fórtiz, M. J., Rodríguez-Almendros, M. L., ve Martínez-Segura, M. J. (2013). Mobile learning technology based on iOS devices to support students with special education needs. *Computers & Education*, 61, 77-90.
- Fesakis, G., Gouli, E., ve Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.
- Fesakis, G., ve Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin*, 41 (3), 258-262.

- Fischer, G. (2000). Lifelong learning-more than training. *Journal of Interactive Learning Research*, 11(3), 265.
- Genç, Z., ve Karakuş, S. (2012). Tasarımla Öğrenme: Eğitsel Bilgisayar Oyunları Tasarımında Scratch Kullanımı. *5th International Computer & Instructional Technologies Symposium*, 22-24 September 2011 Fırat University, ELAZIĞ-TURKEY.
- Gomes, A., ve Mendes, A. J. (2007). Learning to program-difficulties and solutions. *International Conference on Engineering Education-ICEE* .
- Goundar, S. (2011, December). What is the potential impact of using mobile devices in education. *Proceedings of SIG GlobDev Fourth Annual Workshop*.
- Gülbahar, Y., ve Kalelioğlu, F. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*, 113(1), 33-50.
- Hadjerrouit, S. (2008). Towards a blended learning model for teaching and learning computer programming: a case study. *Informatics in Education-An International Journal*, 17(3), 181-210.
- ITU. (2016). 28.06.2016 tarihinde <http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx> adresinden erişilmiştir.
- Jenkins, T. (2002). On the difficulty of learning to program. *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 4(2002), 53-58).
- Keegan, D. (2005). Mobile learning: the next generation of learning. *Distance Education International*, 137-143.
- Kesici, T., ve Kocabaş, Z. (2007). Bilgisayar 2 ders kitabı (2. Baskı). Ankara: Semih Ofset.

- Keskin, N. (2011). Akademisyenler için bir mobil öğrenme sisteminin geliştirilmesi ve sınanması. *Yayımlanmamış Doktora Tezi*.
- Kılıç, M. (2015). Mobil öğrenmeye dayalı android uygulamalarının öğrencilerin kimya dersi atom ve periyodik sistem ünitesindeki akademik başarılarına, kalıcı öğrenmelerine ve motivasyonlarına etkisi. *Yayımlanmamış Yüksek Lisans Tezi*. Kahramanmaraş Sütçü İmam Üniversitesi. Kahramanmaraş.
- Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia-Social and Behavioral Sciences*, 174, 227-232.
- Kodlama Atölyesi (2017). Çocuklar da Kodlama Öğreniyor!. 01 Mart 2017 tarihinde <http://kodlamaatolyesi.com> adresinden alınmıştır.
- Korkmaz, Ö. (2016). The effect of scratch and lego mindstorms ev3 based programming activities on academic achievement, problem solving skills and logical mathematical thinking skills of students. *Malaysian Online Journal of Educational Sciences*, 4(3), 73-88.
- Köse, U., Koç, D., ve Yücesoy, S. A. (2013). An augmented reality based mobile software to support learning experiences in computer science courses. *Procedia Computer Science*, 25, 370-374.
- Malan, D. J., ve Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, 39(1), 223-227.
- Mannila, L., Peltomäki, M., Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, 16(3), 211–227.
- MEB (2017). PISA 2015 Ulusal Raporu. 01 Mart 2017 tarihinde http://pisa.meb.gov.tr/wp-content/uploads/2016/12/PISA2015_Ulusal_Rapor1.pdf adresinden alınmıştır.

- MEB (2017). Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 ve 8. Sınıflar) Öğretim Programı. 01 Mart 2017 tarihinde <http://ttkb.meb.gov.tr/www/ogretim-programlari/icerik/72> adresinden alınmıştır.
- Nam, D., Kim, Y., ve Lee, T., (2010). The Effects of Scaffolding-Based Courseware for the Scratch Programming Learning on Student Problem Solving Skill. *Proceedings of the 18th International Conference on Computers in Education*.
- OECD (2017). PISA 2015 Report. . 01 Mart 2017 tarihinde <http://www.oecd.org/pisa/> adresinden alınmıştır.
- Ortiz L.A. (1997). A framework for visual programming languages. *Yüksek Lisans Tezi*, University of Puerto Rico.
- Osman, M. A., Loke, S. P., Zakaria, M. N., ve Downe, A. G. (2012). Secondary students' perfectionism and their response to different programming learning tools. In *Humanities, Science and Engineering (CHUSER), 2012 IEEE Colloquium on IEEE*. 584-588.
- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., ve Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, 191, 1479-1482.
- Özsoy, G. (2005). Problem çözme becerisi ile matematik başarısı arasındaki ilişki. *Gazi Eğitim Fakültesi Dergisi*, 25(3), 179-190.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S. (1993). The children's machine. *Technology Review-Manchester Nh*, 96, 28-28.
- Park, N. (2013). Application and Analysis of STEAM using Education Programming Language in Elementary School. *International Information Institute (Tokyo). Information*, 16(10), 7311.

- Park, Y. (2011). A pedagogical framework for mobile learning: Categorizing educational applications of mobile technologies into four types. *The International Review of Research in Open and Distributed Learning*, 12(2), 78-102.
- Parker, J. R. (2016). *Python: An Introduction to Programming*. Stylus Publishing, LLC.
- Regan, G. O. (2008). *A brief history of computing*. Springer Science & Business Media.
- Resnick, M. (2012). Mother's Day, Warrior Cats, and Digital Fluency: Stories from the Scratch Online Community. In *Proceedings of the constructionism 2012 conference: Theory, practice and impact*, 52-58.
- Ruf, A., Mühling, A., ve Hubwieser, P. (2014). Scratch vs. Karel: impact on learning outcomes and motivation. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, 50-59.
- Sanjanaashree, P., Anand Kumar, M. ve Soman, K. P. (2014). Language learning for visual and auditory learners using scratch toolkit. In *Computer Communication and Informatics (ICCCI), 2014 International Conference on IEEE*. 1-5.
- Schwartz, J., Stagner, J., ve Morrison, W. (2006). Kid's programming language (KPL). *ACM SIGGRAPH 2006 Educators program* (p. 52). ACM.
- Scratch Web Sitesi. (2017). 01 Mart 2017 tarihinde <https://scratch.mit.edu/about/> adresinden alınmıştır.
- Sebesta, R. W., ve Mukherjee, S. (2002). *Concepts of programming languages*. Reading: Addison-Wesley.
- Seferoğlu, S. S., ve Akbıyık, C. (2006). Eleştirel düşünme ve öğretimi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 30(30).
- Shin, S., ve Park, P. (2014). A Study on the Effect affecting Problem Solving Ability of Primary Students through the Scratch Programming. *Advanced Science and Technology Letters*, 59, 117-120.

- Siegle, D. (2009). Developing student programming and problem-solving skills with visual basic. *Gifted Child Today*, 32(4), 24-29.
- Skiada, R., Soroniati, E., Gardeli, A., ve Zissis, D. (2014). EasyLexia: A mobile application for children with learning difficulties. *Procedia Computer Science*, 27, 218-228.
- Stenerson, M. E. (2012). *Academic game development: practices and design strategies for creating STEM games*. Iowa State University. ABD, Iowa.
- Sun, L., Hou, J., Hu, X., ve Al-mekhlafi, K. (2015). A Context-based Support System of Mobile Chinese Learning for Foreigners in China. *Procedia Computer Science*, 60, 1396-1405.
- Taylor, M., Harlow, A., ve Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences*, 8, 561-570.
- Tetenbaum, T. J., ve Mulkeen, T. A. (1984). LOGO and the teaching of problem solving: A call for a moratorium. *Educational Technology*, 24(11), 16-19.
- Traxler, J. (2005). Mobile Learning: It's here, but what is it. *Interactions*, 9(1), 1-12.
- Turgut, G. (2015). Okul öncesi eğitimi için geliştirilen hikaye oluşturma yazılımına yönelik görüşlerin incelenmesi. *Yayımlanmamış Yüksek Lisans Tezi*. Ege Üniversitesi. İzmir.
- Uden, L. (2006). Activity theory for designing mobile learning. *International Journal of Mobile Learning and Organisation*, 1(1), 81-102.
- Wang, H. Y., Huang, I., ve Hwang, G. J. (2014). Effects of an integrated Scratch and project-based learning approach on the learning achievements of gifted students in computer courses. *Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on IEEE*, 382-387.
- Westcott, S. (2008). *Effectiveness of using digital game playing in a first-level programming course*. Pace University.

- Valanides, N. (1997). Formal reasoning abilities and school achievement. *Studies in Educational Evaluation*, 23(2), 169-185.
- Vinu, P. V., Sherimon, P. C., ve Krishnan, R. (2011). Towards pervasive mobile learning—the vision of 21st century. *Procedia-Social and Behavioral Sciences*, 15, 3067-3073.
- Yıldırım, A. ve Şimşek, H. (2008). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri (6.Baskı)*. Ankara: Seçkin Yayıncılık.
- Yokuş, G. (2016). Eğitim fakültesi öğrencilerinin mobil öğrenmeye ilişkin görüşlerinin incelenmesi ve eğitim bilimlerine yönelik mobil uygulama geliştirme çalışması: Mobil Akademi. *Yayımlanmamış Yüksek Lisans Tezi*. Mersin Üniversitesi. Mersin.
- Yükseltürk, E., ve Altıok, S. (2015). Bilişim Teknolojileri Öğretmen Adaylarının Bilgisayar Programlama Öğretimine Yönelik Görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65.
- Zak, D. (2013). *Introduction to Programming with C++*. Cengage Learning.

Ekler

Ek 1. Programlama Becerisi Başarı Testi

Adı Soyadı :

Sınıfı :

Numarası :

Saygılı öğrenciler, Bu başarı testi sizin Programlama ile ilgili bilgilerinizi ölçme amacıyla hazırlanmıştır. Yanlış cevaplarınız doğru cevaplarınızı etkilemeyecektir. Cevaplarınızı soruların üzerine işaretleyiniz.

Başarılar Dilerim.

Emre YILDIRIM

Bilgisayar ve Öğretim Teknolojileri Eğitimi / Eğitim Bilimleri Enstitüsü
Çanakkale Onsekiz Mart Üniversitesi

SORULAR

1. Scratch programında hazırlanan çalışmaları **kaydetmek** için hangi menü kullanılır?

a) Düzen b) Dosya c) Görünüm d) Yardım

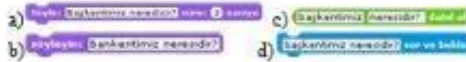
2. Sahneye **Yeni** bir karakter çizip eklemek için aşağıdakilerden hangisi kullanılır?



3. Verilen bir işin **10 defa** ardı ardına yapmasını istiyorsak aşağıdakilerden hangi kod bloğu kullanılmaktadır?



4. Yukarıdaki karakterimizin soru sorması ve cevap alabilmesi için hangi kod bloğu kullanılmalıdır?



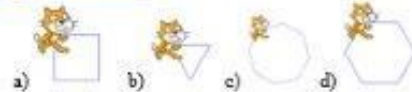
5. Yandaki iki kostümlü kedimizi yürütme animasyonu vermek istiyoruz. Hangi kod bloğunu kullanmalıyız?



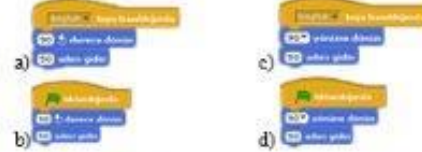
6. Sahnedeki karakterin yukarı doğru gitmesi için hangi kod bloğu kullanılmaktadır?



7. Yandaki kod blokları çalıştırıldığında sahnede aşağıdaki şekillerden hangisi ortaya çıkar?



8. Klavyeden boşluk tuşuna basıldıktan sonra karakterin sağa dönmesini ve 50 adım gitmesini sağlayan kod blokları aşağıdakilerden hangisidir?



9. Yandaki kod blokları çalıştırıldığında karakterimiz topa ulaşırsa oyunu kazandı diye mesaj yazmaktadır. Bu kod bloklarında eksik olan kod bloğu aşağıdakilerden hangisidir?



10. I. Karakter 1 tıkladığında x:50 y:70 konumuna gidip 100 adım attıktan sonra kuş sesi çıkaran ve programı durduran komut bloklarını doğru bir şekilde çalışması için yandaki kod bloklarının doğru sıralanışı aşağıdakilerden hangisidir?

a) II-IV-III-V-I b) I-II-IV-V-III c) II-I-V-IV-III d) II-V-I-IV-III

11. Yandaki kod grubu çalıştırıldığında ekranda hangi sayı görünür?

a) 3 b) 5 c) 8 d) 11

12.  düğmesinin görevi aşağıdakilerden hangisidir?

- a) Sahneye sürpriz karakter ekler.
- b) Sahneye yıldız karakteri ekler.
- c) Karakterin özelliklerini görüntüler.
- d) Projedeki kod hatalarını listeler.

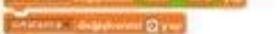
13.  şeklindeki kod bloğunun görevi aşağıdakilerden hangisidir?

- a) Karakterin sahnede resminin çıkmasını sağlar.
- b) Karakteri 180 derece döndürür.
- c) Bu komut içerisine yerleştirilen komutları çalışmayı durdurana kadar tekrarlar.
- d) Karakteri kenarda ise zıpladır.

14. Aşağıdakilerden hangisi Scratch ara yüzüne ait değildir?

- a) Kod Blokları Bölümü
- b) Kodlama Alanı
- c) Sahne ve Karakterler
- d) Durum Çubuğu

15. Klavyeden girilen iki notun ortalamasını alan ve ortalama değişkenine aktaran kod bloğu aşağıdakilerden hangisidir?

- a) 
- b) 
- c) 
- d) 

16.  Yandaki kod blokları çalıştırıldığında ekranda kaç kez "Merhaba!" yazısı görünür?

- a) 2 b) 3 c) 5 d) 8

17.  Yandaki kod blokları çalıştırıldığında aşağıdakilerden hangisi yanlış olacaktır?

- a) Yeşil bayrak tıklanıldığında çalışacaktır.
- b) Kod bloğu sürekli çalışmaktadır.
- c) Eğer beyaz rengi algılasa x:5 y:-110 konumuna gider.
- d) İstenen konuma gitmesi 1 saniyede gerçekleşir.

18. Scratch programında tüm programın ve hareketlerin görüldüğü beyaz alana ne ad verilir?

- a) Kare b) Sahne
- c) Program Penceresi d) Kostüm

19.



Yandaki kod blokları çalıştırıldığında aşağıdakilerden hangisi yanlış olacaktır?

- a) sayı1 ve sayı2 değişkenleri birbirine eşitse bütün blokların çalışması durur.
- b) sayı1 ve sayı2 değişkenleri birbirine eşit değilse bütün blokların çalışması durur.
- c) sayı1 ve sayı2 değişkenleri birbirine eşit değilse ekranda "Yanlış cevap" yazısı görünür.
- d) sayı1 ve sayı2 değişkenleri birbirine eşit değilse "Gong" sesi çalar.

20. Koordinat sisteminin yatay olan eksenine ne ad verilir?

- a) x eksen b) y eksen c) z eksen d) t eksen



21. Yandaki numaralandırılmış düğmelerin sırasıyla görevleri nedir?

- a) 1- hiç bir şekilde dönmez, 2- 360° dönebilir, 3- sağa-sola döner.
- b) 1- 360° dönebilir, 2- sağa-sola dönebilir, 3- hiç bir şekilde dönmez.
- c) 1- 360° dönebilir, 2- hiç bir şekilde dönmez, 3- sağa-sola dönebilir.
- d) 1- sağa-sola dönebilir, 2- hiç bir şekilde dönmez, 3- 360° dönebilir.

22.  Yandaki kod bloğunun görevi aşağıdakilerden hangisidir?

- a) İlk kutudaki değerin ikinci kutudaki değerden küçük durumu kontrolü yapılır.
- b) Belirtilen iki koşulunda doğruluk kontrolünü yapar.
- c) İlk kutudaki değerin ikinci kutudaki değerden büyük durumu kontrolü yapılır.
- d) Belirtilen iki koşuldaki en az birinin doğru olması durumunu arar.

23.



Yandaki kod blokları dükkate alındığında aşağıdaki ifadelerden hangisi yanlıştır?

- a) Top karakterine değmiyorsa 1 saniye bekle.
- b) Komutlar Yeşil bayrak tıklanınca çalışır.
- c) Top karakterine değiyor ise 1 sn bekler ve konuşur.
- d) "Yakala" konuşması 2 saniye sürer.

24. Scratch programında sahnedeki karakterin girilen saniyede x ve y koordinatına gitmesini sağlayan blok aşağıdakilerden hangisidir?

- a)  c) 
- b)  d) 

Ek 2. Belirtke Tablosu

Scratch ile Kodlama Kazanımları	Bilgi	Kavrama
1. Scratch yazılımının arayüzünü tanır.	13, 14, 16, 20, 24	
2. Hareket kod bloklarının görevlerini açıklar.	3, 7	7, 8, 27
3. Kontrol kod bloklarının görevlerini açıklar.	4, 6	8, 9, 10, 18, 19, 22, 26
4. Sahne kavramını açıklar.	21	
5. Sahneye yeni karakter eklemeyi bilir.	2	
6. Karakter kostümü değiştirmeyi bilir.	6	
7. Görünüm kod bloklarının görevlerini açıklar.	6, 10, 12, 22	18, 26
8. Algılama kod bloklarının görevlerini açıklar.	5, 10	19, 26
9. Ses kod bloklarının görevlerini açıklar.		11, 22
10. Operatör kod bloklarının görevlerini açıklar.	25	12, 17, 22
11. Değişken kod bloklarının görevini açıklar.		17, 22
12. Kalem kod bloklarının görevlerini açıklar.	15	8
13. Scratch projesini kayıt etmeyi bilir.	1	
14. Verilen ekran çıktısının hangi kod blokları kullanılarak yapıldığını bilir.		5, 6, 8
15. Scratch programında yönleri tanır.	7	23
16. Tekrarlama ve döngü kavramlarını açıklar.	4, 6, 10	8, 18
17. Verilen kod bloklarının ekran çıktısını bilir.	8	12, 18, 22
18. “Eğer ise” koşul kavramını açıklar.		10, 19, 22, 26
19. Karışık olarak verilen kod blokların doğru olarak sıralar.		11

Ek 3. Scratch Eğitim Videoları İçin İzin Yazışması

Re: Scratch Eğitim Videoları



Ümit ASLAN <umit.aslan@boun.edu.tr>
21.11.2016 (Pzt), 13:51
Siz

Yanıtla

Merhaba Emre,

Elbette videolari tezinde kullanabilirsiniz.

Kolay gelsin.

On Mon, Nov 21, 2016 at 6:52 AM Emre YILDIRIM <emreyildirim@hotmail.com> wrote:
Ümit Hocam Merhabalar,

Ben Çanakkale Onsekiz Mart Üniversitesinde Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümünde yüksek lisans öğrencisiyim.
Scratch için geliştirmiş olduğunuz scratch.eba.gov.tr adresindeki eğitim videolarını izniniz olursa tez çalışmamda kullanabilir miyim?

Scratch Eğitimi
scratch.eba.gov.tr

Merhaba, Scratch Eğitimi'ne hoş geldin. Yukarıdaki menüyü kullanarak konular arasında geçiş yapabilir ya da sağdaki animasyonu seyrederek bu eğitim ...



Ek 4. Araştırma İzni



T.C.
KIRKLARELİ VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : 81588373-605.01-E.594227
Konu : Araştırma İzni

16.01.2017

ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ REKTÖRLÜĞÜNE
(Öğrenci İşleri Daire Başkanlığı)

İlgi : 27.12.2016 tarihli ve 93130991-044-E.146999 sayılı yazınız.

Üniversiteniz Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Yüksek Lisans Programı öğrencisi Emre YILDIRIM'ın "Scratch Programlama Dili Eğitimine Yönelik Bir Mobil Uygulamanın Geliştirilmesi ve Bu Uygulamanın Öğrencilerin Başarı ve Motivasyonuna Etkisinin Değerlendirilmesi" konulu tez çalışması kapsamında Müdürlüğümüze bağlı İstiklal Ortaokulu öğrencilerine anket yapılmasının uygun görüldüğüne dair onay örneği ekte gönderilmiştir.

Bilgilerinizi, ilgiliye tebliğiyle araştırmanın tamamlanmasından itibaren en geç iki hafta içinde araştırmanın iki örneğinin CD'ye kayıtlı olarak Müdürlüğümüze teslim edilmesini arz ederim.

Murat AŞIM
İl Millî Eğitim Müdürü

Ek :

- 1- Onay örneği (1 sayfa)
- 2- Anket Formu (4 sayfa)

Güvenli Elektronik
İmza Aslı ile Aynadır
16/01/2017

Serkan KEÇELİ
Memur

İl Millî Eğitim Müdürlüğü / KIRKLARELİ
Elektronik Ağ: <http://kirkclareli.meb.gov.tr>
e-posta: stratejigelistirme39@meb.gov.tr

Ayrıntılı bilgi için: Serkan KEÇELİ / Memur
Tel: 0 288 214 10 74 (Pbx) / 127
Fax: 0 288 214 11 27

Bu evrak güvenli elektronik imza ile imzalanmıştır. <http://evraksorgu.meb.gov.tr> adresinden 182e-d7cb-3852-8c64-4035 kodu ile teyit edilebilir.

Özgeçmiş

Kişisel Bilgiler

Adı Soyadı : Emre YILDIRIM

Doğum Yeri : Zonguldak / Çaycuma

Doğum Tarihi : 18/10/1986

E-Posta : emreyildirim@hotmail.com

Yabancı Dil : İngilizce

Eğitim Durumu

Yüksek Lisans: Çanakkale Onsekiz Mart Üniversitesi (2017)

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

Lisans : Çanakkale Onsekiz Mart Üniversitesi (2012)

Bilgisayar ve Öğretim Teknolojileri Öğretmenliği

Lise : Silivri Teknik Lisesi (2004)

Bilgisayar Bölümü

İş Deneyimi

Kırklareli Üniversitesi, Uzman, 2014 Eylül - devam ediyor.

Yelkenci Group, Yazılım Uzmanı, 2014 Haziran - Eylül

Çanakkale Onsekiz Mart Üniversitesi, Okutman, 2013-2014

Tiffany- Marka Tekstil A.Ş., Yazılım Uzmanı, 2006 – 2008

Marin Princess Hotel, Bilgi İşlem Sorumlusu, 2005 - 2006