

## BÖLÜM 3 / CHAPTER 3

# EĞİTİMCİLERİN GÖZÜNDEN SCRATCH 3 GÖRSEL PROGRAMLAMA ORTAMININ ARAYÜZ İNCELEMESİ

## AN INVESTIGATION OF THE INTERFACE OF SCRATCH 3 FROM THE PERSPECTIVE OF EDUCATORS

Ali AKKAYA\*, Kaan ARIK\*\*

\*Boğaziçi Üniversitesi, Eğitim Fakültesi, İstanbul, Türkiye

e-posta: ali.akkaya1@boun.edu.tr

\*\*Beykoz Üniversitesi, Sanat ve Tasarım Fakültesi, İstanbul, Türkiye

e-posta: kaanarik@beykoz.edu.tr

DOI: 10.26650/B/ET07.2020.012.03

### ÖZ

Programlama günümüz dünyasının en önemli yeteneklerinden birisidir. Ancak programlama öğrenmeye başlayan bireyler programlama dillerinin yapısı ve programlama geliştirilen ortamların karmaşıklığından dolayı bu derslerde başarısız olmaktadır. Alanyazında yapılan çalışmalara bakıldığında öğrencilerin yaşadıkları problemleri çözmek ve motivasyonlarını artırmak amacıyla öğrenimi kolay ve görselliğin ön planda olduğu programlama ortamları geliştirildiği görülmektedir. Alanyazında yapılan çalışmalarda en popüler, görsel programlama ortamlarından birisi olan Scratch'in öğrencilerin programlama becerileri üzerinde olumlu etkileri olduğu görülmüştür. Ancak Scratch gibi eğitimde yer alan uygulamaların kullanılabilirliği öğrencilerin ve eğitimcilerin akademik performansları açısından önem arz etmektedir. Bu nedenle, bu çalışmada 2019 yılında üçüncü sürümü çıkan Scratch görsel programlama ortamının arayüz incelemesi programlama eğitimi veren eğitmen ve öğretmen adaylarının perspektifinden yapılmıştır. Katılımcılar teker teker kendilerine verilen görevleri Scratch ortamında yerine getirirken katılımcı performanslarının video kaydı alınmış ve daha sonrasında katılımcılardan çevrimiçi bir anketi yanıtlamaları istenmiştir. Araştırmanın sonucunda; katılımcıların Scratch 3'ün tek başına algoritma öğretmek için yeterli olduğu görüşüyle birlikte Scratch'i çoğunlukla ders içi uygulama geliştirme, ödev ve proje geliştirme için kullanmak istedikleri görüşünü benimsedikleri görülmüştür.

**Anahtar Kelimeler:** Algoritma, Görsel Programlama, Scratch, Arayüz Tasarımı

### ABSTRACT

Computer programming is one of the most important skills of today's world. However, students who start to learn computer programming fail in programming courses because of the complex structure of programming

languages and the environments that are used to teach computer programming. In the current literature, it can be seen that researchers developed visual programming environments to help novice programmers overcome their learning problems and increase their motivation for learning computer programming. The research in the current literature points out that Scratch, one of the most popular visual programming environments, has positive effects on students' programming skills. However, the ease of use of such visual programming environments plays a crucial role in the academic performance of both students and instructors. Therefore, in this study an investigation of the interface of the third version of Scratch programmig environment was made from the perspective of educators and computer science teacher candidates. Participants were asked to perform the tasks assigned to them one by one, and while they were doing the tasks a video recording of the participants performance was taken. After the participants completed the tasks, they were asked to answer an online questionnaire. The results of the study indicated that computer science educators and teacher candidates think that Scratch 3 can be solely used to teach algorithmic thinking, and it can also be used in-class application development, homework and project development.

**Keywords:** Algorithm, Visual Programming, Scratch, Interface Design

## GİRİŞ

Programlama, endüstri 4.0 devrimiyle birlikte günümüz dünyasının en popüler yeteneklerinden birisi haline gelmiştir. Özellikle 21. yüzyılın en temel becerileri olarak tanımlanan yaratıcılık, mantıksal düşünme, problem çözme, alternatif düşünceler üretebilme ve üretim gibi yetkinliklerin programlama eğitimiyle geliştirilebileceği vurgulanmaktadır (Pinto ve Escudeiro, 2014). Bu nedenle son dönemde dünya çapında öğrencilerin programlama becerilerini erken yaşlarda edinmesinin gerekliliği ve önemi araştırmacılar tarafından dile getirilmektedir. Ancak programlama öğrenmeye yeni başlayan bireyler yaşadıkları çeşitli problemler sebebiyle programlama öğrenmeye karşı motivasyon kaybına uğramakta ve bu durum programlama derslerine olan talebin de azalmasına neden olmaktadır (Ali ve Shubra, 2010). Literatüre baktığımızda geliştirilen görsel programlama ortamlarının, öğrencilerin programlama öğrenmeye yönelik ilgisini artıran ve görselliği ön plana çıkararak programlama öğrenmeyi kolay hale getirmeyi hedefleyen araçlar olduğunu görmekteyiz. Eğitim ve öğretim süreçlerinde yer alan bu tarz uygulamaların kullanılabilirliğine dayalı olarak ortaya çıkan uygulama problemleri hem öğrenci başarısını olumsuz yönde etkileyebilir hem de eğitimcilerin değerlendirme yaparken güçlüklerle karşılaşmasına sebep olabilir (Crowther, Keller ve Waddoups, 2004). Bu nedenle bu çalışma kapsamında Scratch 3 görsel programlama ortamının arayüz incelemesi programlama eğitimi veren eğitimciler ve öğretmen adaylarının bakış açısından ele alınacaktır.

## PROBLEM

Programlama, bilişim çağının en temel yeteneklerinden birisi olmasına rağmen programlama öğrenmeye yeni başlayan öğrenciler çeşitli problemler sebebiyle bu eğitimlerinde başarısız

olup yeteneklerini kısıtlı ölçüde geliştirebilmektedirler. Öğrencilerin programlama öğrenirken göstermiş oldukları zayıf akademik performans motivasyonlarının düşmesine yol açmakta ve programlama derslerine olan talebin azalmasına sebep olmaktadır (Ali ve Shubra, 2010). Araştırmacılar, öğrencilerin programlama öğrenirken yaşadıkları bu problemlerin çözümü için çeşitli çalışmalar gerçekleştirmiştir.

Alanyazına bakıldığında programlama eğitiminde yaşanan en temel problemin programlama yapabilmek için gerekli olan düşünce yapısının oluşturulamaması olduğu görülmektedir (Kinnunen ve Malmi, 2008; Lahtinen, Ala-Mutka ve Jarvinen, 2005; Özmen ve Altun, 2014; Pea ve Kurland, 1987). Bu temel problemin yanı sıra programlama dillerinin karmaşık yapıda olması ve programlama kavramlarının soyutluğu öğrenciler için problem oluşturan etmenlerdir (Guzdial, 2008; Kölling, 1999a; Xinogalos, 2016). Programlama dillerinin yapısından kaynaklanan problemlerin yanı sıra programlama eğitimi sırasında kullanılan ortamlar da öğrenciler için problem oluşturabilmektedir. Araştırmacılar, öğrencilerin programlama öğrendikleri ve yazılım geliştirdikleri ortamların karmaşık yapıda olmasının öğrenciler için problem oluşturabildiğine ve öğrenme sürecini olumsuz etkilediğine vurgu yapmışlardır (Kölling, 1999b). Kölling'e göre programlama öğretilen ortamlar basit ve kullanılabilir bir arayüze sahip olmalıdır. Basit ve kullanılabilir arayüz tasarımı kullanıcıların bilişsel yükünü azaltmasına olanak sağlaması sebebiyle öğrenme sürecinin verimliliğini artırabilir. Bu problemlere ek olarak, öğrencilerin programlama öğrenmeye yönelik olumsuz bir tutuma sahip olmaları (Gomes ve Mendes, 2007; Hongwarittorn ve Krairit, 2010) ve geleneksel metotlarla sıkıcı ve monoton yapıda dersler işlenmesi de programlama öğrenme sürecini olumsuz etkilemektedir (Byrne ve Lyons, 2001; Prensky, 2003; Sarkar, 2006).

Kodlama öğrenmeye başlayan bireylerin yaşamış oldukları bu sorunların giderilmesi ve öğrencilerin programlama öğrenmeye yönelik motivasyonlarını artırmak amacıyla öğrenimi kolay ve görselliğin ön planda olduğu programlama ortamları geliştirilmiştir. Geliştirilen bu görsel programlama ortamlarında özellikle kodlama dillerinin yapısından kaynaklanan problemlerin ortadan kaldırılması için sözde kodlama yapılması yaklaşımı benimsenmiştir (Xinogalos, 2016). Sözde kod yazımı, bilgisayar bilimi alanında bir problemin çözümünü oluştururken mevcut bir programlama dilinin yapısından uzak, gündelik konuşma diline yakın bir dil kullanılmasıdır. Görsel programlama ortamları her ne kadar öğrencilerin öğrenme süreçlerinde olumlu etkiler meydana getirirse de araştırmacılar bu programlama ortamlarının iyi tasarlanmış eğitim metot ve materyalleriyle birlikte kullanılması gerektiğini belirtmiştir (Meerbaum-Salant, Armoni ve Ben-Ari, 2013; Repenning, Webb ve Ioannidou, 2010).

## Scratch Görsel Programlama Ortamı

Scratch, MIT Medya Laboratuvarı bünyesinde faaliyet gösteren Yaşam boyu Okul Öncesi Grubu (Lifelong Kindergarten Group) tarafından geliştirilmiş olan ücretsiz bir görsel programlama ortamıdır. Scratch ile birlikte kullanıcılar animasyonlu hikayeler, mini oyunlar, müzik videoları ve simülasyonlar gibi farklı çeşitlerde projeler oluşturabilmektedirler (Maloney, Resnick, Rusk, Silverman, ve Eastmond, 2010; Meerbaum-Salant, Armoni, ve Ben-Ari, 2013). Scratch, programlama konusunda herhangi bir ön bilgiye ihtiyaç duyulmadan bireylerin programlama yapabilmesine olanak sağlamayı hedeflemektedir. Alanyazına bakıldığında Scratch'in kodlama eğitimindeki etkilerini farklı açılardan inceleyen çok sayıda çalışma gerçekleştirildiği görülmektedir (Kalelioğlu ve Gülbahar, 2014; Maloney et al, 2008; Meerbaum-Salant, Armoni, ve Ben-Ari, 2013; Nam, Kim ve Lee, 2010; Ozoran, Çağıltay ve Topallı, 2012; Shin ve Park, 2014; Wang, Huang ve Hwang, 2014).

Bu çalışma 2019 yılında üçüncü sürümü çıkan Scratch görsel programlama ortamının arayüz incelemesini insan bilgisayar etkileşimi disiplini kapsamında yapmayı hedeflemektedir. Scratch 3'ün arayüz incelemesi programlama eğitimi veren eğitimciler ya da öğretmen adaylarının perspektifinden gerçekleştirilmiştir. Çalışma kapsamında katılımcıların temel işlemleri gerçekleştirme performansları incelenip, Scratch 3 görsel programlama ortamına karşı tutumları ve derslerinde ne şekilde kullanmak istediklerine dair görüşlerinin edinilmesi hedeflenmektedir.

## YÖNTEM

Bu bölümde yapılan çalışmanın yöntem ve veri toplama aşamaları sunulacaktır. Bölüm; (1) araştırma modeli, (2) katılımcılar ve örneklem, (3) kullanılan materyaller ve (4) verilerin toplanması kısımlarından oluşmaktadır.

### Araştırma Modeli

Araştırma deseni olarak nedensel-karşılaştırma yöntemi tercih edilmiştir. Çalışmada örneklem grubundan alınan anket verileri baz alınarak iki farklı grup oluşturulmuş ve bu grupların belirli bir konudaki eğilimleri betimlenerek birbirlerine göre farklılıklarının tespit edilmesi amaçlanmaktadır.

### Katılımcılar ve Örneklem

Hedef Türkiye'de temel programlama eğitimi veren ya da verecek olan öğretmen ve eğitimcilerdir. Çalışmanın kısıtlı kaynak ve zamanda gerçekleştirilmesi nedeniyle geçerliliği daha yüksek olan rastgele örneklem metodu kullanılamamıştır. Bu nedenle mevcut çalışmada

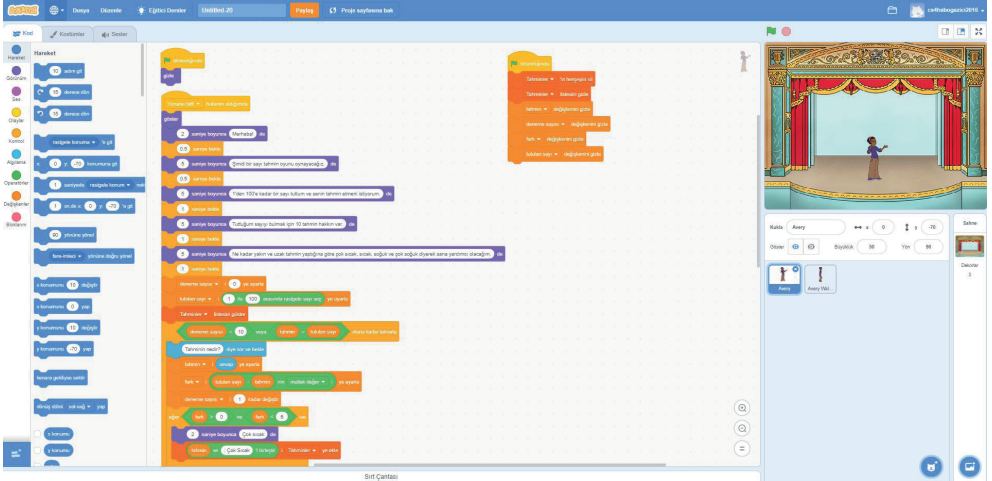
örnekleme tekniği olarak kolaylıkla erişilebilir örnekleme tercih edilmiştir. Katılımcılar seçilirken ön şart olarak aktif bir şekilde programlama eğitimi vermeleri ya da yakın zamanda programlama eğitimi verecek olan aday öğretmen statüsünde olmaları aranmıştır. Çalışmaya, yaşları 22 ile 35 arasında değişen 5 (2 erkek, 3 kadın) bilişim teknolojileri öğretmen adayı ve 2 (1 kadın, 1 erkek) araştırma görevlisi ve 1 (kadın) bilişim teknolojileri dalında öğretmenlik yapan toplam 8 kişi katılmıştır (Tablo 1).

**Tablo 1:** Katılımcıların Cinsiyetlerine Göre Dağılımları

	Frekans (f)	Yüzde(%)
Erkek	3	37,5
Kadın	5	62,5
Toplam	8	100

## Kullanılan Materyal

Çalışmada katılımcılar Scratch 3 görsel programlama ortamının beta sürümünü bir internet tarayıcısı üzerinden kullanmışlardır. Scratch görsel programlama ortamının arayüzünü incelemek için her bir kod grubunu içerecek olan mini bir oyun senaryosu araştırmacılar tarafından tasarlanmıştır (Şekil 1). Senaryo gereği katılımcılardan bir sayı tahmin oyunu kodlamaları istenmiş ve oyunun kullanıcılardan gelen sayı tahminine göre sıcak, soğuk vb. geri bildirimler sunması istenmiştir. Katılımcıların deney sırasındaki aktivitelerinin ekran görüntüsü Articulate Storyline 3 isimli e-öğrenme geliştirme yazılımı kullanılarak alınmıştır.



**Şekil 1:** Scratch 3 Arayüzü

## Verilerin Toplanması

Çalışmanın veri toplama süreci Boğaziçi Üniversitesi Bilgisayar ve Öğretim Teknolojileri Bölümü'nün Bilgisayar Laboratuvarı'nda gerçekleştirilmiştir. Çalışma bilgisayar laboratuvarında veri toplama ve çevrimiçi anket yanıtıma şeklinde iki aşamadan oluşmaktadır. Çalışma öncesinde katılımcılara araştırmanın hedefleri ve çalışma sürecinde kendilerinden yapması beklenen programlama görevi hakkında bilgi verilmiştir. Sonrasında katılımcılar teker teker araştırmacı tarafından kendilerine verilen görevi gerçekleştirmek için Scratch 3 görsel programlama ortamını internet tarayıcısı üzerinde kullanmışlardır. Katılımcıların deney sürecindeki performansları Articulate Storyline 3 ile ekran görüntüsü alınarak kaydedilmiştir. Katılımcılar 45 dakikalık zaman periyodunda kendilerine verilen görev üzerinde çalışmışlardır. Çalışmanın ikinci aşamasında programlama görevini tamamlayan katılımcılara Google Formlar üzerinden çevrimiçi anketler yönlendirilip yanıtları alınmıştır.

## BULGULAR

Bu bölümde, araştırmaya katılan katılımcılardan toplanan performans verilerinin farklı değişkenler bazında ayrı başlıklar altında birbirleriyle olan benzeşmeleri ve farklılıkları gerekli analiz test sonuçları ile birlikte verilecektir. Çalışmada kullanıcıların verilen programlama görevindeki temel işlevleri gerçekleştirdikleri toplam süre ve deneme sayılarıyla görsel programlama ortamını daha önce derslerinde kullanma durumları, Scratch 3'ün geri bildirim mekanizmasını beğenme durumu ve Scratch'in tek başına algoritma öğretmek için yeterli olup olmadığını düşünme durumları kıyaslanacaktır.

Çalışmada yapılan anket ile katılımcıların Scratch programlama ortamına karşı tutumları incelenmiştir. Tablo 2'de katılımcıların Scratch görsel programlama ortamını derslerinde kullanmayı isteme durumuna ait dağılım tablosu verilmiştir. Katılımcıların Scratch görsel programlama ortamını ileriye dönük vereceği derslerde kullanma konusunda iyimser/olumlu görüşe sahip oldukları görülmektedir. Bunun yanı sıra katılımcıların Scratch'i çoğunlukta ders içi uygulama geliştirme, ödevler ve proje/oyun geliştirme ortamı olarak kullanma fikrine sahip oldukları görülmektedir (Tablo 3).

**Tablo 2:** Katılımcıların Verdikleri Derslerde Scratch Kullanmayı İsteme Durumları

Scratch verdiğiniz derslerde kullanmayı düşünüyor musunuz?	Frekans (f)	Yüzde(%)
Evet	6	75
Hayır	1	12,5
Kararsızım	1	12,5
Toplam	8	100

**Tablo 3:** Katılımcıların Scratch 3 Derslerde Ne Şekilde Kullanılabileceğine Dair Görüşleri

	N	Frekans (f)	Yüzde(%)
Ders içi uygulama geliştirme platformu	8	8	100
Ödevler için	8	5	62,5
Proje/oyun geliştirme ortamı olarak	8	6	87,5
Diğer	8	1	12,5

Tablo 4'te araştırmaya katılan bireylerin Scratch programlama ortamını hangi bilgisayar bilimi konularının işlenmesinde kullanılabileceğine dair yanıtlar verilmiştir. Katılımcıların verdikleri yanıtlar ağırlıklı olarak veri tipleri, değişkenler ve sabitler, fonksiyonlar, döngüler, ifade ve eşitsizlikler, problem çözme ve algoritma geliştirme, koşullu ifadeler, görsel tasarım/sahne tasarımı konularında kullanılabileceği yönünde olduğu görülmektedir. Bunun yanı sıra katılımcılar Scratch görsel programlama ortamının diziler ve listeler, nesne ve sınıflar ve hata ayıklama konularının öğretiminde kullanma ihtimallerinin düşük olduğu yönünde görüş bildirmişlerdir.

Yapılan anket sonucunda katılımcılar görsel programlama ortamlarını daha önce derste kullanımlarına, Scratch 3'ün geri bildirim sistemini yeterli bulma durumlarına ve Scratch'in tek başına algoritma öğretmesine yeterli olduğunu düşünme durumuna göre gruplara ayrılmışlardır. Ayrıca katılımcıların kendilerine verilen programlama görevini yerine getirirken gerçekleştirdiği bazı temel işlevlerin toplam süresi ve deneme sayıları kayıt altına alınmıştır. Bu temel işlemler sırasıyla;

- kukla ekleme ve düzenleme,
- dekor ekleme,
- görünüm kod bloğu ekleyerek kostüm değiştirme,
- değişken tanımlama ve değer atama,
- hareket kod bloğu ekleme,
- olay kod bloğu ekleyerek haber salma ve alma,
- kontrol kod bloğu ekleyerek döngü ve koşul yapıları oluşturma
- algılama kod bloğu ekleme
- operatör kod bloğu ekleme şeklindedir.

**Tablo 4:** Katılımcıların Scratch 3’de Hangi Bilgisayar Konularının İşlenmesinde Kullanılabileceğine Dair Görüşler

	N	Frekans (f)	Yüzde(%)
Hata Ayıklama	8	3	37,5
Veri tipleri,değişkenler ve sabitler	8	8	100
Fonksiyonlar	8	5	62,5
Döngüler	8	8	100
İfade ve eşitsizlikler	8	5	62,5
Problem çözüme ve algoritma geliştirme	8	8	100
Koşullu ifadeler	8	8	100
Diziler ve listeler	8	4	50
Nesne ve sınıflar	8	2	25
Görsel tasarım/sahne tasarımı	8	6	25
Diğer	8	1	12,5

### Görsel Programlama Ortamlarının Daha Önce Derslerde Kullanım Durumu ve Çalışmada Verilen Görevi Yerine Getirme Süreleri

**Tablo 5:** Görsel Programlama Ortamını Daha Önce Derslerinde Kullanan ve Kullanmayan Katılımcıların Verilen Görevi Tamamlama Sürelerinin Karşılaştırılması

Levene Testi		<i>T</i>	<i>Sd</i>	<b>P</b> (çift yönlü)	Ortalama Farkı	Standart Sapma Farkı
<b>F</b>	<b>p</b>					
1.065	.342	-2.998	6	.024	-89.00	29.684

Görsel programlama ortamını daha önce derslerinde kullanan ve kullanmayan katılımcıların verilen görevi tamamlama sürelerinin karşılaştırılması Tablo 5’te verilmiştir. Grupların görevi toplam ne kadar sürede tamamladıklarının karşılaştırılması için Bağımsız örneklem t testi uygulanmıştır. Analiz sonucunda görsel programlama ortamını daha önce kullanan eğitimcilerin verilen görevleri tamamlama sürelerinin ortalaması (Ort. 162.5, ss=29.172) görsel programlama ortamını daha önce kullanmayan eğitimcilerin ortalama tamamlama sürelerine (Ort. 251.5,ss=51.708) göre anlamlı düzeyde düşük olduğu görülmektedir ( $t(6) = -2.998, p = .024$ ).



## Görsel Programlama Ortamlarının Daha Önce Derslerde Kullanım Durumu ve Çalışmada Verilen Görevi Yerine Getirirken Toplam Deneme Sayıları

**Tablo 6:** Görsel programlama ortamını daha önce derslerinde kullanan ve kullanmayan katılımcıların görevi kaç denemede yaptıklarının sayısı ile karşılaştırılması

Levene Testi		<i>T</i>	<i>sd</i>	<i>p</i> (çift yönlü)	Ortalama Farkı	Standart Sapma Farkı
<i>F</i>	<i>p</i>					
4.661	.074	-1.910	6	.105	-21.25	11.123

Scratch vb. görsel programlama ortamlarını daha önce derslerinde kullanan ve kullanmayan katılımcıların görevi kaç denemede yaptıklarının sayısı ile karşılaştırılması Tablo 6'da verilmiştir. Grupların görevi kaç denemeye tamamladıklarının karşılaştırılması için Bağımsız örneklem t testi uygulanmıştır. Analiz sonucunda görsel programlama ortamını daha önce kullanan eğitimcilerin verilen görevleri kaç denemede yaptıklarının ortalaması (Ort. 13.75, ss=4.272) ile görsel programlama ortamını daha önce kullanmayan eğitimcilerin deneme sayılarının ortalaması (Ort. 35.00, ss=21.833) arasında istatistiki olarak anlamlı bir fark bulunmamıştır ( $t(6) = -1.910$ ,  $p = .105$ ).

## Scratch 3'ün Geri Bildirim Sistemini Yeterli Bulma Durumu ve Çalışmada Verilen Görevi Yerine Getirme Süreleri

**Tablo 7:** Scratch 3'ün geri bildirim sistemini yeterli bulan ve bulmayan katılımcıların verilen görevi tamamlama sürelerinin karşılaştırılması

Levene Testi		<i>T</i>	<i>Sd</i>	<i>p</i> (çift yönlü)	Ortalama Farkı	Standart Sapma Farkı
<i>F</i>	<i>p</i>					
1.021	.351	-0.975	6	.367	-42.50	43.592

Scratch 3'ün geri bildirim sistemini yeterli bulan ve bulmayan katılımcıların verilen görevi tamamlama sürelerinin karşılaştırılması Tablo 7'de verilmiştir. Grupların görevi toplam ne kadar sürede tamamladıklarının karşılaştırılması için Bağımsız örneklem t testi uygulanmıştır. Analiz sonucunda Scratch 3'ün geri bildirim sistemini yeterli bulan eğitimcilerin verilen görevleri tamamlama sürelerinin ortalaması (Ort. 185.75, ss=47.020) ile geri bildirim sistemini yeterli bulmayan katılımcıların ortalama tamamlama süreleri (Ort. 228.25, ss=73.418) arasında istatistiki olarak anlamlı bir farklılık bulunmamıştır ( $t(6) = -0.975$ ,  $p = .367$ ).

### Scratch 3'ün Geri Bildirim Sistemini Yeterli Bulma Durumu ve Çalışmada Verilen Görevi Yerine Getirirken Toplam Deneme Sayıları

Scratch 3 görsel programlama ortamının geri bildirim sistemini yeterli bulan ve bulmayan katılımcıların verilen görevi kaç denemede yaptıklarına dair verilerin karşılaştırılması Tablo 8'de verilmiştir. Grupların görevi kaç denemeye tamamladıklarının karşılaştırılması için Bağımsız örneklem t testi uygulanmıştır.

Analiz sonucunda Scratch 3'ün geri bildirim sistemini yeterli bulan eğitimcilerin verilen görevleri kaç denemede yaptıklarının ortalaması (Ort. 17.00, ss=7.071) ile geri bildirim sistemini yeterli bulmayan katılımcıların ortalama deneme sayıları (Ort. 31.75, ss=24.514) arasında istatistiki olarak anlamlı bir farklılık bulunmamıştır ( $t(6) = -1.156, p = .292$ ).

**Tablo 8:** Scratch 3'ün geri bildirim sistemini yeterli bulan ve bulmayan katılımcıların verilen kaç denemede yaptıklarının sayısı ile karşılaştırılması

Levene Testi		T	Sd	p (çift yönlü)	Ortalama Farkı	Standart Sapma Farkı
F	P					
3.135	.127	-1.156	6	.292	-14.75	12.757

### Scratch'in Tek Başına Algoritma Öğretiminde Yeterli Olması Durumu ve Çalışmada Verilen Görevi Yerine Getirme Süreleri

**Tablo 9:** Scratch'in tek başına algoritma öğretmek için kullanılmasını yeterli bulan ve bulmayanların verilen görevi tamamlama sürelerinin karşılaştırılması

Levene Testi		T	Sd	p (çift yönlü)	Ortalama Farkı	Standart Sapma Farkı
F	p					
1.134	.328	.627	6	.554	28.50	45.453

Scratch'in tek başına algoritma öğretmek için kullanılabileceğini ve kullanılamayacağını düşünen katılımcıların verilen görevi tamamlama sürelerinin karşılaştırılmasına dair veriler Tablo 9'da sunulmuştur. Grupların görevi toplam ne kadar sürede tamamladıklarının karşılaştırılması için Bağımsız örneklem t testi uygulanmıştır. Analiz sonucunda Scratch'in tek başına algoritma öğretiminde yeterli olduğunu düşünen eğitimcilerin verilen görevleri tamamlama sürelerinin ortalaması (Ort. 221.25, ss=84.740) ile Scratch'in tek başına algoritma öğretiminde yeterli olmadığını düşünen katılımcıların ortalama tamamlama süreleri (Ort. 192.75, ss=32.908) arasında istatistiki olarak anlamlı bir farklılık bulunmamıştır ( $t(6) = -0.627, p = .554$ ).

## Scratch'in Tek Başına Algoritma Öğretiminde Yeterli Olması Durumu ve Çalışmada Verilen Görevi Yerine Getirirken Toplam Deneme Sayıları

Scratch'in tek başına algoritma öğretmek için kullanılabileceğini ve kullanılamayacağını düşünen katılımcıların verilen görevi kaç denemede yaptıklarına dair verilerin karşılaştırılması Tablo 10'da verilmiştir. Grupların görevi kaç denemeye tamamladıklarının karşılaştırılması için Bağımsız örneklem t testi uygulanmıştır.

Analiz sonucunda Scratch'in tek başına algoritma öğretiminde yeterli olduğunu düşünen eğitimcilerin verilen görevleri kaç denemede yaptıklarının ortalaması (Ort. 30.00, ss=25.626) ile Scratch'in tek başına algoritma öğretiminde yeterli olmadığını düşünen katılımcıların ortalama deneme sayıları (Ort. 18.75, ss=7.411) arasında istatistiki olarak anlamlı bir farklılık bulunmamıştır ( $t(6) = 0.843$ ,  $p = .431$ ).

**Tablo 10:** Scratchin tek başına algoritma öğretmek için kullanılmasını yeterli bulan ve bulmayanların verilen görevi tamamlama sürelerinin karşılaştırılması

Levene Testi		T	Sd	p (çift yönlü)	Ortalama Farkı	Standart Sapma Farkı
F	p					
3.214	.123	.843	6	.431	11.25	13.338

## SONUÇ VE TARTIŞMA

Bilgisayar programlama günümüz dünyasının en önemli yeteneklerinden birisi haline gelmiştir. Ancak programlama eğitimi sürecinde programlama dillerinin yapısından ve programlama öğretilen ortamların karmaşıklığından dolayı çeşitli problemler yaşanmaktadır. Öğrencilerin yaşamış oldukları problemler programlama öğrenmeye karşı isteklerini olumsuz yönde etkilemektedir (Ali ve Shubra, 2010). Alanyazına bakıldığında araştırmacıların öğrencilerin yaşadıkları bu problemleri ortadan kaldırmak için programlamayı eğlenceli ve basit hale getirecek görsel programlama ortamları geliştirdikleri görülmektedir. Ancak eğitim ve öğretimde rol alan eğitsel uygulama ve platformların kullanışlılığına dayalı olarak da ortaya uygulama problemleri çıkabilmektedir (Crowther, Keller ve Waddoups, 2004). Ortaya çıkan bu uygulama problemleri hem öğrenci başarısını olumsuz yönde etkileyebilir hem de eğitimcilerin değerlendirme sürecinde güçlüklerle karşılaşmasına sebep olabilir. Bu nedenle bu çalışma kapsamında en popüler görsel programlama ortamlarından birisi olan Scratch'in 2019 yılında çıkan üçüncü sürümünün arayüz incelemesi programlama eğitimi veren eğitimciler ve öğretmen adaylarının bakış açısından ele alınmıştır.

Scratch, MIT tarafından geliştirilen en popüler görsel programlama ortamlarından birisidir. Alanyazında yapılan araştırmalar Scratch programlama ortamının öğrencilerin programlama

becerileri ve öğrenme süreçleri üzerindeki olumlu etkisine vurgu yapmaktadır. Araştırmaya katılan katılımcıların Scratch 3 görsel programlama ortamını vereceği derslerde kullanma konusunda iyimser/olumlu bir görüşe sahip oldukları görülmektedir. Ayrıca katılımcılar ağırlıklı olarak Scratch'in veri tipleri, değişkenler ve sabitler, fonksiyonlar, döngüler, ifade ve eşitsizlikler, problem çözme ve algoritma geliştirme, koşullu ifadeler ve görsel tasarım/sahne tasarımı gibi bilgisayar bilimi konularında kullanılabileceğini belirtmişlerdir. Çalışmada yer alan eğitimciler Scratch'in diziler ve listeler, nesne ve sınıflar ve hata ayıklama gibi konuların öğretilmesi konusunda ise yetersiz kalabileceği şeklinde görüş bildirmişlerdir.

Bazı araştırmacılar tarafından görsel programlama ortamlarının özenle hazırlanmış materyal ve metotlarla kullanılması gerektiğine vurgu yapmaktadırlar (Meerbaum-Salant, Armoni ve Ben-Ari, 2013; Repenning, Webb, ve Ioannidou, 2010). Ancak yapılan çalışmada, programlama eğitimi veren eğitimciler ve öğretmen adayları Scratch'in tek başına algoritma öğretilmesi için yeterli olabileceği görüşünü benimsedikleri görülmüştür. Bununla birlikte çalışmada yer alan katılımcıların Scratch 3 programlama ortamını çoğunlukta ders içi uygulama geliştirme, ödevler ve proje/oyun geliştirme ortamı olarak kullanmak istedikleri görülmüştür.

Bu çalışma kapsamında ayrıca katılımcıların verilen programlama görevindeki temel işlevleri gerçekleştirdikleri toplam süre ve deneme sayılarıyla görsel programlama ortamını daha önce derslerinde kullanma durumları, Scratch 3'ün geri bildirim mekanizmasını yeterli bulma durumu ve Scratch'in tek başına algoritma öğretmek için yeterli olup olmadığını düşünme durumları kıyaslanmıştır. Yapılan analizler sonucunda daha önce görsel programlama ortamında çalışan eğitimcilerin verilen görevleri ortalama tamamlama sürelerinin görsel programlama ortamında ilk defa çalışan eğitimcilere göre anlamlı bir şekilde daha az olduğu görülmüştür. Yapılan istatistiksel incelemeler sonucunda katılımcıların verilen görevi gerçekleştirdikleri toplam süre ve deneme sayıları ile Scratch'in geri bildirim mekanizmasını yeterli bulma durumu ve Scratch'in tek başına algoritma öğretmek için yeterli olup olmaması durumları arasında anlamlı bir ilişki bulunamamıştır. Çalışmadaki bu istatistiksel analizlerin anlamlı bir sonuç çıkarılamamasının en temel nedenlerinden birisi kısıtlı sayıda katılımcıyla çalışmanın gerçekleşmesi olabilir. Bu nedenle çalışmanın geçerliliğini artırmak adına daha fazla sayıda katılımcıyla yeni çalışmalar gerçekleştirilebilir. Ayrıca Scratch görsel programlama ortamıyla ilk defa çalışacak olan öğrencilerin sistem ile etkileşimi göz izleme tekniği ile gerçekleştirilebilir.

## **BİLGİ NOTU / ACKNOWLEDGEMENT**

Bu araştırma, İstanbul Üniversitesi Fen Bilimleri Enstitüsü Enformatik Anabilim Dalı Enformatik Doktora Programında 2018-2019 güz yarıyılında Prof. Dr. Sevinç Gülseçen tara-

findan verilmekte olan İnsan-Bilgisayar Etkileşimi dersi kapsamında Ali Akkaya ve Kaan Arık tarafından 15.01.2019 tarihinde hazırlanmıştır. Araştırmanın veri toplama sürecinde Boğaziçi Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü Bilgisayar Laboratuvarı'nın imkan ve makine-techizatlarından yararlanılmıştır.

## KAYNAKLAR / REFERENCES

- Ali, A., & Shubra, C. (2010). Efforts to reverse the trend of enrollment decline in computer science programs. *The Journal of Issues in Informing Science and Information Technology*, 7, 209–225.
- Crowther, M. S., Keller, C. C., & Waddoups, G. L. (2004). Improving the quality and effectiveness of computer-mediated instruction through usability evaluations. *British Journal of Educational Technology*, 35(3), 289–303.
- Gomes, A., ve Mendes, A. J. (2007, September). Learning to program-difficulties and solutions. *Paper presented at the International Conference on Engineering Education–ICEE*, Coimbra, Portugal.
- Guzdial, M. (2008). Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25–27.
- Hongwarittorn, N., & Krairit, D. (2010). *Effects of program visualization (Jeliot3) on students' performance and attitudes towards Java programming*.
- Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming viascratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 33–50, Vilnius University
- Kinnunen, P., & Malmi, L. (2008). CS minors in a CS1 course. In *Proceeding of the Fourth international Workshop on Computing Education Research Sydney, Australia, 06-07 Ekim 2008*.
- Kölling, M. (1999a). The problem of teaching object-oriented programming. *Journal of Object Oriented Programming*, 11(8), 8–15.
- Kölling, M. (1999b). The problem of teaching object-oriented programming, Part II: Environments. *Journal of Object-Oriented Programming*, 11(9), 6–12.
- Lahtinen, E., Ala-Mutka, K., & Jarvinen, H. M. (2005). *A study of the difficulties of novice programmers*. Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education. New York, USA, 26-29 Haziran.
- Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with scratch. In J. D. Dougherty, S. Rodger, S. Fitzgerald, ve M. Guzdial (Eds.), *Proceedings of the 39th SIGCSE technical symposium on Computer science education* (pp. 367-371). Portland: ACM.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B. & Eastmond, E. (2010). *The Scratch Programming Language and Environment*. *ACM Transactions on Computing Education*, 10, 1–15.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239–264.
- NAM, Dongsoo - KİM, Yungsik - LEE, Taewook (2010). The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill.
- Ozoran, D., Çağıltay, N., & Topallı, D. (2012). Using Scratch In Introduction to Programming Course for Engineering Students. In *2nd International Engineering Education Conference (IEEC2012)* (pp. 125-132).
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 9–27.
- Pea, R. D., & Kurland, D. M. (1987). On the cognitive effects of learning computer programming: A critical look. *New Ideas Psychology*, 2(2), 137–168.

- Pinto, A., & Escudeiro, P. (2014). *The use of Scratch for the development of 21st century learning skills in ICT*. In Information Systems and Technologies (CISTI), 9th Iberian Conference.
- Prensky, M. (2003). Digital game-based learning. *ACM Computers in Entertainment*, 1–4.
- Repenning, A., Webb, D., ve Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In G. Lewandowski, S. Wolfman, T. J. Cortina, & E. L. Walker (Eds.), *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 265-269). New York: ACM.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60–67.
- Sarkar, N. I. (2006). Teaching computer networking fundamentals using practical laboratory exercises. *IEEE Transactions on Education*, 49(2), 285–291.
- Shin, S., & 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.
- WANG Hsiu Ying - HUANG Iwen ve HWANG Gwo Jen (2014). Effects of an Integrated Scratch and Project-Based Learning Approach on the Learning Achievements of Gifted Students in Computer Courses. In *Advanced Applied Informatics (IIAIAI)*, 2014 IIAI 3rd International Conference on (pp. 382-387). IEEE.
- Xinogalos, S. (2016). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. *Education and Information Technologies*, 21(3), 559–588.