GÖRSEL PROGRAMLAMA ORTAMLARINDA YAPILAN OYUN GELİŞTİRME ETKİNLİKLERİNİN ETKİLİLİĞİ[1]

EFFECTIVENESS OF GAME DEVELOPMENT ACTIVITIES IN VISUAL PROGRAMMING ENVIRONMENTS

Hasan BALCI[2] Özgen KORKMAZ[3] Recep ÇAKIR[4] Feray UĞUR ERDOĞMUŞ[5]

Başvuru Tarihi: 08.07.2019 Yayına Kabul Tarihi: 15.11.2020 DOI: 10.21764/maeuefd.589111
(Araştırma Makalesi)

Özet: Bu araştırmanın amacı ilkokul öğrencilerinin görsel programlama araçlarıyla yaptıkları oyun geliştirme etkinliklerinin öğrencilerin programlamaya karşı olan tutumlarına, öz-yeterlilik algılarına etkisi ve öğrencilerin kodlamaya dönük tutumlarını ortaya koymaktır. Bu araştırmada yöntem olarak karma araştırma yöntemi kullanılmıştır. Öntest-sontest kontrol gruplu yarı deneysel araştırma deseni ile birlikte nitel araştırma yöntemlerinden durum çalışması yapılmıştır. Çalışma grubu Amasya ilinden bir ilkokuldan kolay örnekleme yöntemi kullanılarak belirlenmiştir. Her iki grupta da "Bilgisayar Programlama Öğrenmeye Dönük Tutum Ölçeği" ve "Scratch Programına İlişkin Öz Yeterlik Algı Ölçeği" öntest ve sontest olarak kullanılmıştır. Ayrıca görüşme formu hazırlanıp öğrencilere uygulanmıştır. Ön test ve son test analizleri için SPSS programı yardımıyla aritmetik ortalama, standart sapma, t testi analizleri yapılmıştır. Yapılan analizler sonucunda görsel programlama ortamlarında yapılan oyun geliştirme etkinliklerinin öğrencilerin programlamaya dönük tutum son testinde anlamlı farklılık bulunmamıştır. Aynı şekilde öğrencilerin programlamaya karşı öz-yeterlilik algıları ön test ve son test analizlerinde anlamlı bir farklılık görülmemiştir. Öğrencilerin programlamaya karşı tutumlarında ve öz yeterliliklerinde istatiksel olarak anlamlı bir fark olmamasına rağmen nitel veriler analiz edildiğinde deney grubu öğrencilerinin görsel programlamaya karşı ilgilerinin olduğu ve motivasyonlarının yüksek olduğu görülmüştür.

Anahtar Sözcükler: *Programlama, eğitsel robotlar, oyun, tutum, öz-yeterlilik*

Abstract: The aim of this study is to reveal the effects of game development activities of primary school students on visual programming tools on students' attitudes towards programming, self-sufficiency perceptions and students' views on coding. In this research, mixed research method was used. Quantitive data were used with the pretest-posttest control group quasi-experimental research design. Qualitative data were used with case study. The study group was determined using the easy sampling method from a primary school in Amasya. "Computer Programming Learning Attitudes Scale" and "Self-sufficiency Perception Scale for Scratch Program" were used as pre-test and post-test for both groups. In addition, an interview form was prepared and applied to the students. As a result of the analysis, there was not a significant difference in the post-test of students' programming towards game programming activities in visual programming environments. Similarly, there was no significant difference between self-sufficiency perceptions of students. Although there was no significant difference, when the qualitative data were analyzed, it was seen that students had an interest in visual programming and their motivation was high.

Keywords*: Programming, educational robots, game, attitude, self-efficacy*

[1] Bu araştırmanın bir kısmı "International Symposium on Contemporary Education and Social Sciences (ISCESS-2018)" adlı sempozyumda sözlü bildiri olarak sunulmuş ve tam metni yayınlanmıştır.

[2] Yüksek Lisans Öğrencisi, Amasya Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, E-Posta: balciihasan13@gmail.com, https://orcid.org/0000-0003-2072-6813

[3]Prof. Dr., Amasya Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, E-Posta: ozgenkorkmaz@gmail.com, https://orcid.org/0000-0003-4359-5692

[4]Doç. Dr., Amasya Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, E-Posta: recepcakir@gmail.com, https://orcid.org/0000-0002-2641-5007

[5]Doktor Öğretim Üyesi, Amasya Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, E-Posta:ferayugur@gmail.com, https://orcid.org/0000-0002-9401-3405

Giriş

Bilişim teknolojilerinin hızla gelişmesi yazılım alanında çalışabilecek, kendini bu alanda geliştirmiş birey ihtiyacını arttırmaktadır. Yetişen bu bireyler dünyada gelişen bilim alanına katkı sağlayarak teknolojik gelişiminde önemli roller alacaklardır. Programlama eğitimi, bilişim teknolojilerinin temelinde yer alan yazılım alanının varlığını ve devamlılığını sağlayan önemli bir eğitim ortamıdır (Kert ve Uğraş, 2009).

Bir bilgisayar fonksiyonel olarak işlem yapabilmek için bazı komutlar dizisine ihtiyaç duymaktadır. Gerçek hayatta yapılan işlemlerin, komutların bilgisayar diline yani 0/1 mantığına çevrilmesi, düzenlenmesi ve çalıştırılması işlemleri kısaca bilgisayar vasıtasıyla kullanıcıların kullanabileceği uygulamalara dönüştürülmesi işlemine programlama denir (Kesici ve Kocabaş, 2007).

Kesici ve Kocabaş (2007), bilgisayar uygulamasının/programının yazılmasının beş temel aşamadan oluştuğunu belirtmişlerdir. Bunlar "problemin tanımlanması", "problem çözme aşamalarının saptanması", "programın geliştirilmesi", "programın yorumlanması ve derlenmesi" ve "programdaki hataların saptanması ve giderilmesi" aşamalarıdır. Bu aşamalar özetle şunlardır (Kesici ve Kocabaş, 2007):

1. Problemin tanımlanması: Bir problemi belirleyebilmek için problem konusunun iyi analiz edilmesi ve konu ile ilgili detaylı bir araştırmanın yapılması gerekmektedir. Problemi ortaya çıkarmak için araştırma sonucunda çıkan bilgilerin irdelenmesi bu aşamada gerçekleştirilir.
2. Problem çözme aşamalarının saptanması: Problemin belirlenmesinden sonra problemin çözümü için izlenecek yollar adım adım yazılarak bir akış şeması hazırlanır. Bu adımlar semboller kullanılarak algoritma şeması oluşturulur. Bu süreçte kullanacak işlem adımları daha önceden var olan bir çözüm yoluna ait olabilir veya özgün bir çözüm yolu olabilir.
3. Programın geliştirilmesi: Belirlenen problemin çözüm yolu belirlendikten sonra kullanılacak olan metin tabanlı veya görsel tabanlı programlama dili kullanılarak yazılması aşamasıdır.
4. Programın yorumlanması ve derlenmesi: Kullanılan programlama aracında kodlama yaptıktan sonra programın bilgisayarın anlayacağı makine diline çevrildiği aşamadır. Bir başka ifadeyle programın makina diline çevrilip son kullanıcının kullanabileceği dosya formatına dönüştürüldüğü süreçtir.
5. Programdaki hataların saptanması ve giderilmesi: Son aşamada yapılan işlem programdaki hataları belirlemek ve bu hataları gidermektir. Program içerisindeki mantıksal ve yazım hataların tespit edilip giderilmesi bu adımda gerçekleşir. Daha sonra yazılım tekrar derlenip son kullanıcın kullanabileceği şekle dönüştürülür.

Demirer ve Sak (2016)'a göre zor ve maharet isteyen programlama eğitimi öğrencilerin üst düzey becerilerine yönelik çalışmalar gerektirmektedir. Genç ve Karakuş (2011)'a göre bu algı

gençlerin kodlamaya bakış açılarının olumsuz yönde etkilenmesine, kodlamanın sıkıcı, zor ve zahmetli bir iş olduğunu düşünmelerine yol açtığını ortaya koyduğunu ifade etmektedirler.

Literatürde pek çok ülkenin programlama öğretiminin ne derece önemli olduğunun farkına vardığı ve günümüzün ihtiyaçlarını da ele alarak bilişim teknolojileri (BT) eğitimlerine bilgisayarların kodlanması ve programlanması gibi yazılım ön planda olduğu içerikler eklendiği görülebilir. Örneğin Özçakmak (2014)'a göre, Güney Kore, Bilim ve Gelecek Planlama Bakanlığı ilkokul düzeyinde öğrenim gören öğrencilere programlama öğretimini derslerinde zorunlu hale getirileceğini, 2017 yılında ilkokul öğrencilerinin, 2018'de ise lise öğrencilerinin belirli aşamada programlama dersi alacaklarını açıklamıştır. Ülkemizde ise anne ve babaların bilgisayar ve internet ortamının çocuklarda bağımlılık yaptığını düşünerek çocukların bilgisayar ve internet ortamından uzak tutmaya çalışmaktadırlar (Özçakmak, 2014). Bunun nedeni çocuklarda bilgisayara ve internete bağlılığın artması olarak gösterilmektedir. Çocukların bilgisayar başında çok fazla vakit geçirmeleri dolayısıyla ailelerin programlama öğretimine karşı ilgilerin düşük olduğunu söyleyebiliriz. Tüm dünyada bu tarz gelişmeler olduğu gibi Türkiye'de de yazılıma yönelik ihtiyaç artmıştır (Demirer ve Sak, 2016). Bunun için Demirer ve Sak (2016)'ın ortaokul 5. Sınıftan itibaren ülkemizde ders olarak verilen Bilişim Teknolojileri dersinin adını "Bilişim Teknolojileri ve Yazılım" şeklinde değiştirilmiş ve programlama eğitimleri ülkemizde küçük yaştaki bireylere verilmeye başlanmıştır.

Çatlak, Tekdal ve Baz (2015)'a göre programlama yapmayı öğrenmek, programlama yapmanın da ötesinde bireylerin problemleri çözebilme, işbirlikçi öğrenme, özgün düşünebilme ve kritik düşünme gibi üst düzey düşünme becerilerini kazanmalarında önemli olduğunu belirtmişlerdir. Buna göre öğrencilerin özelliklede küçük yaştaki öğrencilerin görsel programlama ortamları ve blok temelli programlama ortamlarını etkin bir şekilde kullanmaları öğrencilerin belli düzeydeki psikomotor becerileri kazanmasında etkilidir (Çatlak ve diğerleri, 2015). Alanyazında programlama öğretimi alan öğrencilerin karmaşık matematik problemlerinin öğretiminde, sonuç odaklı problemlerin çözülmesinde, işbirlikçi, sistematik ve yaratıcı düşünme gibi üst düzey düşünme becerilerinin geliştirilmesinde programlama öğretiminin önemli bir rolü ifade edilmektedir (Taylor, Harlow ve Forret, 2010).

Görsel programlama ortamları öğrencilerin verilen işlemleri en hızlı bir biçimde, mantıklı sonuçlara ulaşmalarında, verilen karmaşık işlemleri sonuç odaklı çözmelerinde, çevrelerindeki bireylerle bilgi alışverişi yaparak daha etkin sonuçlara ulaşmalarında ve kendi hayal dünyalarını etkin bir şekilde kullanarak yeni bir ürün ortaya koymalarında etkilidir. Alanyazın incelendiğinde programlama öğretiminin görsel programlama araçlarıyla desteklenerek küçük yaştaki öğrencilere katkısının incelendiği bir araştırmada, kodlama öğretimi alan öğrencilerin özgün ve farklı düşünmede, problem çözme ve yaratıcı düşünebilme gibi becerilerin kodlama eğitimiyle tanışmamış ve bu konu hakkında bilgisi olmayan öğrencilere göre daha yüksek olduğu belirlenmiştir (Clements ve Gullo, 1984). Bunun için öğrencilerin öğrenmesinin kolay olduğu ve görselliğin ön planda olduğu programlama dilleri ortaya çıkmıştır. Baz (2018)'a göre dünya çapında yaygın olarak kullanılan görsel programlama ortamları Tablo 1'deki gibi özetlemiştir.

Tablo 1

Görsel Programlama Ortamları

Scratch	code.org	Kodable	The Foos
Tynker	Box Island	Cargo Bot	Daisy
Blockly	Move the Turtle	Bitsbox	Code Monkey
Code Combat	Lightbot	Grok Learning	Kidsruby
Bomberbot	Touch Develop	Tech Rocket	RoboMind
Mad Learn	Green Foot	Thimbleby Mozilla	App Inventor
AllcanCode	Code Avengers	Treehouse	You++
Tryruby	Dartlang	GamestarMechanic	Hakitzu Elite
Udemy	Coursera	Code Warriors	CodinGame

Alanyazın incelendiğinde metin tabanlı programlama dilleri yerine görselliğin ön planda olduğu blok tabanlı programlama ortamları çocukların anlayabilmesi açısından önemli bir yere sahiptir (Durak, 2017). Günümüzde blok temelli programlamaya uygun ve hemen hemen herkesin erişim fırsatı sağlayabildiği Scratch ve Code.org blok temelli ve görsel tabanlı programlama ortamlarıdır (Yükseltürk ve Altıok, 2016). Blokları algoritmik sırayla üst üste koyarak programlamayı görsel bir şekilde önümüze seren scratch ve code.org ortamı küçük çocuklardan başlayarak yetişkinlere kadar kodlama fırsatı sağlayabilmektedir (Grover & Pea, 2013). Sürükle bırak şeklinde blokları üst üste koyarak küçük yaştaki bireylerin programlamayı sadece öğrenmekle kalmayıp aynı zamanda problem çözme becerilerinin gelişmesinde de faydalıdır (Zhang, Yang, Luan, Yang, & Chua, 2014).

Literatürde görsel programlama ortamlarının öğrencilerin başarılarına etkisini belirlemek amacıyla çok sayıda çalışma yapılmıştır (Gülbahar ve Kalelioğlu, 2014; Nam, Kim ve Lee, 2010; Ozoran, Cagiltay ve Topalli, 2012; Shin ve Park, 2014; Wang, Huang ve Hwang, 2014). Nam ve diğerleri (2010)'nin Scratch programlama dilini öğrenmek için hazırlanan nesne tabanlı malzeme yazılımının öğrencilerin problem çözme becerileri üzerine etkisinin incelendiği bir araştırmanın sonucuna göre bu çalışmada geliştirilen nesne tabanlı eğitim yazılımı kullanarak öğrenen öğrencilerin problem çözme becerilerini geliştirme üzerinde anlamlı bir farklılaşmanın olduğu gözlemlenmiştir. Ozoran, Çağıltay ve Topallı (2012)'nın yapmış olduğu bir çalışma sonucunda metin tabanlı programlama dillerinden C programlama dili ile birlikte mühendislik öğrencilerine verilen Scratch eğitiminin faydalı olabileceği şeklindedir. Shin ve Park (2014), ilköğretim öğrencilerinin Scratch görsel programlama ortamında verilen eğitimin öğrencilerin problem çözme yeteneğine etkisini inceleyen bir çalışma yapmışlardır. Araştırma sonucunda ilköğretim öğrencilerinin scratch görsel programlama ortamlarında aldığı eğitimin problem çözme becerilerine olumlu etkisinin bulunduğunu tespit edilmiştir.

Alanyazın incelediğinde özellikle ilkokul öğrencilerine dönük görsel programlama ortamlarıyla ilgili yeterince araştırmaya rastlanmamıştır. Bu araştırmanın amacı görsel programlama araçlarıyla yapılan oyun geliştirme etkinliklerinin, ilkokul 4. sınıf öğrencilerinin programlamaya

dönük tutumlarına ve öğrencilerin kodlamaya yönelik öz-yeterlilik algılarına etkisini ortaya koymak ve öğrencilerin uygulamaya yönelik görüşlerini betimlemek şeklinde belirlenmiştir.

Araştırma Problemi

Görsel programlama ortamlarında yapılan oyun geliştirme etkinliklerinin ilkokul 4. sınıf öğrencilerinin programlamaya dönük tutumlarına ve kodlamaya yönelik öz-yeterlilik algılarına etkisi ve programlamaya yönelik görüşleri nelerdir?

- Deneysel müdahale öncesinde grupların programlamaya dönük tutumları ve kodlama yeterlilikleri nasıldır?
- Görsel programlama ortamlarında yapılan oyun geliştirme etkinlikleri öğrencilerin programlamaya dönük tutumları üzerinde etkili midir?
- Görsel programlama ortamlarında yapılan oyun geliştirme etkinlikleri öğrencilerin kodlamaya yönelik öz-yeterlilik algıları üzerinde etkili midir?
- Öğrencilerin görsel programlama ortamlarında oyun geliştirme etkinliklerine yönelik düşünceleri nasıldır?

Yöntem

Araştırma Deseni

İç-içe karma araştırma deseni kullanılarak yapılan araştırmanın nicel kısmında öntest-sontest kontrol gruplu yarı deneysel araştırma deseni kullanılmıştır. Nitel kısmından ise kodlama eğitimi bittikten sonra öğrencilerle kodlama eğitimi hakkında görüşme yapılmıştır. Karma yöntemin amacı pek çok durumda bir fikri doğrulamak ya da desteklemek değil, kişinin olayla ilgili anlayışını genişletmektir (Onwuegbuzie ve Leech, 2004). Çoğu araştırma sorusunun cevabının yalnızca nicel veya nitel araştırmayla ulaşmak mümkün olmayabilir. Ancak karma yöntem kullanılarak araştırmayı hem genelleyebilir hem de derinlemesine inceleme fırsatı yakalanmış olur (Johnson ve Onwuegbuzie, 2004). Fraenkel ve Wallen (2006), deneysel araştırmaların iki farklı özelliği nedeniyle diğer yöntemlerden ayrıldığını ifade etmektedirler. Bu özelliklerden birincisi bir olaydaki değişkenin sonuca etkilerinin gözlenebildiği tek yöntem olması, ikinci özellik ise sebep-sonuç ilişkisini test eden en geçerli ve güvenilir yol olduğudur (Özmen, 2015).

Çalışma Grubu

Çalışma grubunu Amasya İli Merkez İlçesinde bir ilkokuldan kolay örnekleme yöntemiyle belirlenen toplamda 48 kişiden oluşmaktadır. Örnekleme yöntemi olarak basit tesadüfi örnekleme kullanılmıştır. Okulda bulunan sekiz şubeden ‘C’ ve ‘E’ şubesi seçilmiştir. Bu iki gruptan rastgele 4/C deney grubu, 4/E ise kontrol grubu olarak saptanmıştır Çalışmanın nitel grubunu ise deney grubu öğrencilerinden 10 kişi oluşturmaktadır. Deney ve kontrol grubunun cinsiyet ve sınıf dağılımı Tablo 2’de özetlenmiştir.

Tablo 2

Öğrencilerin cinsiyete ve sınıfa göre dağılımı

	Sınıf Mevcudu	Kız	Erkek
4/C	25	15	10
4/E	23	12	11
Toplam	48	27	21

Verilerin Toplanması

Araştırmada Korkmaz ve Altun (2014) tarafından geliştirilen "Bilgisayar Programlama Öğrenmeye Dönük Tutum Ölçeği (BPÖTÖ)" ve Altun ve Kasalak'ın (2017) geliştirmiş olduğu "Scratch Programına İlişkin Öz Yeterlik Algı Ölçeği" gerekli izinler alındıktan sonra deney ve kontrol gruplarına öntest ve sontest olarak uygulanmıştır. Nicel veri toplamak için öğretim öncesinde ve sonrasında ilgili ölçekler uygulandı, nitel veri için ise görüşme formu verilmiştir. Araştırma sonucunu daha derinlemesine incelemek amacıyla görüşme formu hazırlanmıştır.

Bilgisayar Programlama Öğrenmeye Dönük Tutum Ölçeği (BPÖTÖ): Bu ölçek, Korkmaz ve Altun (2014) tarafından geliştirilmiştir. 3 faktör ve toplamda 20 maddeden oluşmaktadır. Ölçeğin iç tutarlılık katsayısı 0,866'dır. Birinci faktör "İsteklilik" (1-9) maddelerinden, ikinci faktör "Negatif Tutum" (10-15) maddelerinden, üçüncü faktör ise "Gereklilik" (16-20) maddelerinden oluşmaktadır. Ölçek 5'li likert tipindedir. '1-Kesinlikle Katılmıyorum', '2-Katılmıyorum', '3-Kararsızım', '4-Katılıyorum' ve '5-Kesinlikle Katılıyorum' başlıklarını içermektedir. 'İsteklilik' faktöründen toplamda elde edilebilecek maksimum puan 45 minimum puan 9'dur. 'Negatif Tutum' faktöründen elde edilebilecek maksimum puan 30 minimum puan 6'dır. 'Gereklilik' faktöründen elde dilebilecek maksimum puan 25, minimum puan 5'tir. Ölçekte 'Gereklilik' faktörü altındaki '16, 17, 18, 19, 20.' Maddeler olumsuz maddelerdir.

Blok Temelli Programlamaya İlişkin Öz-Yeterlik Algısı Ölçeği: Altun ve Kasalak (2017) tarafından geliştirilen ölçek, 12 maddeden ve iki bölümden oluşmaktadır. Ölçme aracının birinci bölümü "cinsiyet", "daha önce Scratch'le programlama dersi alma", "daha önce Scratch'le program yazma", "halen Scratch'le programlama dersi alma","halen Scratch'le program yazma" şeklinde soruların yer aldığı demografik bilgi bölümüdür. Ölçek maddelerinin iç tutarlılık kat sayısı 0.893'tür. Ölçekte iki boyut mevcuttur. Birinci boyutu 'Basit Blok Temelli Programlama Görevleri' ikinci boyut ise 'Karmaşık Blok Tabanlı Programlama Görevleri'dir. İkinci bölümde ise blok temelli programlamaya ilişkin sorular yer almaktadır. Ölçek 5'li likert tipindedir ve '1-Hiç Güvenmiyorum', '2- Biraz Güveniyorum', '3-%50/%50', '4- Oldukça Güveniyorum', '5-Tamamen Güveniyorum' belirtecek şekilde puanlanmıştır. Ölçekte toplamda elde edilebilecek maksimum puan 60, minimum puan 12'dir.

Görüşme Formu: Öğrencilerin görsel programlama ortamlarında hazırlanan oyun geliştirme etkinlikleri hakkındaki görüşleri ve düşüncelerini belirlemek amacıyla 15 açık uçlu sorudan oluşan bir görüşme formu hazırlanmıştır. Görüşme formu hazırlanırken 3 uzman görüşünden

yararlanıldı. Görüşme deney grubu öğrencilerinden 10 kişi ile yapılmıştır. Örnekleme yöntemi olarak amaca uygun örnekleme yönteminden aykırı durum örnekleme yöntemi seçilerek uygulama yapılan süreçte derse aktif katılıp başarı gösteren öğrencilerden en başarılı 10 ve en başarısız 10 öğrenci gruplarından 5'er kişi olarak rastgele seçilmiştir. Görüşmenin uygulanması esnasında her bir öğrencinin izniyle bu görüşmeler ses kaydına alınmıştır. Görüşme ortamı olarak okulda belirlenen boş bir sınıf seçilmiştir. Görüşmenin amacı problem durumunun ve nicel veri analizi sonucunda ortaya çıkan sonuçların daha da derinlemesine incelenmesidir. Görüşmede temel olarak programlamanın ne olduğunu, algoritmanın ne olduğunu, görsel programlama ortamlarının ne olduğunu ve code.org ortamlarında oyun geliştirme etkinlikleri hakkında bilgi alınması gibi başlıklar altında sorular yer almaktadır. Ayrıca öğrencilerin gerçekleştirilen uygulamaya dönük memnuniyet ve isteklilik durumlarını belirlemeye dönük sorular yer almaktadır.

Deneysel İşlemler

Deney grubunda 4 hafta boyunca Code.org görsel programlama ara yüzüyle öğretim yapılarak, öğretim sonunda öğrencilerin kendi oyunlarını tasarlamaları sağlanmıştır. Deney grubuna verilen bu eğitimde bilgisayarlar, tabletler, projeksiyon cihazı ve etkileşimli tahta kullanılmıştır. Bu medya araçları yardımıyla sunuş yoluyla eğitim verilmiştir. Kontrol grubu öğrencilerine ise 4 hafta boyunca geleneksel yöntem ile kodlama eğitimi sunulmuştur.

Çalışmanın süresi toplamda 03.11.2018 ile 4.12.2018 tarihleri arasında 5 hafta olarak planlanmıştır. Bu süre zarfında deney grubuna yönelik 2 saatlik code.org görsel programında öğretim yapılırken, aynı hafta kontrol grubuna ise geleneksel yaklaşım ile programlama ve görsel kodlama hakkında eğitimler verilmiştir. Code.org görsel programlama ile öğretim yapılması planlanan öğretimde kullanılan bilgisayarlar, tabletler ve projeksiyon cihazı derse girmeden önce hazır hale getirilmiştir. Kontrol grubunda ise geleneksel olarak "kodlama eğitimi nedir?", "programlama nedir?", "algoritma nedir?", "bilgisayarca düşünme becerileri nasıl kazanılır?", "görsel programlama dilleri nelerdir?" konuları hakkında çalışmalar yapılmıştır.

03-07.11.2018 tarihleri arasında ilk hafta deney ve kontrol gruplarına "Scratch Programına İlişkin Öz Yeterlik Algı Ölçeği" ve "Bilgisayar Programlama Öğrenmeye Dönük Tutum Ölçeği (BPÖTÖ)" ölçekleri ön test olarak uygulanmıştır.

10-14.11.2018 tarihleri arasında ikinci hafta deney grubu öğrencilerine yönelik Code.org görsel programlamada bulunan kodlama görevleri sırasıyla işlenmiştir. Öğrencilere konularla ilgili örnekler verilerek çocuklardan bilgisayarlarında ve tabletlerinde uygulamaları yapmaları istenmiştir. Aynı hafta kontrol grubunda ise "programlama nedir?" ve algoritma mantığının işlenişi hakkında öğretim yapılmıştır.

17-21.11.2018 tarihleri arasında üçüncü hafta Deney grubu öğrencilerine yönelik Code.org görsel programlamada bulunan kodlama görevleri sırasıyla işlenmeye devam edilmiştir. Öğrencilere konularla ilgili örnekler verilerek çocuklardan bilgisayarlarında ve tabletlerinde uygulamaları

yapmaları istenmiştir. Aynı hafta kontrol grubunda ise bir önceki hafta işlenen konular hakkında kısa bir hatırlatma yapılarak yeni konular işlenmeye başlanmıştır. Bu hafta algoritmanın adımlarından bahsedilerek çocukların günlük hayattan algoritmaya örnekler vermeleri istenilmiştir.

24-28.11.2018 tarihleri arasında dördüncü hafta deney grubu öğrencilerine yönelik Code.org görsel programlamada bulunan kodlama görevleri sırasıyla işlenmeye devam edilmiştir. Öğrencilere konularla ilgili örnekler verilerek çocuklardan bilgisayarlarında ve tabletlerinde uygulamaları yapmaları istenmiştir. Aynı hafta kontrol grubu öğrencilerine görsel programlama hakkında bilgiler verilerek Code.org görsel programlama etkinliği resimler yardımıyla öğrencilere gösterilmiştir.

31.11.2018 ile 4.12.2018 arasındaki son hafta ise öğrencilere sontestler ile görüşme formun uygulanmış ve deneysek süreç tamamlanmıştır.

Verilerin Analizi

Öğrencilerin Bilgisayar programlama öğrenmeye dönük tutumları ve programlamaya ilişkin öz-yeterlilik algıları tanımlayıcı istatistikler (aritmetik ortalama, standart sapma vb.) ve çıkarımsal istatistikler (t-testi) kullanılarak değerlendirilmiştir. Bu süreçte öncelikle grupların benzerlik durumları t testi ile araştırılmış, bazı faktörlerde farklılaşmaların olduğu görüldüğünden, deneysel uygulamanın etkililiği araştırılırken sontes-öntest fark puanları kullanılarak öntestlerin etkisi kontrol altına alınmaya çalışılmıştır. Araştırma sonucunda elde edilen verilerin normal dağılım arasında olup olmadığını belirlemek için Ki-Kare, Kolmogorow-Smirnov, Lilliefors ve Shapiro – Wilk normallik testlerine bakılmıştır. Yapılan testler sonucunda verilerin normal değer arasında olduğu görülmüştür.

Araştırma kapsamında toplanan nitel veriler ise İçerik analizi yöntemi ile analiz edilmiştir. İçerik analizi için Nvivo kullanılmıştır.

Bulgular

Nicel Bulgular

Deneysel çalışma öncesinde grupların Bilgisayar programlama öğrenmeye dönük tutumları ve Bilgisayar programlamaya ilişkin öz-yeterlilik algılarına ilişkin bulgular Tablo 3'de özetlenmiştir.

Tablo 3.

Deney ve Kontrol Grubundaki Öğrencilerin Tutum Öntest Puanları Arasındaki Farklılaşma

	Grup	N	X	SS	t	sd	p
İsteklilik	Deney	25	27.08	7.38	-2.27	46	0.02
	Kontrol	23	32.43	8.90			
Negatif Tutum	Deney	25	14.20	4.70	2.18	46	0.03
	Kontrol	23	11.04	5.29			
Gereklilik	Deney	25	18.32	4.39	-1.88	46	0.06
	Kontrol	23	20.78	4.64			
Tutum Toplam	Deney	25	59.60	10.20	-1.61	46	0.11
	Kontrol	23	64.26	9.70			

Tablo 3'e göre öğrencilerin bilgisayar programlama öğrenmeye dönük tutum öntest puanlarında isteklilik faktörü ($t(46)= -2.27$, $p<0.05$) ve negatif tutum ($t(46)= 21.86$ $p<0.05$) faktöründe istatistiksel olarak anlamlı bir farklılık bulunurken gereklilik faktöründe istatistiksel olarak anlamlı bir farklılık bulunmamıştır ($t(46)= -1.61$, $p>0.05$). Grupların programlama öğrenmeye dönük tutumları öntest ortalama puanları incelendiğinde deney grubunun ortalamasının X=59.60 kontrol grubunun ortalamasının ise X=64.26 olduğu belirlenmiştir. Buna göre deney ve kontrol gruplarının deneysel süreç öncesinde programlama öğrenmeye dönük isteklilik ve negatif tutum faktörleri açısından farklı, gereklilik faktörü açısından benzer olduğu söylenebilir. Deney ve kontrol grubundaki öğrencilerin öz-yeterlilik öntest puanları arasındaki farklılaşmaya ilişkin bulgular Tablo 4'de özetlenmiştir.

Tablo 4

Deney ve Kontrol Grubundaki Öğrencilerin Öz-yeterlilik Öntest Puanları Arasındaki Farklılaşma

	Grup	N	X	ss	t	sd	p
Basit blok Temelli Programlama Görevleri	Deney	25	13.28	5.63	0.17	46	0.86
	Kontrol	23	12.95	6.93			
Karmaşık Blok Tabanlı Programlama Görevleri	Deney	25	20.36	7.16	0.74	46	0.46
	Kontrol	23	18.65	8.71			
Öz-yeterlilik Toplam	Deney	25	33.64	12.24	0.50	46	0.61
	Kontrol	23	31.60	15.32			

Tablo 4'e göre öğrencilerin programlamaya ilişkin öz-yeterlilik algıları öntest puanlarının bütün faktörler açısından anlamlı düzeyde farklılaşma olmadığı görülmektedir($t(46)= 0.50$, $p>0.05$). Grupların programlamaya ilişkin öz-yeterlilik algıları öntest ortalama puanları incelendiğinde deney grubunun ortalamasının X=33.64 ve kontrol grubunun ortalamasının ise X=31.60 olduğu belirlenmiştir. Anlamlı farklılaşma puanları incelendiğinde p=0.61 olduğu görülmektedir. Buna göre deney ve kontrol gruplarının deneysel süreç öncesinde programlamaya ilişkin öz-yeterlilik algıları açısından benzer olduğu söylenebilir. Code.org ortamında yapılan oyun geliştirme etkinlikleri öğrencilerin programlamaya dönük tutumları üzerinde etkisine ilişkin bulgular Tablo 5'de özetlenmiştir.

Tablo 5

Deney ve Kontrol Grubundaki Öğrencilerin Tutum Sontest-Öntest Fark Puanları Arasındaki Farklılaşma

	Grup	N	Sontest-Öntest Fark(X)	ss	t	sd	p
İsteklilik	Deney	25	2.16	8.87	1.24	46	0.21
	Kontrol	23	-1.43	11.02			
Negatif Tutum	Deney	25	-2.68	5.45	-1.17	46	0.24
	Kontrol	23	-.69	6.24			
Gereklilik	Deney	25	1.88	6.41	1.12	46	0.26
	Kontrol	23	.00	4.97			
Tutum Toplam	Deney	25	1.36	12.51	1.02	46	0.31
	Kontrol	23	-2.13	10.93			

Tablo 5'e göre öğrencilerin bilgisayar programlama öğrenmeye dönük tutum sontest-öntest fark puanlarının anlamlı düzeyde farklılaşma olmadığı belirlenmiştir ($t(46)= 1.02$, $p>0.05$). Grupların programlama öğrenmeye dönük tutumları sontest ortalama puanları incelendiğinde deney grubunun sontest-öntest farkı X=1.36, kontrol grubunun sontests-öntest farkı ise X=-2.13 olduğu belirlenmiştir. Buna göre deney ve kontrol gruplarının deneysel süreç sonrasında programlama öğrenmeye dönük tutum açısından istatistiksel olarak anlamlı bir farklılaşma olmadığı söylenebilir. Faktörler açısından da bir farklılaşma bulunmamaktadır. Buna göre deney grubundaki öğrencilerin ortalamaları daha yüksek olmakla birlikte, deneysel uygulamanın öğrencilerin programlama öğrenmeye dönük tutum düzeylerine geleneksel yönteme göre daha fazla katkı sağlamadığı saptanmıştır. Deney ve kontrol grubundaki öğrencilerin öz-yeterlilik sontest puanları arasındaki farklılaşmaya ilişkin bulgular Tablo 6'da özetlenmiştir.

Tablo 6

Deney ve Kontrol Grubundaki Öğrencilerin Öz-yeterlilik Sontest-Öntest Fark Puanları Arasındaki Farklılaşma

	Grup	N	Sontest-Öntest Fark(X)	ss	t	sd	p
Basit blok Temelli Programlama Görevleri	Deney	25	1.28	5.87	-0.95	46	0.34
	Kontrol	23	3.00	6.57			
Karmaşık Blok Tabanlı Programlama Görevleri	Deney	25	-1.16	6.35	-1.33	46	0.19
	Kontrol	23	1.52	7.59			
Öz-yeterlilik Toplam	Deney	25	.12	11.34	-1.24	46	0.21
	Kontrol	23	4.52	13.06			

Tablo 6'ya göre öğrencilerin programlamaya ilişkin öz-yeterlilik algıları sontest-öntest fark puanlarının anlamlı düzeyde farklılaşma olmadığı belirlenmiştir ($t(55)= -1.24$, $p>0.05$). Grupların programlamaya ilişkin öz-yeterlilik algıları sontest ortalama puanları incelendiğinde deney grubunun sontest-öntest farkı X=.12 ve kontrol grubunun sontest-öntest farkı ise X=4.52 olduğu belirlenmiştir. Faktörler açısından da bir farklılaşma bulunmamaktadır. Buna göre deneysel

sürecin öğrencilerin programlamaya ilişkin öz-yeterlilik algıları üzerinde etkili olmadığı söylenebilir. Deney grubundaki öğrencilerin programlamaya ilişkin öz-yeterlilik algıları ve programlama öğrenmeye dönük tutumları öntest-sontest ortalama puanları arasındaki farkın anlamlılığı için yapılan t-testine ilişkin bulgular Tablo 7'de özetlenmiştir.

Tablo 7

Deney Grubundaki Öğrencilerin Öntest-Sontest Karşılaştırması

	Grup	**N**	**ss**	**X**	**t**	**sd**	**p**
Basit blok Temelli Programlama Görevleri	Öntest	25	6.93	12.95	-2.18	46	.040
	Sontest	23	5.96	15.95			
Karmaşık Blok Tabanlı Prog. Görevleri	Öntest	25	8.71	18.65	.961	46	.347
	Sontest	23	8.92	20.17			
İsteklilik	Öntest	25	8.90	32.43	.624	46	.539
	Sontest	23	10.08	31.00			
Negatif Tutum	Öntest	25	5.29	11.04	.535	46	.598
	Sontest	23	5.58	10.34			
Gereklilik	Öntest	25	4.64	20.78	.000	46	1.00
	Sontest	23	4.87	20.78			
Öz-Yeterlilik Toplam	Öntest	25	15.32	31.60	-1.66	46	.111
	Sontest	23	14.33	36.13			
Tutum Toplam	Öntest	25	9.70	64.26	.934	46	.360
	Sontest	23	8.88	62.13			

Tablo 7'ye göre deney grubu öğrencilerin programlamaya ilişkin öz-yeterlilik algıları ($t(46)= -1.66$, $p<0.05$) ve programlama öğrenmeye dönük tutum ($t(46)= .934$, $p>0.05$) sontest-öntest karşılaştırılmasında bir yükselme belirlenmiştir. Ancak bu yükselme istatistiksel olarak anlamlı değildir. Fakat basit programlama görevleri faktörüne bakıldığında istatistiksel olarak anlamlı düzeyde bir farklılaşma mevcuttur. Grupların programlamaya ilişkin öz-yeterlilik algıları ve programlama öğrenmeye dönük tutum sontest-öntest ortalama puanları incelendiğinde programlamaya ilişkin öz-yeterlilik algıları çalışma öncesi öntest ortalaması X=31.60 iken çalışma sonrası uygulanan sontest ortalaması X=36.13 olarak yükselmiştir. Programlamaya öğrenmeye dönük tutum algıları öntest ortalaması çalışma öncesi öntest ortalaması X=64.26 iken çalışma sonrası uygulanan sontest ortalaması X=62.13 olarak düşmüştür. Buna göre deneysel sürecin öğrencilerin programlamaya ilişkin öz-yeterlilik algılarına anlamlı düzeyde katkı sağlamadığı söylenebilir. Deneysel sürecin öğrencilerin programlama yönelik tutum öntest-sontest karşılaştırmasının herhangi bir etki yarattığı söylenemez.

Tablo 8

Kontrol Grubundaki Öğrencilerin Öntest-Sontest Karşılaştırması

	Grup	**N**	**ss**	**X**	**t**	**sd**	**p**
Basit blok Temelli Programlama Görevleri	Öntest	25	5.63	13.28	-1.09	46	.286
	Sontest	23	5.64	14.56			
Karmaşık Blok Tabanlı Prog. Görevleri	Öntest	25	7.16	20.36	.913	46	.371
	Sontest	23	7.74	19.20			
İsteklilik	Öntest	25	7.38	27.08	-1.21	46	.236
	Sontest	23	8.39	29.24			
Negatif Tutum	Öntest	25	4.70	14.20	2.45	46	.022
	Sontest	23	5.03	11.52			
Gereklilik	Öntest	25	4.39	18.32	-1.46	46	.156
	Sontest	23	4.86	20.20			
Öz-Yeterlilik Toplam	Öntest	25	12.24	33.64	-.053	46	.958
	Sontest	23	12.48	33.76			
Tutum Toplam	Öntest	25	10.20	59.60	-.543	46	.592
	Sontest	23	8.72	60.96			

Tablo 8'e göre kontrol grubu öğrencilerin programlamaya ilişkin öz-yeterlilik algıları ($t(46)= -0.053$, $p>0.05$) ve programlama öğrenmeye dönük tutum($t(46)= -0.543$, $p>0.05$) sontest-öntest karşılaştırılmasında az da olsa olarak bir yükselme belirlenmiştir. Ancak bu yükselme istatistiksel olarak anlamlı değildir. Negatif tutum faktöründe diğer faktörlere göre istatistiksel olarak biraz daha fazla yükselme olduğu görülmektedir. Grubun programlamaya ilişkin öz-yeterlilik algıları ve programlama öğrenmeye dönük tutum sontest-öntest ortalama puanları incelendiğinde programlamaya ilişkin öz-yeterlilik algıları çalışma öncesi öntest ortalaması X=12.48 iken çalışma sonrası uygulanan sontest ortalaması X=12.48 olarak yükselmiştir. Programlamaya öğrenmeye dönük tutum algıları öntest ortalaması çalışma öncesi öntest ortalaması X=10.20 iken çalışma sonrası uygulanan sontest ortalaması X=8.72 olarak düşmüştür. Buna göre control grubu öğrencilerin programlamaya ilişkin öz-yeterlilik algılarında sontest-öntest karşılaştırmasının istatistiksel olarak anlamlı olmasa da ufak bir etkiye sahip olduğu söylenebilir. Kontrol grubu öğrencilerin programlamaya yönelik tutum öntest-sontest karşılaştırmasının herhangi bir etki yarattığı söylenemez.

Çalışmanın Nitel Boyutu

İçerik analizi yöntemi ile analiz edilen verilere göre öğrencilerin algoritmik düşünme becerisi tanımlarının bir işlemi adım adım yapmak şeklinde olduğu görülmüştür. Örneğin Katılımcı 1'in belirtmiş olduğu gibi "Arıyı 10 adım ileri götür. Sağa dön. Nektarı al ve bal yap" şeklinde tanımlamıştır. Katılımcıların hepsi günlük hayatlarından algoritma örneği verebilmiştir. Örneğin Katılımcı 7 "Okulda dolabıma giderken dolabı açmak için elimi uzatıyorum. Kitapları alıyorum, sırama koyuyorum" şeklinde bir örnek vermiştir. Bu bulgular ışığında katılımcıların algoritma kavramını doğru olarak anladıkları gözlenmiştir.

Nitel analizler sonucunda ortaya çıkan diğer bir tema ise görsel programlama hakkındaki görüşleridir. Bu tema altında programın değerlendirilmesi, programın eksiklikleri ve oyun yapımı gibi alt temalar ortaya çıkmıştır. Katılımcılar genel olarak programın olumlu ve yararlı olduğu, eğlenerek öğrendiklerini ifade etmektedirler. Örneğin katılımcı 9 "Faydalı ve eğlenceli. Zekamızı açıyor. Ayrıca mantık yürütme becerimizi geliştiriyor ve gelişmemizi sağlıyor" gibi görüş belirtmiştir. Örneğin katılımcı 2'nin oyun yapımı hakkında "Ben oyun tasarlayabilirim ama benim yaptığım oyun paralı olur. Çünkü sonuçta bir emek verdim ben ona. Ama sevdiğim insanlara ücretsiz oynatabilirim" şeklinde yorumlamıştır. Bu bulgular doğrultusunda katılımcıların görsel programlama ortamları hakkında genel olarak olumlu ve pozitif yönde görüş belirttikleri, kendi oyunlarını tasarlayabilecekleri sonucu gözlemlenmiştir.

Nitel analizler sonucunda ortaya çıkan bir başka tema ise oyunlar hakkında bilgileri nelerdir. Bu temanın altında ise ne sıklıkla oyun oynadıkları, oyun oynamayı ne kadar sevdikleri, hangi oyun türlerini oynadıkları gibi alt temalar oluşmuştur. Genel olarak katılımcıların oyun oynamayı sevdikleri, erkek katılımcıların daha çok silah ve araba oyunları oynadıkları kız çocukların ise daha çok barbie ev oyunlarını tercih ettikleri gözlemlenmiştir. Örneğin katılımcı 3 "Okuldan çıktıktan hemen sonra oyun oynamaya başlıyorum. Gece 12'ye kadar oyun oynuyorum" şeklinde ifade etmektedir. Başka bir örnek verecek olursak katılımcı 5 "oyun oynamayı çok severim. Özellikle silah oyunlarını çok severim" diye görüş belirtmiştir. Bu bulgular doğrultusunda katılımcıların oyun oynamayı sevdikleri ve oynadıkları bu oyunlarla vakit geçirmelerinin onlar için bir eğlence olduğu gözlemlenmiştir.

Katılımcıların bildikleri görsel programlama ortamları hakkındaki görüşleri incelendiğinde ortaya bazı katılımcıların sadece code.org görsel programlama ortamını bildiklerini, bazılarının ise code.org, scratch, kodu game lab community ve lego kodlama bildikleri yönünde görüş belirtmişlerdir. Örneğin katılımcı 4 "Scratch ve code.org ile çalıştım. Neredeyse hepsi ile çalıştım" şeklinde görüş belirtmiştir. Katılımcı 7 ise "Sadece code.org ile çalıştım. Diğerlerini hiç görmedim" şeklinde olmuştur. Bu bulgular ışığında bütün katılımcıların code.org hakkında bilgi sahibi olduğu, bazı öğrencilerin daha fazla görsel programlama ortamı hakkında bilgi sahibi oldukları gözlemlenmiştir.

Nitel analizler sonucunda hangi görsel programlama ortamı ile oyun tasarlamak isterdin teması altında neden bu ortamda tasarlardın teması mevcuttur. Genel olarak katılımcıların seçimi lego kodlama olurken bazı katılımcılar code.org ortamını tercih etmiştir. Örneğin katılımcı 10 "code.org ortamında kendi oyunumu yapabilirim" şeklinde görüş belirtmiştir. Nedenini sorduğumuzda sadece bu ortamı öğrendim şeklinde yanıtlar alınmıştır. Ama katılımcı 9'un belirtmiş olduğu gibi "kodu game lab community ile oyun yapmak isterim. Çünkü 3 boyutlu bir ortam sunuyor bize. Daha eğlenceli geliyor" şeklinde görüş belirtmiştir. Bu bulgular doğrultusunda bilmeyen katılımcıların code.org ortamıyla oyun yapmaları şeklindeki görüşlerine karşın başka görsel programlama ortamlarını bilen katılımcıların code.org'u değil diğer programlarda oyun yapmayı tercih ettikleri gözlemlenmiştir.

Tartışma, Sonuç ve Öneriler

Bu çalışmada, code.org ortamlarında yapılan oyun geliştirme etkinliklerinin ilkokul 4. sınıf öğrencilerinin programlamaya ilişkin öz-yeterlilik algılarına etkisi ve öğrencilerin bilgisayar programlama öğrenmeye dönük tutumları incelenmiştir. Programlamaya ilişkin öz-yeterlilik algılarına etkisi ölçeği sontest sonuçlarına göre; deney ve kontrol grupları arasında anlamlı bir farklılaşma görülmemiştir. Analiz sonuçlarına göre, deney grubundaki öğrencilerin ortalamaları daha yüksek olmakla birlikte, deney grubunda code.org görsel programlama ortamı ile eğitim verilirken kontrol grubunda geleneksel yöntem ile eğitim verilmesine rağmen grupların görsel programlamaya yönelik öz-yeterlilik algıları benzerdir. Öğrencilerin bilgisayar programlama öğrenmeye dönük tutum Ölçeği sontest sonuçlarına göre; deney ve kontrol grupları arasında anlamlı bir farklılaşma görülmemiştir. Analiz sonuçlarına göre deney grubunda code.org görsel programlama ortamı ile eğitim verilirken kontrol grubunda geleneksel yöntem ile eğitim verilmesine rağmen grupların görsel programlama öğrenmeye dönük tutumları benzerdir. Öğrencilerin bilgisayar programlamaya ilişkin öz-yeterlilik algıları ve programlamaya yönelik tutumları öntest ve sontest ortalama puanları arasındaki farkın anlamlılığını incelediğimizde, öntest verilerin sontest verilerine göre daha yüksek olduğu belirlenmiştir.

Araştırma sonuçlarına göre görsel programlama ortamlarından code.org ortamında hazırlanan oyun geliştirme etkinliklerinin öğrencilerin Programlamaya ilişkin öz-yeterlilik algılarına etkileri ve bilgisayar programlama öğrenmeye dönük tutumlarında anlamlı düzeyde bir farklılık oluşturmamaktadır. Fakat nitel veriler incelendiğinde ortaya çıkan sonuçlara göre katılımcıların görsel programlama ortamlarına karşı olumlu ve pozitif yönde ilgilerinin olduğu, bu ortamları hem eğlenerek hem de öğrenerek oyun yapma amacıyla kullanabilecekleri ve bu ortamların kendilerini geliştirdiğine dair görüşler belirtmişlerdir. Literatür incelendiğinde programlama eğitiminin ana sınıfından başlayarak ilkokul, ortaokul seviyesinde de devam ettirilerek verilmesiyle ilişkin çalışmalar mevcuttur (Hill, 2015). Alanyazın incelendiğinde küçük yaşta programlama eğitimi verilmesinde rastlanan en sık sorunlar ise uygulama kısmında yaşanan sorunlar olarak ele alınabilir. Öğrencilerin öz-yeterlilik algılarında çalışma bittikten sonra artış olmuştur. Bu süre zarfında öğrencilerin görsel programlama ortamlarına karşı bir farkındalık oluşturduğu gözlemlenmiştir. Genç ve Karakuş (2016)'un da yapmış olduğu "Tasarımla Öğrenme: Eğitsel Bilgisayar Oyunları Tasarımında Scratch Kullanımı" çalışmanın sonuçları incelendiğinde scratch görsel programlama ortamının öğrencilerin tutumlarında pozitif bir artış gözlemlerken metin tabanlı programlama ortamlarında verilen kodlama eğitiminin programlamaya karşı olan ilgilerinde ve tutumlarında scratch görsel programlama ortamından daha az olduğunu ortaya koymuştur. Ayrıca programlama eğitiminin bilişim teknolojileri dersi dışındaki diğer derslerdeki başarıya etki edip başarının artmasına yardımcı olacağı (Yavuz, Gülmez ve Özkaral, 2009) ve kodlama başarısının problem çözme, eleştirel düşünme, mantıksal düşünme ve analitik düşünme gibi üst düzey düşünme becerilerini geliştirdiği saptanmıştır. Code.org ortamında yapılan oyun geliştirme etkinlikleri öğrencilerin algoritmik düşünme becerilerinin ve derse karşı olan ilgilerinin her geçen hafta boyunca bir önceki haftaya göre

olumlu ve pozitif yönde ufak bir artış oluşturmuştur. Yapılan çalışmada öğrencilerin görsel programlama ortamları hakkındaki bilgileri artmış, verilen eğitim öğrencilerin problem çözme becerilerini geliştirmiştir. Erken yaştaki bireylere programlama eğitimine dönük çalışmalarda genellikle nitel araştırma yöntemi kullanılmıştır. Bu durumun nedeni küçük yaştaki çocuklarda bulunan özelliklerin araştırmacıları nitel yöntemi tercih etmelerine neden olmuştur (Durak, Karaoğlan-Yılmaz, Yılmaz-R. ve Seferoğlu, 2017). Ancak bu araştırmada karma araştırma modeli kullanılmıştır. Korkmaz (2016)'a göre scratch eğitiminin öğrencilerin başarılarında pozitif yönde bir etkiye sahip olduğunu ortaya koymuş olduğunu belirten bir araştırma da mevcuttur. Yapılan bu çalışma incelendiğinde öğrencilerin görüşlerinin code.org görsel programlama ortamına karşı olumlu ve pozitif yönde olduğu belirlenmiştir. Ancak deney süresinin kısalığından dolayı öntest ve sontestlerde öğrencilerin programlamaya karşı olan öz-yeterlilikleri ve programlamaya karşı olan tutumlarında ortaya konulamamıştır. Bu durumun nedeni olarak deneysel sürecin kısalığından olduğu düşünülmektedir.

Görsel programlama eğitimi günümüz toplumunda her geçen gün önemini artıran bir uygulama alanı olarak karşımıza çıkmaktadır (Kert ve Uğraş, 2009). 21. Yüzyıl öğrenci profillerine bakıldığında problem çözme becerisinin öğrenciler açısından ne derece önemli olduğu görülmektedir (Trilling ve Fadel, 2009). Bunun için öğrencilerin problem çözme becerilerinin ve yaratıcı düşünme becerilerinin geliştirmelerin gerçek hayattaki problemlerin çözümünde önemli katkı sağlayacağı açıkça belirtilmiştir (Trilling ve Fadel, 2009). Literatür incelendiğinde öğrencilerin problem çözme becerilerinin gelişmesinde programlamanın çok önemli bir anlam taşıdığı belirlenmiştir (Lai ve Yang, 2012; Calder, 2010; Fessakis, Gouli ve Mavroudi, 2012). Ancak ilköğretim düzeyinde yapılan bir çalışmada Gömleksiz ve Bozpolat (2012) sınıf düzeyleri açısından anlamlı bir farklılık bulamamışlardır.

Öğrencilerin görsel programlama ortamlarından code.org ortamında kendilerine ufak oyunlar tasarlama ve bu oyunları daha sonradan oynayabilmeleri ve çevresindeki insanlara oynatabilmeleri hakkındaki görüşleri incelendiğinde, code.org görsel programlama ortamının diğer görsel programlama ortamlarına göre bazı eksiklikleri olduğunu belirtmişlerdir. Ancak kodlama eğitimi almamış bireylerin giriş olarak code.org görsel kodlama programında kendilerine oyun tasarlayarak kodlama eğitimine başlayabileceklerini ifade etmektedirler. Öğrencilerin code.org eğitimini hem eğlenerek hem de severek ve isteyerek derse karşı gösterdikleri olumlu ve pozitif bir bakış açıları olduğu gözlemlenmiştir. Ders bitiminde öğrencilerin bir sonraki hafta tekrardan bu uygulamayı yapmak istemeleri öğrencilerin görsel programlama ortamlarından code.org ortamına karşı göstermiş oldukları pozitif ilgiyi göstermektedir.

Bu alanda yapılan pek çok çalışmanın sonucuna bakıldığında programlama eğitiminin günümüz dünyasında erken yaştaki bireylere verilmesinin çocuklarda; işbirlikçi öğrenme, problem çözme, mantıksal düşünme ve analitik düşünme gibi üst düzey becerileri kazanması açısından oldukça önemlidir. Bu eğitimin faydalarına ilişkin alanyazında pek çok kanıt mevcuttur (Çatlak vd., 2015). Örnek verecek olursak Sanjanaashree, Kumar ve Soman (2014)'ın yapmış olduğu bir

çalışmada blok temelli programlama ortamlarından scratch ile verilen eğitimin öğrencilerin motivasyonlarını ve başarılarını arttırdığı yönde sonuçlara ulaşılmıştır. Öğrencilerin 3 boyutlu materyallerle kendilerinin isteyecekleri bir şeyler tasarlamak istedikleri ve daha sonra bu tasarladıkları materyalleri kodlamak istemeleri bu robotic materyallerin önemini vurgulamaktadır. Yapılan bu materyalleri kodlayarak hareket özelliği kazandırmayı arzulamışlardır. Damar, Durmaz ve Önder (2017)'in yapmış olduğu "Ortaokul Öğrencilerinin FeTeMM Uygulamalarına Yönelik Tutumları ve Bu Uygulamalara İlişkin Görüşleri" çalışmasında öğrencilerin kendilerinin yapmış oldukları robotik materyalleri kodlamak, bu tarz atölye çalışmalarına daha fazla katılmanın öğrencilerin ilgi ve motivasyonlarında pozitif artışın olduğunu vurgulamışlardır. Ayrıca öğrenciler bu tarz etkinliklerin daha fazla olması ve gerekirse ders dışında da böyle etkinliklere devam etmesi konusunda istekli olduklarını belirtmişlerdir (Damar, Durmaz ve Önder, 2017). Buradan yola çıkarak öğrencilerin kodlama eğitimini görselleştirerek ve ilgilerini çekebilecek şekilde görsel programlarla verilmesinin dersi daha iyi kavrayabileceklerini, motivasyonlarının ve derse karşı olan tutumlarının artabileceğini belirtmişlerdir.

Ancak öğrencilerin programlamaya ilişkin öz-yeterlilik algılarına etkileri ve bilgisayar programlama öğrenmeye dönük tutumlarının incelendiği bu çalışmada anlamlı düzeyde bir farklılık görülmemişken öğrencilerin görüşlerinin tam tersi yönde olduğu görülmektedir. Öğrencilerin ilgilerini, motivasyonlarını, eğlenceli bir şekilde öğrenmelerini, dersi sevmeleri ve bu tarz etkinliklerin onlar açısından her alanda faydalı olabileceğini belirtmeleri bu tarz benzer etkinliklerin öğrencilerin gelişimi açısından önemli olduğu belirtebilir. Bu çerçevede deneysel sürecin kısalığından dolayı bu farklılaşmanın anlamlı düzeyde gerçekleşmediği söylenebilir. Bu durumun olası nedenleri aşağıda sıralanmıştır:

Deneysel uygulamanın 4 hafta olması, tutum ve öz-yeterlilik gibi psikometrik özelliklerin değişebilmesi açısından yeterli olmamış olabilir. Deney ve kontrol grubunda bulunan bazı öğrencilerin görsel programlama hakkında ileri derecede bilgiye sahip olmaları, bir başka ifadeyle tutum ve öz-yeterliliklerin deneysel süreç öncesinde zaten yüksek olması, ortalamaların farklılaşmaması neden olmuş olabilir. Ortaya çıkan bu sonuçların daha iyi anlaşılabilmesi için görüşme yapılarak sonuçların daha derinlemesine incelenmesi amaçlanmıştır. Yapılan bu deneysel çalışmada, öğrencilerin tutumlarını ve öz-yeterlilik algılarını ölçebilmek için çalışma süresinin uzatılması etkili olabilir. Ayrıca ilkokul düzeyinde görsel programlama araçlarıyla verilen eğitimler derinlemesine incelenerek nitel araştırma yöntemiyle yapılabilir. İlkokul düzeyinde farklı kademelerdeki öğrenciler üzerinde bu tür çalışmalar yapılabilir. Code.org dışında diğer görsel programlama ortamlarına dönük çalışmalar da yapılabilir.

Kaynakça

Altun, A. ve Kasalak, İ. (2017), Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algısı Ölçeği Geliştirme Çalışması: Scratch Örneği, *Eğitim Teknolojisi Kuram ve Uygulama, 8*(1).

Baz, F. Ç. (2018). Çocuklar İçin Kodlama Yazılımları Üzerine Karşılaştırmalı Bir İnceleme. *Curr Res Educ, 4*(1), 36-47.

Calder, N. (2010). Using Scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9-14.

Clements, D. H., and Gullo, D. F. (1984), Effects of Computer Programming On Young Children's Cognition, *Journal of Educational Psychology*, *76*(6), 1051-1058.

Çatlak, Ş., Tekdal, M., ve Baz, F. Ç. (2015), Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir Doküman İnceleme, *Journal of Instructional Technologies & Teacher Education, 4*(3).

Damar, A, Durmaz, C. ve Önder, İ. (2017). Orta okul Öğrencilerinin FeTeMM Uygulamalarına Yönelik Tutumları ve Bu Uygulamalara İlişkin Görüşleri. *Journal of Multidisciplinary Studies in Education, 1*(1), 47-65.

Davies, P. (2000). Contributions from Qualitative Research. In H. T. Davies, M. N. Sandra, and P. Smith (Eds). *What works? Evidence-based Policy and Practice in Public Services*, 291-316,

Demirer, V. ve Sak, N., (2016). Programming Education and New Approaches Around The World and in Turkey / Dünyada ve Türkiye'de Programlama Eğitimi ve Yeni Yaklaşımlar, *Eğitimde Kuram ve Uygulama*, 12 (3), 521-546.

Durak, G. (2017). The Effects of Scratch Software on Students,' Computational Thinking Skills. *Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi (EFMED)*, 11 (2), 502-517.

Baz, F. Ç. (2018). Çocuklar için kodlama yazılımları üzerine karşılaştırmalı bir inceleme. *Curr Res Educ*, 4(1), 36-47.

Durak, H., Karaoğlan-Yılmaz, G., Yılmaz-R. ve Seferoğlu, S. S. (2017). Erken yaşta programlama eğitimi: Araştırmalardaki güncel eğilimlerle ilgili bir inceleme. H. F. Odabaşı, B. Akkoyunlu ve A. İşman (Ed). *Eğitim teknolojileri okumaları 2017*, 12. Bölüm, 205-236.

Fessakis, G., Gouli, E., and 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.

Fraenkel, J. R., and Wallen, N. E. (2006). How to design and evaluate research in education (6th ed.). New York, NY: McGraw-Hill.

Genç Z. ve Karakuş s. (2011). Tasarımla öğrenme: eğitsel bilgisayar oyunları tasarımında Scratch kullanımı. *5th International Computer & Instructional Technologies Symposium Elazığ*.

Gömleksiz, M. N., ve Bozpolat, E. (2012). İlköğretim 4. ve 5. Sınıf Öğrencilerinin Problem Çözme Becerilerine İlişkin Görüşlerinin Değerlendirilmesi. *Abant İzzet Baysal Üniversitesi Eğitim FAkültesi Dergisi,* 12(2), 23 40.

Grover, S., and Pea, R. (2013). Computational thinking in K-12: *A review of the state of the field, Educational Researcher*, *42*(1), 38–43.

Gürbüz, S. ve Şahin, F. (2014). Sosyal bilimlerde araştırma yöntemleri. Ankara: *Seçkin Yayıncılık.*

Hill, C. (2015). Programming environments for children: *Creating a language that grows with you*.

Johnson, R. B., and Onwuegbuzie, A. J. (2004), "Mixed methods research: A research paradigm whose time has come". *Educational Researcher, 33*(7), 14-26.

Kert, S.B. ve Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. *I. Uluslararası Eğitim Araştırmaları Kongresi,* Çanakkale.

Kesici, T. ve Kocabaş, Z. (2007). *Bilgisayar 2 Ders Kitabı (2. Baskı).* Ankara: Semih Ofset.

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.

Korkmaz, Ö., Altun, H. (2014), A Validity and Reliability Study Of The Attitude Scale Of Computer Programming Learning, *MEVLANA International Journal of Education, 4*(1), 30-43.

Lai, A., and Yang, S. (2011). The learning effect of visualized programming learning on 6th graders' problem solv- ing and logical reasoning abilities, *In: International Conference on Electrical and Control Engineering (ICECE),* 16–18 Sept. 2011, Yichang, 6940–6944.

Gülbahar, Y., Kalelioğlu, F. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education.* 13(1). 33-50.

Nam, D., Kim, Y., Lee, T. (2010). The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill, *Department of Computer Education,* Korea.

Onwuegbuzie, A. J., and Leech, N. L. (2004), "Enhancing the Interpretation of "Significant" Findings: The Role of Mixed Methods Research". *The Qualitative Report, 9*(4), 770-792.

Ozoran, D., Cagiltay, N., and Topalli, D. (2012). Using Scratch In Introduction to Programming Course for Engineering Students. *In 2nd International Engineering Education Conference (IEEC2012),* 125-132.

Özçakmak, Ş. (2014). Bilgisayar kullanımı çocukta bağımlılık yapar mı? 15.11.2018 tarihinde http://www.haberturk.com/polemik/haber/973204-bilgisayar-kullanimi-cocukta-bagimlilik-yapar-mi? adresinden ulaşılmıştır.

Özmen, H. (2015). Kuramdan Uygulamaya Eğitimde Bilimsel Araştırma Yöntemleri, . Metin, M (Ed.). *Araştırma ve temel özellikleri* (ss. 9-11)., Ankara Pegem Akademi.

Sanjanaashree, P., Kumar, M. A., and Soman, K. P. (2014). Language learning for visual and auditory learners using scratch toolkit. *In Computer Communication and Informatics (ICCCI), 2014 International Conference on.*

Shin, S., and 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.

Taylor, M., Harlow, A., and Forret, M. (2010), Using a Computer Programming Environment and An İnteractive Whiteboard To İnvestigate Some Mathematical Thinking, *Procedia-Social and Behavioral Sciences*, 8, 561-570.

Trilling, B., and Fadel, C. (2009). 21ST Century Skills: Learning For Life In Our Times. *San Francisco: Jossey-Bass*.

Wang, H. Y., Huang, I., and Hwang, G. J. (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 (IIAIAAI), 2014 IIAI 3rd International Conference on,* 382-38.

Yavuz, M., Gülmez, D., ve Özkaral, T. C. (2016), Fen Lisesi Öğrencilerinin Akademik Başarıları ile İlgili Deneyimlerinin Değerlendirilmesi, *Kastamonu Eğitim Dergisi,* 24(4), 1655-1672.

Yükseltürk, E. ve Altıok, S. (2016). Bilişim Teknolojileri Öğretmen Adaylarını Programlama Öğretiminde Scratch Aracının Kullanımına İlişkin Algıları, *Mersin Üniversitesi Eğitim Fakültesi Dergisi,* 12(1).

Zhang, H., Yang, Y., Luan, H., Yang, S., and Chua, T.-S. (2014). Start from Scratch, *In Proceedings of the ACM International Conference on Multimedia*, 187–196.

Summary

Introduction

Programming languages provide mathematical thinking and computing thinking skills in individuals. In the literature, people who study programming have an important role in teaching complex math problems, they develop result-oriented problem-solving skills, collaborative, systematic and creative thinking skills. However, there may be significant difficulties in teaching text-based programming languages to individuals in primary and secondary schools. When reviewing the literature, in a study, which examined the contribution of the programming activities in the younger individuals with the support of visual programming tools, it was found that students who had programming education had higher thinking, creativity abilities and metacognition and orientation skills than students who did not have programming education. For this purpose, programming languages such as Alice, Code Gam Lab COMMUNITY, Code.org, Scratch have emerged which are easy to learn and have a visual importance. Numerous studies have been conducted in the literature to determine the effect of these environments on programming teaching. The aim of this study is to determine the effects of game development activities of primary school students with visual programming tools on students 'attitudes towards programming, self-sufficiency perception and students' attitudes towards coding.

Method

Mixed research method was used. Qualitative data were used with the pretest-posttest control group quasi-experimental research design. Qualitative data were used with case study. The study group was determined using the easy sampling method from a primary school in Amasya. In the experimental group, the students were given training with Code.org visual programming interface for 4 weeks to design their own games. The control group students were offered traditional coding and coding training for 4 weeks. “Computer Programming Learning Attitude Scale (BPÖTÖ) “Pre-Test and Post-Test developed by Korkmaz and Altun (2014). were applied to both groups. This scale consists of 3 factors: “Willingness”, “Negative Attitude” and “Requirement”. It consists of 20 items in total. The internal consistency coefficient of the scale is 0.866. Likewise, Altun and Kasalak (2017) developed “Self-sufficiency Perception Scale for Scratch Program” was applied as Pre-Test and Post-Test. The scale consists of 12 items in total and the internal consistency value is 0.893. In addition, an interview form was prepared and applied to the students. As a result of the analysis, there was no significant difference in students' programming towards game programming activities in visual programming environments.

Discussion and Conclusions

There was no significant difference in self-sufficiency perceptions of students in pre-test and post-test analyzes. As a result, there was no statistically significant difference in the attitudes of game development activities in the visual programming environments towards students 'attitudes

towards coding and students' attitudes towards programming. In order to examine this situation in depth, students' their views towards programming were examined. Although there was no significant difference, when the qualitative data were analyzed, it was seen that students had an interest in visual programming and their motivation was high.

Visual programming education is an area of application that increases its importance in today's society every passing day (Kert and Uğraş, 2009). When we look at 21st century student profiles, it is seen how important problem-solving skills for students are (Trilling and Fadel, 2009). For this purpose, it is clearly stated that the improvement of students' problem-solving skills and creative thinking skills will contribute to the solution of real-life problems (Trilling and Fadel, 2009). When the literature is examined, it is determined that programming has a very important meaning in the development of students' problem-solving skills (Lai and Yang, 2012; Calder, 2010; Fessakis, Gouli and Mavroudi, 2012). "Learning by Design: Using Scratch in Educational Computer Games Design" done by Genç and Karakuş (2016). When the results of the study were examined, while the scratch visual programming environment showed a positive increase in students' attitudes, it was found that coding education given in text-based programming environments was less than scratch visual programming environment in their attitudes and attitudes towards programming. Game development activities made in Code.org environment have a positive and positive increase in the students' algorithmic thinking skills and their interest in the course. In this study, students' knowledge about visual programming environments has been increased, education given has developed the problem-solving skills of the students. When this study is examined, it is determined that students' opinions are positive and positive towards code.org visual programming environment. However, due to the short duration of the experiment, pre-test and post-test students' self-efficacy against programming and their attitudes towards programming could not be demonstrated. There is much evidence in the literature about the benefits of this training (Çatlak, Tekdal and Baz, 2015). As an example, in a study by Sanjanaashree, Kumar and Soman (2014), it was found that the education provided by scratch from block-based programming environments increased the students' motivation and success. "Attitudes of Secondary School Students towards STEM Practices and Their Opinions About These Applications" done by Damar, Durmaz and Önder (2017). In this study, it is emphasized that there is a positive increase in the students' interest and motivations in order to encode the robotic materials that students have done themselves and to participate more in this kind of workshops. Based on this, it was stated that visualization of the coding education of the students and giving them with their interest in visual programs could better understand the lesson, their motivation and attitude towards the course could be increased. In this study, the effects of students on self-efficacy perceptions of programming and their attitudes towards learning computer programming are examined. While there was no significant difference, it was seen that the students' opinions were in the opposite direction. Students 'interest, motivation, fun to learn, love the course and such activities can be useful for them in terms of any area of this kind of similar activities may indicate the importance of students' development. In this context, it can be said that this differentiation was not significant due to the shortness of the experimental process.

Etik Beyan: *"Görsel Programlama Ortamlarında Yapılan Oyun Geliştirme Etkinliklerinin Etkililiği" başlıklı çalışmanın yazım sürecinde bilimsel, etik ve alıntı kurallarına uyulmuş; toplanan veriler üzerinde herhangi bir tahrifat yapılmamış, karşılaşılacak tüm etik ihlallerde "Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi Yayın Kurulunun" hiçbir sorumluluğunun olmadığı, tüm sorumluluğun Sorumlu Yazara ait olduğu ve bu çalışmanın herhangi başka bir akademik yayın ortamına değerlendirme için gönderilmemiş olduğunu taahhüt ederim.*