



T.C.

ULUDAĞ ÜNİVERSİTESİ

EĞİTİM BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANA BİLİM DALI

**SCRATCH İLE PROGRAMLAMA ÖĞRETİMİNİN ORTAOKUL 5. VE
6. SINIF ÖĞRENCİLERİNİN PROBLEM ÇÖZME BECERİSİ
ÜZERİNDEKİ ETKİSİNİN İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Özkan VATANSEVER

BURSA

2018



T.C.

ULUDAĞ ÜNİVERSİTESİ

EĞİTİM BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANA BİLİM DALI

**SCRATCH İLE PROGRAMLAMA ÖĞRETİMİNİN ORTAOKUL 5. VE
6. SINIF ÖĞRENCİLERİNİN PROBLEM ÇÖZME BECERİSİ
ÜZERİNDEKİ ETKİSİNİN İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Özkan VATANSEVER

Danışman

Doç. Dr. Şehnaz BALTACI GÖKTALAY

BURSA

2018

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim.

Özkan VATANSEVER

08/02/2018



YÖNERGEYE UYGUNLUK ONAYI

“Scratch İle Programlama Öğretiminin Ortaokul 5. ve 6. sınıf Öğrencilerinin Problem Çözme Becerisi Üzerindeki Etkisinin İncelenmesi” adlı Yüksek Lisans tezi, Uludağ Üniversitesi Eğitim Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlanmıştır.

Tezi Hazırlayan

Özkan VATANSEVER

Danışman

Doç. Dr. Şehnaz BALTACI GÖKTALAY

Bilgisayar ve Öğretim Teknolojileri Eğitimi ABD Başkanı

Prof. Dr. Aysan ŞENTÜRK

T.C.

ULUDAĞ ÜNİVERSİTESİ

EĞİTİM BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı'nda 801290009 numaralı Özkan Vatansever'in hazırladığı "Scratch İle Programlama Öğretiminin Ortaokul 5. ve 6. sınıf Öğrencilerinin Problem Çözme Becerisi Üzerindeki Etkisinin İncelenmesi" konulu Yüksek Lisans çalışması ile ilgili tez savunma sınavı, 08/02/2018 günü 14:30 – 15:30 saatleri arasında yapılmış, sorulan sorulara alınan cevaplar sonunda adayın tezinin **(başarılı / başarısız)** olduğuna **(oy birliği / oy çokluğu)** ile karar verilmiştir.

Üye

Üye

Tez Danışmanı ve Sınav Komisyonu Başkanı

Prof. Dr. Aysan ŞENTÜRK

Doç. Dr. Şehnaz BALTACI GÖKTALAY

Uludağ Üniversitesi

Uludağ Üniversitesi

Üye

Yrd. Doç. Dr. Muzaffer ÖZDEMİR

Çanakkale Onsekiz Mart Üniversitesi

Önsöz

Yüksek lisans öğrenimim süresince bana desteğini esirgemeyen değerli öğretmenim ve danışmanım Doç. Dr. Şehnaz Baltacı Göktalay'a teşekkür ederim. Başta Prof. Dr. Aysan Şentürk olmak üzere tüm Bilgisayar ve Öğretim Teknolojileri Eğitimi ailesine teşekkür ederim.

Bu çalışmamın her aşamasında bana destek olan anne ve babama, lisansüstü eğitimim boyunca bana büyük destek olan ve her konuda yardımcı olan sevgili eşime teşekkür etmeyi bir borç bilirim. Ayrıca çalışmamda bana yardımcı olan Beyhan Çıtak'a teşekkür ederim. Desteklerinden dolayı çalışma grubumu oluşturan öğrencilerime de ayrıca teşekkür ederim.

Özkan Vatansever

Özet

Yazar	: Özkan VATANSEVER
Üniversite	: Uludağ Üniversitesi
Ana Bilim Dalı	: Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı
Bilim Dalı	: Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı
Tezin Niteliği	: Yüksek Lisans Tezi
Sayfa Sayısı	: XV + 95
Mezuniyet Tarihi	: 13/02/2018
Tez	: Scratch İle Programlama Öğretiminin Ortaokul 5. ve 6. sınıf Öğrencilerinin Problem Çözme Becerisi Üzerindeki Etkisinin İncelenmesi
Danışmanı	: Doç. Dr. Şehnaz BALTACI GÖKTALAY

SCRATCH İLE PROGRAMLAMA ÖĞRETİMİNİN ORTAOKUL 5. VE 6. SINIF ÖĞRENCİLERİNİN PROBLEM ÇÖZME BECERİSİ ÜZERİNDEKİ ETKİSİNİN İNCELENMESİ

Pek çok yerel, ulusal ve uluslararası araştırmaların sonuçları ilköğretim ve ortaöğretim öğrencilerinin en başarısız oldukları alanlar içerisinde problem çözme becerisinin olduğunu göstermiştir. Bu araştırmanın amacı, Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisini araştırmak ve bu sürece ilişkin öğrenci görüşlerini değerlendirmektir. Karma yöntem araştırmalarından açıklayıcı desen kullanılan bu araştırmanın nicel aşamasında basit deneysel modellerden tek grup ön test / son test model, nitel aşamasında araştırma deseni olarak durum çalışması kullanılmıştır. Araştırmanın örneklemini Bursa-Osmangazi Şükrü Naili Paşa Ortaokulunda öğrenim gören

109 erkek, 117 kız öğrenci olmak üzere toplam 226 5. ve 6. sınıf öğrencisi oluşturmaktadır. 5. ve 6. sınıf öğrencilerinin problem çözme becerilerini belirlemek amacıyla ön test ve son test olarak yapılandırılmamış problemlerin çözümüne yönelik Ge (2001) tarafından geliştirilen, Coşkun (2004) tarafından Türkçe 'ye çevrilen problem çözme becerisi ölçeği uygulanmıştır. Çalışma sırasında, öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımlarını belirlemek için, 27 öğrenciye ölçekteki problem çözme adımlarına ilişkin görüşme soruları yöneltilmiştir.

Wilcoxon işaretli sıralar testi sonucuna göre, problem çözme becerisi ölçeğinin birinci ve ikinci ölçümlerinin ortalamaları arasında istatistiksel olarak anlamlı bir fark bulunmuştur. Öğrencilerin problem çözme becerileri cinsiyete ve sınıf seviyesine göre farklılık göstermemektedir. Etki büyüklüğü ($d=0,39$) bize Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerinde orta düzeyde bir etkisi olduğunu göstermektedir. Öğrenciler Scratch ile oyun tasarımında ağırlıklı olarak sırasıyla; oyunu zihinlerinde gerçek hayat şartlarına uygun kurgulamaktadırlar, kurguladıkları oyunun tasarımını metne dökmektedirler, tasarıma uygun olarak kod bloklarını yerleştirmektedirler, programı çalıştırıp bir sonraki aşamaya geçmektedirler ve hata durumunda ilgili bölümü tasarıma göre kontrol etmektedirler.

Ortaya çıkan sonuçlara göre; problem çözme becerilerine sahip öğrenciler yetiştirmek için alternatif bir metot olarak Scratch ile programlama öğretiminde oyun tasarımının gerçekleştirilmesi önerilmektedir. Öğrencilerin problem çözme adımlarını incelediğimizde Scratch ortamında yapılandırılmamış problemleri çözebilmeleri ve oyun projesini başarı ile tamamlamaları için tasarım odaklı yaklaşım uygulanması önerilmektedir.

Anahtar sözcükler: kodlama, oyun tasarımı, problem çözme becerisi, programlama öğretimi, Scratch.

Abstract

Author	: Özkan VATANSEVER
University	: Uludag University
Field	: Computer Education and Instructional Technology
Branch	: Computer Education and Instructional Technology
Degree Awarded	: Master
Page Number	: XV + 95
Degree Date	: 13/02/2018
Thesis	: Examining the Effects of Using Scratch Programming on 5th and 6th Graders' Problem Solving Skills
Supervisor	: Assoc. Prof. Şehnaz BALTACI GÖKTALAY

EXAMINING THE EFFECTS OF USING SCRATCH PROGRAMMING ON 5TH AND 6TH GRADERS' PROBLEM SOLVING SKILLS

The results of many local, national and international researches have shown that primary and secondary schools' students are having the most difficulties on problem solving. The aim of this study is to investigate the effect of teaching programming with Scratch on the skills of problem solving of 5th and 6th grade students and to evaluate student views on this process. Mixed method was used for the data analysis with pretest/posttest explanatory design for the quantitative data and case study design for the qualitative data. The sample of the study consisted of 109 male students and 117 female students; that is, totally 226 students from the 5th and 6th grade at Bursa-Osmangazi Şükrü Naili Paşa Secondary School. In order to examine the 5th and 6th grade students' problem solving process, Problem Solving Scale, which was developed by Ge (2001) and translated into Turkish by (Coşkun, 2004), was

applied as pretest and posttest. In order to determine the process steps applied by students on solving the unstructured problems, 27 students were interviewed based on the steps of problem solving scale.

Wilcoxon signed-rank test results showed a statistically meaningful difference between the mean of pretest and posttest scores of problem solving scale. Problem solving skills of the students do not differ according to gender and class level. The effect size ($d=0,39$) showed a medium level effect of Scratch programming on 5th and 6th grade students' problem-solving skills. The students mainly carry out the following procedures about the game design with Scratch respectively: They fictionalize the game in their minds in accordance with real life conditions, write the game design they have fictionalized in mind, place the code blocks according to the design, go to the next stage after running the program and check the relevant part in case of error regarding the design.

According to the results, to raise the students with adequate problem-solving skills, Scratch might be used as an alternative method. It is advised to the teachers to use design-based instruction giving unstructured problems to students and having them solve those using Scratch programming.

Keywords: coding, game design, problem solving skill, Scratch, teaching programming.

İçindekiler

Sayfa No

ÖNSÖZ.....	iv
ÖZET.....	v
ABSTRACT.....	vii
İÇİNDEKİLER.....	ix
TABLolar LİSTESİ.....	xiii
ŞEKİLLER LİSTESİ.....	xiv
KISALTMALAR LİSTESİ.....	xv
1. BÖLÜM: GİRİŞ.....	1
1.1. Problem Durumu.....	1
1.2. Araştırma Soruları.....	5
1.3. Amaç.....	6
1.4. Önem.....	6
1.5. Varsayımlar.....	7
1.6. Sınırlılıklar.....	7
1.7. Tanımlar.....	7
2. BÖLÜM: LİTERATÜR.....	9
2.1. Scratch Programının Özellikleri.....	9
2.2. Scratch Kullanımına Ait Öğrenci Görüşleri.....	12
2.3. Scratch ve Görsel Programlama.....	12

2.4. Scratch ve Problem Çözme Becerisi	14
2.5. Scratch ve Yaratıcı Düşünme Becerisi	16
2.6. Oyun Tasarımı Öğrenmenin Faydaları	17
2.7. Bilgi İşlemsel Düşünme	19
2.8. Problem Çözme Becerisi	20
2.9. Problem Türleri	21
2.10. Yapılandırılmamış Problemlerin Çözüm Süreci	22
2.10.1. Problemin tanımlanması	23
2.10.2. Problemin çözümü	23
2.10.3. Karar verme	24
2.10.4. İzleme ve değerlendirme	24
3. BÖLÜM: YÖNTEM	26
3.1. Araştırmanın Modeli	26
3.2. Çalışma Grubu	29
3.3. Veri Toplama Araçları	30
3.4. Verilerin Toplanması ve Çözümlemesi	32
3.5. Güvenirlik	36
4. BÖLÜM: BULGULAR	44
4.1. Problem Çözme Becerisi	44
4.1.1. Etki büyüklüğü	47
4.2. Yapılandırılmamış Problemlerin Çözüm Süreci	47

4.2.1. Yorumlama ve problemi tanımlama.....	48
4.2.2. Çözümler geliştirme ve çözüm sürecini izleme.	50
4.2.3. Karar verme ve çözüm sürecini değerlendirme.....	51
4.3. Problem Çözme, Programlama ve Özgün Ürün Geliştirme Alanındaki Yeterlikler.....	53
4.3.1. Özel stratejiler.	53
4.3.2. Hayal gücü.....	55
4.3.3. Scratch programının olumlu yönleri.	58
4.3.4. Scratch programının olumsuz ve eksik yönleri.	62
5. BÖLÜM: TARTIŞMA VE ÖNERİLER.....	64
5.1. Problem Çözme Becerisi.....	64
5.2. Yapılandırılmamış Problemlerin Çözüm Süreci.....	65
5.2.1. Yorumlama ve problemi tanımlama.....	65
5.2.2. Çözümler geliştirme ve çözüm sürecini izleme.	66
5.2.3. Karar verme ve çözüm sürecini değerlendirme.....	66
5.3. Problem Çözme, Programlama ve Özgün Ürün Geliştirme Alanındaki Yeterlikler.....	67
5.3.1. Özel stratejiler.	67
5.3.2. Hayal gücü.....	68
5.3.3. Scratch programının olumlu yönleri.	68
5.3.4. Scratch programının olumsuz ve eksik yönleri.	69
KAYNAKÇA.....	70
EKLER.....	77

Ek 1: Problem Çözme Becerisi Ölçeği.....	77
Ek 2: Görüşme Formu	78
Ek 3: İçerik Analizi Kod Tablosu	80
Ek 4: İzin Yazısı.....	83
Ek 5: 5. Sınıf Etkinlikler	84
Ek 6: 6. Sınıf Etkinlikler	89
ÖZGEÇMİŞ	94

Tablolar Listesi

<i>Tablo</i>	<i>Sayfa</i>
1. Yapılandırılmamış problemler için uygulama süreci	24
2. Problem çözme sürecine ilişkin görüşme soruları.....	31
3. İkinci araştırma sorusu için betimsel analiz örnek tablo	33
4. Üçüncü araştırma sorusu için içerik analizi örnek tablo	34
5. Yardımcı araştırmacıya gönderilen Excel tablosunun örneği	37
6. Hayal gücü teması altındaki kodların uyumu.....	38
7. Scratch programının olumlu yönleri teması altındaki kodların uyumu.....	39
8. Scratch programının olumsuz ve eksik yönleri teması altındaki kodların uyumu	40
9. Normallik testi.....	44
10. Wilcoxon işaretli sıralar testi: Farklar puan dizisinin betimsel istatistikleri	45
11. Wilcoxon işaretli sıralar testi anlamlılık test sonucu.....	45
12. Pozitif sıralar ön test ve son test puan ortalamaları betimsel istatistikleri	46
13. Mann-Whitney U testi anlamlılık test sonucu: Cinsiyete göre	46
14. Mann-Whitney U testi anlamlılık test sonucu: Sınıf seviyesine göre	46
15. Yorumlama ve problemi tanımlama.....	48
16. Çözümler geliştirme ve çözüm sürecini izleme	50
17. Karar verme ve çözüm sürecini değerlendirme.....	51
18. Özel stratejiler	53
19. Scratch ile oyun programlamanın faydaları	55
20. Hayal gücü.....	56
21. Scratch programının olumlu yönleri	58
22. Scratch programının olumsuz ve eksik yönleri	62

Şekiller Listesi

<i>Şekil</i>	<i>Sayfa</i>
1. Bilgi işlem üçgeni.....	20
2. Açıklayıcı desene göre gerçekleştirilen araştırma süreci	28



Kısaltmalar Listesi

EBA: Eğitim Bilişim Ağı

MEB: Millî Eğitim Bakanlığı



1. Bölüm

Giriş

Trilling ve Fadel (2009) 21. yüzyılda insanların sahip olması gereken becerileri dijital medya okuryazarlığı, yenilik getirme, yaratıcılık, eleştirel düşünme, problem çözme ve yansıtıcı düşünme becerisi olarak sıralamıştır. Türkiye’de ilkokuldan üniversiteye kadar olan süreçte öğrencilere bu becerilerin ne derece kazandırıldığı ise sürekli güncelliğini koruyan bir tartışma konusudur. Bugünün karmaşık ve hızla gelişen dünyasında her zamankinden daha fazla problem çözme becerilerine sahip genç öğrencilere ihtiyacımız vardır. Scratch ile programlama öğretiminde oyun tasarımı gerçekleştirmek problem çözme becerilerine sahip öğrenciler yetiştirmek için alternatif bir metot olarak düşünülebilir. Oyun tasarımı süreci doğal olarak tasarım ve problem çözmeyi gerektirmektedir. Bu yüzden problem çözme becerilerini geliştirmek için oyun tasarımı kullanmak ideal bir eşleşme olarak kabul edilebilir.

1.1. Problem Durumu

Pek çok yerel, ulusal ve uluslararası araştırmaların sonuçları ilköğretim ve ortaöğretim öğrencilerinin en başarısız oldukları alanlar içinde problem çözme becerisinin olduğunu göstermiştir. Pek çok öğrencinin programlamayı öğrenmeleri için karşılaştığı güçlük, genel problem çözme becerilerinin olmamasıdır (Gomes & Mendes, 2007). Türkiye’de yazılım eğitimi ağırlıklı olarak lisans seviyesinde verilmektedir. Öğrencilerin lisans seviyesine geldiklerinde programlama mantığını kavramakta güçlük çekmemeleri için erken yaşlarda programlama eğitimi verilmesi önem taşımaktadır. Bunun için görsel iki boyutlu bir programlama aracı olan Scratch geliştirilmiştir (Karabak & Güneş, 2013). Scratch programı gençlerde teknolojik akıcılığı iyileştirme niyetindedir. Scratch programında söz dizimi hataları yoktur ve Java diline geçiş kapısı olarak görülmektedir. Genç öğrencilerin

programlamayı öğrenmeleri için ilk dil olarak Scratch programı tavsiye edilmektedir (Malan & Leitner, 2007).

Scratch 'ta oyun programlamak Lego oynamaya benzer. Lego oyunundaki gibi bloklar birleştirilir. Eğer bir amaca yönelik olarak doğru bloklar uygun bir şekilde birleştirilirse, Scratch ile basit veya kısmen profesyonel düzeyde bir program, animasyon veya oyun geliştirmek mümkündür. Scratch, günümüzde programlama eğitiminde en çok kullanılan programlama dillerinden birisidir. Scratch programının ilk 12 yıllık eğitim sürecinde kullanımı hızla artmaktadır ve Harvard Üniversitesi de dahil bazı üniversitelerde programlamaya geçiş aşamasında kullanılmaktadır (Resnick ve diğerleri, 2009).

Ülkemizde, Bilişim Teknolojileri ve Yazılım dersi 5. ve 6. sınıflarda haftada iki ders saati zorunlu, 7. ve 8. sınıflarda ise haftada iki ders saati olmak üzere seçmeli bir ders olarak okutulmaktadır. Dersin Millî Eğitim Bakanlığı tarafından yayımlanan öğretim programında belirlenen öğrenme alanlarının başlıkları şu şekildedir;

1. Bilişim Okur-Yazarlığı,
2. Bilişim Teknolojilerini Kullanarak İletişim Kurma, Bilgi Paylaşma ve Kendini İfade Etme,
3. Araştırma, Bilgiyi Yapılandırma ve İşbirlikli Çalışma,
4. Problem Çözme, Programlama ve Özgün Ürün Geliştirme,
 - Bir problemi çözmek ve projeyi gerçekleştirmek için strateji geliştirebilir, çözüm üretirken farklı bakış açıları ve yaklaşımları kullanabilir.
 - Yazarlık ve programlama dillerini tanıyabilir, en az bir yazarlık/ programlama dilini etkili biçimde kullanabilir.
 - Sistemleri ve konuları incelemek için model, benzeşimler ve canlandırmalar oluşturabilir (Milli Eğitim Bakanlığı [MEB], 2012).

Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanında Scratch programının kullanımının uygun olacağı düşünülmektedir. Millî Eğitim Bakanlığı <http://scratch.eba.gov.tr> sitesinde bir takım eğitim videoları yayınlarak Scratch kullanımını teşvik etmektedir. Scratch ile öğrenciler kendi projelerinin tasarımlarını yaparak gerçek hayatta karşılaştıkları problemlere yaratıcı çözümler uygulayabilirler (Karabak & Güneş, 2013; Lee, 2011).

Bir öğretim aracı olarak Scratch, problem çözme için matematiksel akıl yürütme becerilerini uygulama konusunda oldukça elverişlidir. Bu nedenle Scratch akademik performansı arttırmak için kullanılabilir. Scratch ile işlenen dersler öğrencilerin problem çözme becerileri öğrenmesinde yararlı olabilir (Brown ve diğerleri, 2008).

Ayrıca yapılandırılmamış problemlerin çözümünde işbirliği de önemlidir ve Scratch paylaşım sitesi bu işbirliğini desteklemektedir (Kafai, Fields, Roque, Burke & Monroy-Hernández, 2012; Resnick ve diğerleri, 2009). Öğrenciler Scratch ile oyun, etkileşimli hikâye ve animasyonlar yolu ile problem çözme becerilerini eğlenceli bir şekilde akranlarıyla etkileşime geçerek geliştirirler (Adams, 2010; Calder, 2010; Maloney, Resnick, Rusk, Silverman & Eastmond, 2010). Özellikle oyun tasarımı, programlama kavramlarını kullanırken öğrencilerin bunları öğrenmesi için içsel motivasyonu artırmaktadır. Scratch oyun projelerinde koşullu ifadeler ve döngüler mecburen kullanılmak zorunda olduğundan öğrenciler kompütasyonel kavramları da süreç içerisinde öğrenmektedirler (Adams & Webster, 2012; Claypool, 2013). Scratch programı çok hızlı bir şekilde programlamaya giriş yapabilme imkânı sağlamakta (Maloney ve diğerleri, 2010) ve programın yapısı öğrencilerin sıkılmasını engellemektedir (Adams, 2010).

Jonassen ve Kwon II (2001)'e göre en çağdaş öğrenme ortamlarında dahi öğrenme hedefi problem çözme becerisine sahip birey olmayı gerektirmektedir. Problem temelli öğrenme ortamlarında kullanılacak problem türlerini yapılandırılmış (well-structured) ve

yapılandırılmamış (ill-structured) olarak ikiye ayırmışlardır. Gerçek hayattaki problemlerin çoğu yapılandırılmamış problem iken eğitim hayatımızdaki problemlerin çoğu yapılandırılmış problemlerdir. Yapılandırılmış problemlerin çözüm süreci bellidir, iyi tanımlanmışlardır. Yapılandırılmamış problemlerin ise birden çok çözüm yolu vardır. Öğrenciler eğitim öğretim hayatları süresince yapılandırılmış problemleri sürekli olarak çözmektedirler. Fakat önemli olan yapılandırılmamış problemlere çözüm üretebilme imkânı sağlayacak ortamlar oluşturmaktır. Scratch'ın yaratıcı problem çözme sürecini ve matematiksel düşünmeyi kolaylaştırdığı geri bildirimlerden ortaya çıkmıştır (Brown ve diğerleri, 2008; Calder, 2010).

Scratch ile öğrenciler oyun, animasyon vb. tasarımında karşılarına çıkacak yapılandırılmamış problemleri çözmek için çaba harcayacaklardır. Yapılandırılmamış problem ortamının Scratch ile sağlanabileceği düşünülmektedir. Yapılandırılmamış problemlerin çözümü biraz zor olabilir, çünkü birden fazla disiplindeki bilgilerin kullanılması gerekmektedir. Dolayısıyla, Scratch programı farklı disiplinlerdeki (matematik, bilgisayar bilimi, dil bilimi, sosyal bilgiler gibi) bilgilerin öğrenimini de desteklemektedir ("Scratch", 2018).

Scratch tasarım ortamında yapılandırılmamış problemlerin çözümünde günlük hayatla ilişki kurulacağından, öğrenciler açısından ilgi çekici olacağı düşünülmektedir. Scratch programının problem çözme için ilgi çekici ve nispeten kullanımı kolay ve motive edici bir ortam olduğu yapılan çalışmalarla desteklenmiştir (Brown ve diğerleri, 2008; Calder, 2010; Giannakos, Jaccheri & Proto, 2013; Kalelioğlu & Gülbahar, 2014). Örneğin oyundaki karakterin duvara çarptığında geri gelmesi için, öğrencinin bunun çözümünde matematikteki koordinat bilgisini kullanması ve olayı zihninde canlandırması gerekmektedir. "Günlük hayatta duvara çarparsam ne olur?" şeklinde bir soruyu kendisine sorarak meydana gelecek tepkiyi hayal etmesi gereklidir veya yüksek bir yerden atladığında hangi koordinat eksenindeki değerin zamanla azalacağını ve ne zamana kadar düşmeye devam edeceğini

düşünmesi gerekmektedir. Bu süreçte öğrenci matematiksel ve kompütasyonel kavramları öğrenmektedir (Brown ve diğerleri, 2008; Calder, 2010).

Bu çalışmada, Scratch ile yapılandırılmamış problemlerin çözümü öğrenciler tarafından gerçekleştirilmiştir. Scratch nispeten kolay, ilgi çekici bir ortam sağladığından öğrenciler tasarım ortamında problemi çözme isteği duymuşlardır. Çünkü problemi çözemedikleri takdirde bir sonraki aşamaya geçememektedirler.

1.2. Araştırma Soruları

1. Scratch ile programlama öğretiminin öğrencilerin problem çözme becerileri üzerinde anlamlı bir etkisi var mıdır?
 - H_0 : Öğrencilerin ön test ve son test puan ortalamaları arasında anlamlı bir farklılık yoktur.
 - H_a : Öğrencilerin ön test ve son test puan ortalamaları arasında anlamlı bir farklılık vardır.
 - a. Scratch ile programlama öğretiminde problem çözme becerisi cinsiyete göre farklılık göstermekte midir?
 - H_0 : Kız ve erkek öğrencilerin ön test ve son test puan ortalamaları arasında anlamlı bir farklılık yoktur.
 - H_a : Kız ve erkek öğrencilerin ön test ve son test puan ortalamaları arasında anlamlı bir farklılık vardır.
 - b. Scratch ile programlama öğretiminde problem çözme becerisi sınıf seviyesine göre farklılık göstermekte midir?
 - H_0 : 5. ve 6. sınıf öğrencilerinin ön test ve son test puan ortalamaları arasında anlamlı bir farklılık yoktur.
 - H_a : 5. ve 6. sınıf öğrencilerinin ön test ve son test puan ortalamaları arasında anlamlı bir farklılık vardır.

2. Scratch ile programlama öğretiminde öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımları nelerdir?
 - a. Yorumlama ve problemi tanımlama aşamasında uyguladıkları işlem adımları nelerdir?
 - b. Çözümler geliştirme ve çözüm sürecini izleme aşamasında uyguladıkları işlem adımları nelerdir?
 - c. Karar verme ve çözüm sürecini değerlendirme aşamasında uyguladıkları işlem adımları nelerdir?
3. Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanındaki yeterliklere ilişkin öğrenci görüşleri nelerdir?
 - a. Öğrencilerin problem çözüm sürecinde kullandıkları özel stratejiler nelerdir?
 - b. Scratch programının hayal gücüne katkısına ilişkin öğrenci görüşleri nelerdir?
 - c. Öğrenciler açısından, Scratch programının olumlu yönleri nelerdir?
 - d. Öğrenciler açısından, Scratch programının olumsuz ve eksik yönleri nelerdir?

1.3. Amaç

Araştırmanın amacı, Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisini araştırmak ve bu sürece ilişkin öğrenci görüşlerini değerlendirmektir.

1.4. Önem

2005-2006 eğitim öğretim yılından bu yana uygulanan öğretim programlarında temel becerilerden biri olarak yer alan problem çözme becerisini kazanma düzeyinin belirlenmesi önem taşımaktadır. Millî Eğitim Bakanlığı ders müfredatları yapılandırmacı yaklaşıma göre hazırlanmaktadır. Scratch programı yapısı itibarıyla yapılandırmacı yaklaşıma uygundur (Meerbaum-Salant, Armoni & Ben-Ari, 2011). 21. yüzyıl becerilerinden olan problem çözme becerisinin Scratch ile öğrencilere kazandırılması, yapılandırılmamış problemleri tasarlama ve çözme ortamı sağlaması eğitim açısından önemlidir. Ayrıca 21. yüzyıl becerileri içerisine

kodlama becerisinin de eklenmesi isabetli bir karar olarak görülmektedir. Öğrenciler açısından oyun oynamak ile kendi oyununu tasarlamak açısından öğrenme sürecinde büyük farklılık vardır.

1.5. Varsayımlar

Bu araştırmanın sonuçları aşağıdaki varsayımların kabul edilmesine bağlıdır.

- Araştırmanın katılımcıları görüşme formundaki sorulara ve problem çözme becerisi ölçeğine samimi cevap vermişlerdir.
- Araştırma sürecindeki 11 haftalık Scratch eğitimindeki oyunlar öğrenciler tarafından eksiksiz olarak tamamlanmıştır.

1.6. Sınırlılıklar

Araştırma bulguları ve sonuçları, 2014-2015 eğitim öğretim yılı içerisinde Bursa-Osmangazi Şükrü Naili Paşa Ortaokulu 5. ve 6. sınıfta öğrenim gören 226 öğrenci ile sınırlıdır. Öğrencilerin problem çözme becerileri ölçek sorularına göre belirlenmiştir. Bu kısıtlılığın dezavantajını en aza indirmek için, daha önce geçerlik ve güvenilirlik çalışması yapılmış olan ölçme aracından faydalanılmıştır. Öğrencilerin gruplara seçkisiz olarak atanma imkânı bulunmadığından ve öğretim programındaki kazanımların gerçekleştirilebilmesi için kontrol grubu oluşturulamayacağından dolayı, araştırma gerçek deneysel desenler ile gerçekleştirilememiştir.

1.7. Tanımlar

Scratch: MIT (Massachusetts Institute of Technology) üniversitesi tarafından geliştirilen görsel bir programlama dilidir. Bu dil özellikle 8-16 yaş grubu öğrenciler için oluşturulmuştur. Bu dili kullanarak kendi oyununuzu programlayabilir, kendi animasyonlarınızı oluşturabilir ve yaratıcılığınız dâhilinde birbirinden güzel çalışmalar üretebilirsiniz.

Sürükle Bırak Arayüzü: Metin / yazı olarak kodların yazılmasını gerektirmeyen, tüm nesne ve hazır kodların ilgili alana sürüklenip bırakılması özelliğini içinde barındıran arayüzdür.

Programlama: Bilgisayarın ne yapması gerektiğinin adım adım söylenmesidir. Bilgisayarlar bizim isteklerimiz doğrultusunda çalışan elektronik aletlerdir. İstedğimiz iş ve işlemleri yapabilmeleri için programlanmaları gerekir. Programlama ile biz bilgisayara işleri nasıl yapması gerektiğini anlatırız.



2. Bölüm

Literatür

Araştırma soruları ile ilgili Scratch çalışmalarını aşağıdaki dört ana başlık altında toplayabiliriz. Devamında ise oyun tasarımı öğrenmenin faydaları ile araştırmanın teorik çerçevesi olan bilgi işlemsel düşünme süreci ve problem çözme becerisi ile ilgili literatür bulunmaktadır.

2.1. Scratch Programının Özellikleri

MIT üniversitesi tarafından tasarlanan Scratch, Türkçe dâhil 50 dili desteklemektedir. 150'den fazla farklı ülkede kullanılabilir. Scratch özellikle 8-16 yaş grubu öğrenciler için tasarlanmıştır. Scratch ile etkileşimli hikâyeler, oyunlar, animasyonlar programlanabilir ve bu uygulama ile gerçekleştirilen projeler çevrimiçi topluluk ile paylaşılabilir. Öğrenciler internet sitesine üye olarak projelerini paylaşabilirler, diğer öğrenciler ile iş birliğine giderek projelerini geliştirebilirler ve kod blokları kolaylıkla 50 dile çevrilebilmektedir. Scratch web sitesinde 28.001.201 paylaşılan proje, 23.951.940 üye kullanıcı, 140.768.970 yorum ve 4.072.689 stüdyo bulunmaktadır ("Scratch", 2018).

Görsel programlama dilleri öğrencileri keşif ile öğrenmeye teşvik etmektedir (Rosenbaum, 2009). Scratch, öğrenciyi problemi çözmek için algoritma tasarlamaya başlamayı ve yaratıcılığı teşvik etmeyi, bilgisayar bilimine olan ilgiyi arttırmayı ve kaygıyı azaltmayı amaçlamaktadır. Scratch programında söz dizimi hataları olmadığı için, öğrenci teşvikinde pozitif etki yapmaktadır. Ayrıca Scratch programının renkli ara yüzü animasyonlu bir oyun yaratmak için öğrencileri teşvik ettiğinden, öğretmeni öğretim aşamasında rahatlatmaktadır (Meerbaum-Salant ve diğerleri, 2011; Scaffidi & Chambers, 2012).

Scratch programı, öğrencilere kod bloklarını birleştirerek programlamayı öğretmeyi ve program çıktısını anında gözleme imkânı sağlamaktadır. Scratch programında geometrik ve matematiksel kavramlar kullanılmaktadır. Öğrenciler Scratch ile kodlamayı

öğrendiklerinde, problemleri çözmek için önemli stratejileri, proje tasarlamayı ve iletişim fikirlerini öğrenirler. Öğrencilerin 21. yüzyıl becerilerinden yaratıcı düşünme, sistematik düşünme ve işbirlikli çalışma becerilerinin gelişmesine yardımcı olmaktadır (Calder, 2010; Resnick ve diğerleri, 2009).

Scratch çok fazla programlama bilgisi olmadan yaratıcılığı geliştirmek (Meerbaum-Salant ve diğerleri, 2011), etkileşimli projeler yapmak ve projeleri sosyal paylaşım sitesinde paylaşma imkânı sağlayacak şekilde tasarlanmıştır. Scratch ile programlama yapboz yapmak gibidir ya da Lego parçalarını birleştirmek gibidir (Malan & Leitner, 2007; Wilson & Moffat, 2010).

Scratch, öğrencilerin akranları ile iş birliğine girerek bireyin kendi kendine öğrenmesini desteklemektedir. Ayrıca gençleri proje paylaşımına, araştırmaya ve diğer programlama dillerine göre daha az öğretime odaklayarak öğrenmeye teşvik etmiştir. Günde ortalama 1500 yeni proje <http://scratch.mit.edu> sitesine yüklenmektedir. Scratch sosyal paylaşım sitesinde, kullanıcılar projelerini paylaşarak geri dönütler alabilirler, diğer kullanıcılar tarafından teşvik edilirler ve diğer projelerden öğrenirler. Scratch programının özellikleri şu şekilde sıralanabilir (Maloney ve diğerleri, 2010):

- Tek pencereci kullanıcı arayüzü,
- Canlılık,
- Kod bloklarını canlandırma,
- Hata mesajı olmaması,
- Komut setinin en aza indirilmesi.

Programlama; sistematik düşünmeyi, problem çözmeyi ve stratejileri tasarlamayı desteklemektedir. Programlama kişiye düşüncesini yansıtmaya imkânı sağlamaktadır. Scratch programının üç temel ilkesi şunlardır; daha fazla düşünebilme, daha fazla anlam katma ve diğer programlama dillerine göre daha fazla sosyallik sağlamaktır. Daha fazla anlam katma

ilkesi ile çeşitlilik ve kişiselleştirme özellikleri üzerinde durulmaktadır. Daha fazla sosyallik ilkesi ile de kişilerin birbirlerini desteklemesi, işbirliği ve birbirlerinin projelerini eleştirmeleri ve tamamlamaları üzerinde durulmaktadır (Resnick ve diğerleri, 2009).

Scratch esnek ve görsel yapısı ile programlamanın önündeki engelleri kaldırarak gençlere gelişmiş animasyonlar ve oyunlar yapma imkânı sağlamaktadır. Scratch sahnesini gerçek dünya tiyatro sahnesine benzetebiliriz. Scratch becerikli öğrencilere, hayal güçlerini sahneye yansıtma imkânı sağlamaktadır. Ayrıca yazılım, öğrencilere koordinat sistemi gibi matematiksel kavramları öğretir (Lee, 2011). Scratch sadece teknik becerileri öğretmekle kalmaz aynı zamanda öğrencilere iş birliğine gitmeyi teşvik eder. <http://scratch.mit.edu> öğrenciler için, <http://scratched.media.mit.edu> sitesi ise eğitimciler için tasarlanmıştır. <http://scratch.mit.edu/> sitesi kullanıcılar arasında yapıcı geribildirim oluşturulması için iyi bir alt yapı sağlamaktadır. Scratch sitesinde kullanıcıların projelerinin kod blokları görülebilmektedir, bilgisayara indirilebilmektedir ve yorum yazılabilmektedir (Dahotre, Zhang & Scaffidi, 2010; Lee, 2011). Scratch programının amacı kullanıcılara programlama becerilerinin kazandırılmasında yardımcı olmaktır. Scratch programı teknik beceriler kazandırmanın yanı sıra sosyal paylaşım sitesi ile kullanıcılarına sosyal beceriler de kazandırmaktadır. Scratch paylaşım sitesinde kullanıcılar projelerine yapılan yorumlar ile geri dönüt olarak projelerini diğer kullanıcıların katılımıyla yeniden tasarlayabilirler (Dahotre ve diğerleri, 2010).

Öğrenciler Scratch ile yaratıcı etkinlikler tasarlayabilir, problem çözme becerilerini geliştirebilir ve problemlere birden çok çözüm yolu bulabilirler, Scratch projeleri ile çalışarak deneyimlerini arttırabilir, bu projeleri yeniden tasarlayabilir ve işbirlikçi öğrenme aktiviteleri gerçekleştirebilirler (Kordaki, 2012; Resnick ve diğerleri, 2009).

2.2. Scratch Kullanımına Ait Öğrenci Görüşleri

Fesakis ve Serafeim (2009)'in yaptığı çalışmada Scratch kursuna katılan öğrencilerin %65'i Scratch programının kullanımının kolay olduğunu, %85'i ara yüzün basit ve anlaşılır olduğunu, %80'i ise Scratch arayüzünün estetik ve geliştirme ortamının tatmin edici olduğunu ifade etmişlerdir. Öğrencilerin %80'i, Scratch'ın basit bir programlama dili olduğunu, %90'ı ise grafik kod bloklarını faydalı bulduklarını belirtmişlerdir. Öğrencilerin %80'i Scratch programına katılmadan önce, eğitimde teknoloji kullanımının ilgi çekici olduğunu ifade etmişlerdir. Bu oran eğitimden sonra %92 olmuştur.

Genç ve Karakuş (2012) eğitimde bilgisayar oyunları tasarımı dersinde, Scratch kullanımına dair öğrenci görüşlerini ve deneyimlerini belirlemişlerdir. Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde eğitim gören 109 ikinci sınıf öğrencisi ile karma çalışma gerçekleştirilmiştir. Nitel veriler, araştırmacının ders bloguna bırakılan yorumlar ve öğrenci bloglarından, nicel veriler ise bloglardaki anketlerden elde edilmiştir. Öğrenciler daha önce blog deneyimi yaşamış olup, görsel bir ortam kullanılmayan programlama dersinde birçoğu başarısız olmuştur. Öğrencilerin çoğu Scratch programını basit, kolay ve eğlenceli bulmuşlardır. Scratch ile programlama yapılarının öğrenilmesinin kolay olduğunu belirtmişlerdir. Dersin kapsamında oyun temasının olması öğrencileri motive etmiştir.

2.3. Scratch ve Görsel Programlama

Adams ve Webster (2012)'in araştırmasında, 2008-2011 yılları arasındaki Scratch kampına katılan öğrenciler, video oluşturmaktan ziyade oyun oluşturmaya tercih etmişlerdir. Kamp temel programlama kavramlarını ortaokul ve lise öğrencilerine tanıtmak amacıyla tasarlanmıştır. 322 ortaokul ve lise öğrencisinin 300 farklı Scratch projesindeki kod bloklarının istatistiğini çıkartarak, t-test ile analiz etmişlerdir. Değişkenler, döngü deyimleri, koşul ifadeleri vb. temel öğelerin nasıl kullanılacağı öğrencilere gösterilmiştir. Öğrencilerin hikâye oluştururken koşul ve döngü ifadeleri kullanmasına ihtiyaç olmamakla birlikte oyun

tasarımları sürecinde koşul ifadeleri ve döngüleri kullanmaları gerekmektedir. Dolayısıyla oyun tasarımı projeleri, öğrencileri programlama deyimlerini öğrenmeye zorlamaktadır. Yaz kampında 209 Alice projesi, 103 Scratch oyun projesi, 10 Scratch video projesi yapılmıştır. Scratch oyun projelerinde daha fazla değişken, kod bloğu, koşullu ifade, döngü deyimleri kullanıldığı görülmüştür. Hikâyelerde en çok diyaloglar kullanılmaktadır. Videolarda neredeyse oyunlar kadar döngüler kullanılmış ama değişkenler ve koşullu ifadeler çok daha az kullanılmıştır.

Adams (2010) araştırmasında haftanın 5 günü 9:30-14:30 arası 30 lisans öğrencisi ile Scratch çalışması gerçekleştirmiştir. Öğrenciler olumlu bir bilgisayar deneyimi yaşadıklarını, programlamada temel fikirleri öğrendiklerini, bilgisayar alanında kariyer yapma isteği duyduklarını ifade etmişlerdir. Katılımcılar Scratch programının zorluğunu orta olarak nitelmişlerdir. Öğrencilerin %93'ü Scratch programı ile oyun, geri kalanlar ise video oluşturmuşlardır. Oyunlarda koşul ve mantık ifadeleri yoğun olarak kullanılmıştır. Videolarda ise hiçbir koşul ifadesi kullanılmamıştır. Kamp öğrenciler için olumlu geçmiştir ve programlama hakkında daha fazla bilgi edinmek istemişlerdir.

Lewis (2010) Logo ve Scratch programlarının tutum ve öğrenme çıktılarını karşılaştırmıştır. Öğrenciler iki gruba ayrılarak, ilk grup ilk 6 gün Scratch ile devamında Logo ile ikinci grup ilk 6 gün Logo devamında Scratch ile çalışmıştır. Öğrencilerin ilk altı gündeki algıları, hedefleri ve öğrenme çıktıları araştırılmıştır. İkinci ve beşinci günde yazılı değerlendirme yapılmıştır. Anketler yedinci günde doldurulmuştur. İlk grup 26, ikinci grup ise 24 6. sınıf öğrencisinden oluşmaktadır. Scratch programını öğrenen öğrenciler Logo programını öğrenen öğrencilere göre koşullu ifadeleri yorumlamakta daha başarılı olmuşlardır. Bu sonuç Scratch programının kompütasyonel kavramları öğretmekte başarılı olduğunu göstermektedir (Adams, 2010; Adams & Webster, 2012; Claypool, 2013). Bununla

birlikte Logo öğrenen öğrencilerin programlama yeteneğine güven düzeyleri Scratch programını öğrenen öğrencilere göre daha yüksek çıkmıştır.

Meerbaum-Salant, Armoni ve Ben-Ari (2013) Scratch programının öğrencilere bilgisayar bilimi kavramlarını öğretmede başarılı olup olmadığını araştırmışlardır. 20 haftalık sürede haftada iki ders saati, iki eğitim öğretim yılı süresince 204 ortaokul öğrencisi ve sekiz öğretmen ile araştırma gerçekleştirilmiştir. Gözlem ve görüşmeler ile nitel veriler toplanmıştır. Öğrencilerin tekrarlı yapılar, değişkenler gibi önemli bilgisayar bilimi kavramlarını öğrendikleri tespit edilmiştir.

Malan ve Leitner (2007) araştırmalarında, lisenin yaz okulunda bilgisayar bilimi dersinde ilk defa öğrencilere Scratch programı göstermişlerdir. Amaç daha sonra Java programına geçişi kolaylaştırmaktır. Öğrencilerin %76'sı sonraki Java deneyimlerinde Scratch ile çalışmalarının faydalı olduğunu ifade etmişlerdir, %16'sı kararsız kalmıştır.

2.4. Scratch ve Problem Çözme Becerisi

Kalelioğlu ve Gülbahar (2014) çalışmasında, Scratch ile programlama öğretiminin 5. sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisini araştırmışlardır. 49 5. sınıf öğrencisiyle 5 haftalık açıklayıcı karma yöntem araştırması gerçekleştirilmiştir. Ön test/son test yarı deneysel çalışmada kullanılan problem çözme becerisi ölçeğini, eşleştirilmiş örneklem t test ile analiz etmişlerdir. Eşleştirilmiş örneklem t test sonuçlarına göre, öğrencilerin problem çözme becerileri puanları arasında anlamlı bir farklılık olmadığı bulunmuştur. Araştırmacılar anlamlı bir farklılık bulunamamasının eğitim süresinin kısalığından ve örneklemin küçüklüğünden kaynaklanabileceğini ifade etmişlerdir. Sözü edilen yetersizliklerden dolayı başka çalışmalarda farklı sonuçlar bulunabileceği ifade edilmiştir. Karabak ve Güneş (2013)'in çalışmasında belirtildiği gibi Bilişim Teknolojileri ve Yazılım dersinin öğretim programında Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanının yıllık plandaki süresi 11 haftalık 22 ders saati süresine tekabül etmektedir.

Kaleliođlu ve Glbahar (2014) alıřmalarında 5 haftalık bir eđitim planlamıřlar, dolayısıyla problem özme becerisinin ölçlebilmesi için yeterli bir süre ayrılmamıř olabilir. Nitel veriler, alıřma sonunda odak grup görřme formu ve problem özme becerisi gözlem formu ile elde edilmiřtir. Nitel verileri özlemek için içerik analizi kullanılmıřtır. Arařtırmanın nitel bölümünde öđrenciler bir gözlem formu ile bađımsız bir gözlemci tarafından gözlenmiřtir. Nitel arařtırma sonuçlarına göre öđrencilerin çođu Scratch programında problem özme sürecinde zorlandıklarını, problemleri özzerken farklı yolları kullandıklarını, karakterlere komut vermekten hoşlandıklarını, Scratch programının kullanımını kolay bulduklarını ve kendilerini programlama konusunda geliřtirmek istediklerini ifade etmiřlerdir.

Calder (2010) altı yařındaki 26 öđrenci ile Scratch programlama dilinde alıřırken matematiksel düşünme yollarını arařtırmıřtır. Arařtırma sürecinin alt amaları problem özme sürecinde iliřkisel düşünme, durumu yorumlama ve deđerlendirme için mantık ve sebep-sonuç iliřkisi kurma ařamalarını arařtırmaktır. Arařtırma nitel arařtırma örnek olay incelemesidir. Öđrencilerin süreç ve yansımalar hakkındaki web günceleri, öđrenci ve öđretmen görřmeleri ve sınıf gözlemleri içerik analizi ile analiz edilmiřtir. Arařtırmanın sonuçları Scratch yazılımının matematiksel kavramları keřfetmek için deđerli ve motive edici bir programlama ortamı sađladığını, yaratıcı problem özme sürecini ve matematiksel düşünmeyi kolaylařtırdığını ve problem özme için ilgi ekici ve nispeten kullanımı kolay ve motive edici bir program olduđunu göstermiřtir.

Brown ve diđerleri (2008) ortaokul öđrencilerine problem özme becerilerini öđretmek için Scratch kullanımını arařtırmıřlardır. 113 ortaokul 5. ve 6. sınıf öđrencisine bir ay süresince haftada 45 dk. olmak üzere toplamda dört Scratch dersi verilmiřtir. Deney grubu öđrencilerinin problem özme becerileri kontrol grubu öđrencilerine göre daha yüksek çıkmıřtır. Arařtırmacılar, bir öđretim aracı olarak Scratch'ın problem özme için matematiksel akıl yürütme becerilerini uygulamada etkili bir yol olacađını, bu nedenle

Scratch'ın akademik performansı arttırmak için kullanılabileceğini ve Scratch ile işlenen derslerin öğrencilerin problem çözme becerilerini öğrenmesinde yararlı olabileceğini önermişlerdir.

Shin ve Park (2014) Scratch ortamında programlamanın ortaokul öğrencilerinin problem çözme kapasitesine etkisini ölçmek amacıyla 46 6. Sınıf öğrencisine beş faktörden oluşan problem çözme kapasitesi bilişsel test aracını uygulamışlardır. Çalışmada, Scratch programlama ile matematiksel mantık problemleri çözülmüştür. Araştırmanın sonucuna göre öğrencilerin problem çözme becerileri olumlu yönde etkilenmiştir. Farklı düşünme, karar verme ve planlama yeteneklerinde deney grubu lehine istatistiksel olarak anlamlı bir farklılık bulunmuştur.

Nam, Kim ve Lee (2010) Scratch programlama öğretimi için 60 6. Sınıf öğrencisi için bir kurs hazırlayarak, bu kursun problem çözme becerileri üzerindeki etkisini incelemişlerdir. Dört haftalık sürede sekiz saatlik Scratch programlama eğitimi verilmiştir. Ön test ve son test olarak öğrencilerin seviyelerine uygun olacak şekilde değiştirilen PISA 2003 problem çözme testi kullanılmıştır. Deney grubu öğrencilerinin kontrol grubu öğrencilerine göre problem çözme becerileri puanlarında istatistiksel olarak anlamlı bir farklılık bulunmuştur.

2.5. Scratch ve Yaratıcı Düşünme Becerisi

Giannakos ve diğerleri (2013) yaratıcı etkinlikler yoluyla bilgisayar biliminde öğrencileri motive etmek ve heyecanlandırmak için görsel programlama ile pasif öğrenenler yerine dijital medya yaratıcıları olma fikrini tanıtmayı amaçlamışlardır. Bunun için “Bilgisayar Bilimleri alanına öğrencilerin ilgisini çekmek için yaratıcı etkinlikler yapmak yeterli olur mu?” sorusuna cevap aramışlardır. İki farklı okuldan iki sınıftaki 29 kişiden oluşan 12 yaşındaki öğrenciler ile 2 günlük çalışma gerçekleştirmişlerdir. Dört yüksek lisans öğrencisi proje yöneticisi olmuştur. Nitel araştırmada veri toplama araçları olarak;

görüşmeler, fotoğraflar, gözlemler ve videolar kullanılmıştır. 11 öğrenci ile atölye çalışması hakkında görüşme yapılmıştır. Görüşmeler içerik analizi ile analiz edilmiştir.

Katılımcılar atölye çalışmasını genel olarak olumlu ve yaratıcı bulmuşlardır. Ayrıca kolay, ilgi çekici ve keyifli olarak tanımlamışlardır. Scratch programının yaratıcılığı teşvik etmek ve programlamayı öğrenmek için mükemmel bir araç olduğunu ifade etmişlerdir. Atölye çalışması kız öğrencilerin bilgisayar bilimine ilgi duymasını sağlamıştır. Öğrenciler atölye çalışması sonucunda Scratch programında altı adet etkileşimli projeyi tamamlayıp, paylaşmışlardır. Scratch programında kendi oluşturdukları karakter ile bir hikâye yazmanın öğrenciler için ilgi çekici olduğu belirtilmiştir.

Wilson ve Moffat (2010) araştırmalarında, sekiz yaşındaki 21 öğrenciye toplam sekiz ders saati eğitim vermiştir. Araştırma nitel-örnek olay incelemesidir. Veriler öğretmen görüşmesi ve öğrenci gözlemleri ile elde edilerek, içerik analizi ile analiz edilmiştir. Araştırmanın amacı çocukların yaratıcılığını eğlenceli bir şekilde geliştirmek ve aynı zamanda programlamayı tanıtmaktır. Araştırma sonuçlarına göre Scratch kullanımı programlama dersinin öğrenimini daha eğlenceli hale getirmiştir, öğrenme deneyiminde hayal kırıklığı ve kaygı yaşanmamıştır.

Burke ve Kafai (2010) bilgisayar programı yazmanın çocukların dijital hikâyelerini oluşturmada ve yaratıcılığını geliştirmedeki etkisini araştırmışlardır. 11 Ortaokul öğrencisine okul sonrası altı haftalık Scratch eğitimi verilerek 11 proje yapılmıştır. Katılımcıların %60'ı Scratch ile hikâye oluşturma sürecinin kendilerinin hikâye oluşturma yeteneklerini zamanla arttırdığını ifade etmişlerdir.

2.6. Oyun Tasarımı Öğrenmenin Faydaları

Oyun oynamaya kıyasla, oyun tasarım süreci karmaşık tasarım görevleri içeren bilişsel bir süreçtir. Tasarım görevleri, tam ve işlevsel bir sistem oluşturmak için pek çok birbiri ile ilişkili değişkenlerin ve parametrelerin kullanılmasını gerektirir (Robertson, 2012).

Oyun tasarım görevleri, yapılandırılmamış problemlerin çözümünde bizlere uygun pratik yapma ortamını sağlamaktadır (Jonassen, 2000) ve bu görevler problem durumları olarak tanımlanabilir. Çünkü hedeflenen oyunu tamamlamak için, süreçte önümüze bir sürü problem durumu ortaya çıkacaktır.

Akcaoglu ve Koehler (2014) oyun tasarımıyla öğrenmenin öğrencilerin problem çözme becerileri üzerindeki etkisini araştırmak için, kontrol grubu 24, deney grubu ise 20 öğrenciden oluşan yarı deneysel bir çalışma tasarlamışlardır. Ön ve son test olarak PISA problem çözme testi kullanılmış olup problem çözme testinde üç farklı problem tipinde sorular öğrencilere yöneltilmiştir. Sorular oyun tasarımına özgü sorular değildir. Yani oyun tasarımına özgü olmamakla birlikte diğer alanlara genelleştirilebilmektedir. Problem tipleri şu şekilde verilmiştir:

1. Sistem analizi ve tasarımı,
2. Karar verme,
3. Sorun giderme,

Kontrol grubundaki öğrenciler, gelecekte bir yaz programına katılmak isteyen öğrenciler arasından seçilmişlerdir ve okul sonrası Scratch oyun tasarım faaliyetlerine katılmamışlardır. Deney grubu öğrencileri ise toplamda 15 saat Scratch programında oyun tasarlamışlardır. Deney grubu öğrencilerinin problem çözme becerileri kontrol grubu öğrencilerinden anlamlı farklılık oluşturacak şekilde yüksek çıkmıştır. Sistem analizi ve tasarımı, karar verme ve sorun giderme becerilerinin her birinde deney grubunun puanı kontrol grubunun puanından anlamlı farklılık oluşturacak şekilde yüksek çıkmıştır. Araştırma sonuçlarına göre oyun tasarımıyla öğrenme öğrencilerin problem çözme becerilerinin artışına yol açmıştır.

2.7. Bilgi İşlemsel Düşünme

Bilgi işlemsel düşünme; bilgisayar biliminin kavramlarından faydalanarak problemleri çözmeye, sistemleri tasarlama ve insan davranışını anlamaya yönelik bir yöntem olarak tanımlanmaktadır (Barr, Harrison & Conery, 2011; Wing, 2006). Yani, bireyin gerçek hayatta karşılaştığı bir problemi çözmek için bilgisayar bilimcisi gibi düşünmesi gerektiğini ifade etmektedir.

Barr ve diğerleri (2011), bilgi işlemsel düşünmeyi bir problem çözme süreci olarak ele almaktadırlar. Bu sürecin basamaklarını ise şöyle ifade etmektedirler:

- Problemleri bilgisayar veya başka araçlar yardımı ile çözebilmek için formülleştirme,
- Verileri mantıklı bir şekilde düzenleme ve çözümlenme,
- Modeller, simülasyonlar aracılığı ile verileri sunma,
- Algoritmik düşünme yoluyla çözümleri otomatikleştirme,
- Kaynakları etkili bir şekilde kullanarak en optimum çözümleri tanımlama, çözümlenme ve uygulama,
- Bu problem çözme sürecini çeşitli problemlere genelleme ve transfer etme.

Bilgi işlemsel düşünme becerisinin geliştirilmesi için; karmaşık problemlerle uğraşırken kişinin kendine güvenmesi ve sabırlı olması, açık uçlu problemlerle çalışması, ortak bir hedef veya çözüme ulaşmak için başkalarıyla iletişim kurması ve birlikte çalışması gerekmektedir. Problemlerin çözümünde insan düşüncesinin gücünün bilgisayarlar ve diğer dijital araçlar ile genişletilmesi gerekmektedir (Barr ve diğerleri, 2011).

Aho (2012) ve Michaelson (2015) ise bilgi işlemsel düşünmeyi, bireylerin problemlerin çözümüne yönelik attıkları adımların algoritma olarak formüle edilmesi ile ilgili düşünce süreçleri olarak görmektedir (Şekil 1).

Şekil 1

Bilgi işlem üçgeni Michaelson (2015)



Bilgi işlem üçgeninde yer alan kavramlar aşağıda detaylandırılmıştır.

- Bilgi işlemsel düşünme problemi anlamamızı sağlar.
- Bilgi ve iletişim teknolojileri kullanılarak problemlerin çözüm yollarının araştırılmasında bilgi işlemsel düşünme gerekmektedir.
- Bilgisayar bilimi:
 - Teori ve uygulama arasında bir sinerjidir.
 - Yazılım araçlarıyla sosyal ihtiyaçları karşılamaktadır (Michaelson, 2015).

2.8. Problem Çözme Becerisi

Problem çözme, belirli bir durumdan istenilen duruma ulaşana kadarki engellerin aşılma sürecidir (Funke, 2010). Programlar ve programlama yoluyla geliştirilen yazılımlar,

komutlardan veya kodlardan meydana gelmiştir. Ancak komutların sıralanışı rastgele değil, tam tersine bir problemi çözmek üzere planlı bir şekilde tasarlanmalıdır. Bu nedenle, öğrenenlerin bir problemi çözmek için bir “program tasarımı” yapmaları gerekmektedir. Bu durum ise problem çözme becerisi ile açıklanabilir (Gültekin, 2006).

Bilişim Teknolojileri ve Yazılım dersi kapsamında kazandırılması hedeflenen bilgi, beceri ve değerlere ilişkin yeterliklerde aşağıdaki ifadeler yer almaktadır (MEB, 2012).

Problem Çözme, Programlama ve Özgün Ürün Geliştirme Alanındaki Yeterlikler:

- Bir problemi çözmek ve projeyi gerçekleştirmek için strateji geliştirebilir, çözüm üretirken farklı bakış açılarını ve yaklaşımları kullanabilir.
- Yazarlık ve programlama dillerini tanıyabilir, en az bir yazarlık/ programlama dilini etkili biçimde kullanabilir.
- Sistemleri ve konuları incelemek için model, benzeşimler ve canlandırmalar oluşturabilir.

2.9. Problem Türleri

Problemler yapısal karmaşıklığına göre yapılandırılmış problemler ve yapılandırılmamış problemler olmak üzere ikiye ayrılmaktadır (Jonassen & Kwon II, 2001). Yapılandırılmış problemler özellikle ders kitaplarındaki ünite değerlendirme sorularıdır. Bu tür problemler iyi tanımlanmış parametreler dâhilinde sınırlı sayıda kavram, kural ve ilkenin sınırlı sayıda çözüm için uygulanmasını gerektirir. Yapılandırılmış problemlerin temel özellikleri şunlardır (Jonassen, 1997):

- Problemin tüm unsurları sunulur.
- Olası çözüm önerileri, iyi tanımlanmış problem ile öğrenenlere sunulur.
- Sınırlı sayıda kural ve ilke, çözüm sırasında kullanımları tahmin edilebilir biçimde uygulanır.
- Doğru ve tahmin edilebilir cevaplara sahiptirler.

- Bu tür problemlerin çözümlerinden kazanılan beceriler benzer alanlara aktarılabilir.

Yapılandırılmamış problemler ise birden fazla çözüme sahiptirler. Çözüm için birden fazla disiplindeki bilgilerin organize edilmesi gerekebilir. En iyi çözüm için hangi kavram, kural ve ilkeler kullanılacaksa organize edilir. Bu tür problemlerin çözümü daha zor olmakla birlikte, günlük yaşamda karşılaşılan problemler olduğundan dolayı bireyler için daha ilgi çekicidir. Problemin çözümünde hangi bilgi ve becerilerin kullanılacağına karar vermek için problemin tanımlanması gereklidir. Yapılandırılmamış problemlerin bazı temel özellikleri şunlardır (Jonassen, 1997):

- Yapılandırılmamış olarak adlandırılırlar, çünkü problemin bir yâda daha fazla unsurları bilinmez ya da eksik olarak bilinmektedir.
- Çözüm için istenilenler belli belirsiz tanımlanmıştır ya da açık değildir.
- Bazen birden çok çözüm yoluna sahiptirler, bazen de hiç çözümleri yoktur.
- Çözümlerin değerlendirilebileceği ölçüt sayısı birden fazladır.
- Kontrol edilebilecek parametre sayısı azdır.

Çoklu bakış açısı oluşturarak çözüme ulaşabilmek için, öğrencileri problem hakkında fikirlerini birbirlerine söylemeye, yargıda bulunmaya ve buldukları yargıyı savunmaya zorlar. Bu nedenle bu tür problemlerin çözümü işbirlikli çalışmayı gerektirir.

Yapılandırılmamış problem görevleri ile meşgul olmak gerçek hayatta bilginin uygulanmasını kolaylaştırmakla birlikte bilginin transfer sürecini de kolaylaştırmaktadır. Yapılandırılmamış problemler, yapılandırılmış problemlere göre daha karmaşıktır ve çözümü daha zordur.

2.10. Yapılandırılmamış Problemlerin Çözüm Süreci

Kavramsal olarak yapılandırılmamış problemleri çözmek, problem çözümü için sistematik olarak bir araştırma değil de bir tasarım süreci olarak düşünülebilir. Voss ve Post (1988)'e göre yapılandırılmamış problem çözme süreci problemi tanımlama, çözüm üretme ve

değerlendirme adımlarından oluşmaktadır. Problemin tanımlanması aşamasını problemin çözümü için en önemli adım olarak nitelemektedirler. Çünkü problemin tanımlanması esnasında kavramların ve ilişkilerin ortaya çıkarılması gerekecektir.

Hong (1998) yapılandırılmamış problemlerin çözüm sürecini: problemin tanımlanması, problemin çözümü, izleme ve değerlendirme aşamaları olarak tanımlamıştır. Yapılandırılmamış problemlerin çözüm sürecinde genel olarak literatürdeki adımlar problemin tanımlanması, çözümü, karar verme, izleme ve değerlendirme olarak görülmektedir (Hong, 1998; Voss & Post, 1988)

2.10.1. Problemin tanımlanması. Yapılandırılmış bir problemi çözerken problemin amacı kolaylıkla tanımlanabilir, bu yüzden problemin tanımlanması aşaması problemin ayrıştırılmasına ve problem türünün sınıflandırılmasına odaklanır. Oysaki yapılandırılmamış problemlerin çözüm sürecinde, problemin açıklamaları net değil veya amaçları iyi tanımlanmamışsa, yapılandırılmamış problem birden fazla sunuma ve anlayışa sahip olabilir. Bu yüzden problemin tanımlanması aşaması problem çözme sürecinin en önemli adımıdır (Jonassen, 1997). Problemin tanımlanması aşamasında problemi çözen birey verilen bilgiyi çözümleyerek problemi anlamaya çalışır veya mevcut bilgi ile bağlantı kurmaya çalışarak bilgilerin entegre edilmesiyle problemi tanımlar. Bu aşamada problemin bazı özellikleri hafızadaki bilgiyi harekete geçirerek zihinde bir şema etkinleştirilebilir (Ge, 2001). Problemin tanımlanması, problemin yorumlanmasına veya anlaşılmasına bağlıdır. Problemin tanımlanması aşaması yapılandırılmamış problemlerin çözümünde önemli bir süreçtir, çünkü problemin çözüm sürecini kolaylaştırır. Bu süreçte, olası çözümleri test etmek ve bilgiyi kodlamak için problemin başlangıçtaki durumu detaylandırılmalı ve kısıtlamalar tanımlanmalıdır (Chi & Glaser, 1983).

2.10.2. Problemin çözümü. Bir problem ile karşı karşıya kalındığında beynimiz hafızada daha önceki benzer bir durumu arar. Öğrenci, önceki ve mevcut sorun arasındaki

benzerlikleri tanımaya teşvik edilebilirse, önceki problemde kullanılan çözüm yöntemini hatırlayabilir (Jonassen, 1997). Yapılandırılmamış problemlerin, belirli bir karar alma süreci kullanılarak saptanabilen tek bir çözüm yolu yoktur. Etki alanı ve yapısal bilgi doğal olarak çözüm sürecini kolaylaştırır. Etki alanına özgü bilgi en iyi çözüm yolunun seçilmesine yardım eder (Chi & Glaser, 1983). Metabilişsel beceriler de çözüm sürecini kolaylaştırır. Uygun çözümlerin seçilmesi; çözüm sürecine yansıtılması, karşılaştırılması ve alternatiflerin değerlendirilmesi yeteneklerini gerektirir (Voss & Post, 1988).

2.10.3. Karar verme. Argümanlar oluşturma yapılandırılmamış problem çözümünün önemli bir sürecidir. En iyi eylem seçeneğini belirlemek ve alternatif çözümlerin uygulanabilirliğini değerlendirmek için, problemin tanımlanması ve problemin çözümü süreçlerinin her ikisine de ihtiyaç vardır. Çünkü yapılandırılmamış problemler birden fazla şekilde tanımlanabilir ve birden fazla çözüm yolu vardır (Jonassen, 1997).

2.10.4. İzleme ve değerlendirme. Yapılandırılmamış problemlerin çözümünde tercih edilen çözüm yolunu alternatif çözümlere karşı savunmak için en uygun, en savunulabilir ve en inandırıcı argüman oluşturulmalıdır. Problemi çözen kişi kanıtlara, destekleyici gerçeklere veya varsayımlara dayanan argümanlar oluşturmalıdır (Jonassen, 1997). Problem çözüm süreci, çözümün ve seçeneklerin haklı çıkarılması için sürekli olarak izlenmeli ve değerlendirilmelidir (Voss & Post, 1988).

Tablo 1’de yapılandırılmamış problemler için uygulama sürecinde tasarımcı ve öğrencilerin rolleri özetlenmiştir.

Tablo 1

Yapılandırılmamış problemler için uygulama süreci (Jonassen, 1997)

Tasarımcı/Geliştirici

Öğrenci

	Hedefleri ifade edin / Problemi doğrulayın
Problem alanını ifade edin	Problem alanıyla ilgili problem hedeflerini
Problem kısıtlamalarını tanıttın	ilişkilendirin
Bulun, seçin ve durumları geliştirin	Alternatif bakış açınızı netleştirin
Bilgi tabanını yapılandırın/Öğrencilere sunun	Probleme çözümler üretin
Bilgi kaynakları sağlayın	Destekleyici/reddedici kanıtları toplayın
Argüman oluşturmayı destekleyin	Geçerliğini belirleyin/Argümanlar inşa
Problem çözümlerini değerlendirin	edin
	Uygulayın ve çözümü izleyin
	Çözümü uyarlayın

Bu tez çalışmasında, yapılandırılmamış problemlerin çözüm süreci için kullanılan ölçek: yorumlama ve problemi tanımlama, çözümler geliştirme ve çözüm sürecini izleme, karar verme ve problem çözüme sürecini değerlendirme ve özel stratejiler ile ilgili maddelerden oluşmaktadır. Bu çalışma, 2014-2015 eğitim öğretim yılı ikinci döneminde Bilişim Teknolojileri ve Yazılım dersi kapsamında 11 haftalık bir sürede gerçekleştirilmiştir. 11 haftalık süreçte öğrencilerle birlikte yapılan etkinlikler Ek 5 ve Ek 6'da verilmiştir.

3. Bölüm

Yöntem

Yöntem bölümünde araştırmanın modeli, örneklem (çalışma grubu), veri toplama araçları ile verilerin toplanması ve çözümlenmesi süreçleri tanımlanmıştır.

3.1. Araştırmanın Modeli

Scratch ile programlama öğretiminin öğrencilerin problem çözme becerileri üzerindeki etkisini saptamak, öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımlarını belirlemek ve Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanındaki yeterliklere ilişkin öğrenci görüşlerini değerlendirmek üzere Bursa-Osmangazi Şükrü Naili Paşa Ortaokulu'nda gerçekleştirilen bu çalışmada karma yöntem araştırması kullanılmıştır.

Creswell ve Clark (2007) karma yöntem araştırmasını, nitel ve nicel araştırma yöntemlerini kullanarak verileri toplama, analiz etme ve bulguları bütünleştirmeye olanak veren araştırma olarak tanımlamaktadır. Yıldırım ve Şimşek (2013)'e göre olay ve olgular karmaşık ve çok boyutludur, özellikle sosyal bilimlere ait problemlerin bütüncül ve zengin bir çerçevede anlaşılması için nitel ve nicel yöntemlerin bir arada kullanılması gerekmektedir. Mason (2006) her olay ve olgunun (ya da gerçeğin) çok boyutlu olduğunu, sadece nitel ya da nicel boyutuyla incelenmesi halinde ortaya çıkan algının eksik olacağını vurgulamaktadır. Algının eksik kalmaması için farklı yöntemler aynı çalışmada birlikte kullanılmalıdır. Bu çalışmada, karma araştırma yöntemi kullanılarak problem durumlarını açıklamada daha detaylı verilere ulaşılmak amaçlanmıştır.

Karma yöntem araştırması sırasında izlenmesi gereken adımlar altı başlık altında toplanabilir (Creswell & Clark, 2007):

1. Araştırma sorularına dayalı olarak nitel ve nicel verilerin ikna edici titizlikle toplanması ve analiz edilmesi,

2. Aynı anda iki veri türünü bir araya getirerek, birbiri ardına kullanılarak veya birini diğerinin içine katıştırarak bütünleştirme,
3. Araştırmada vurgulanmak istenen veri türlerinden birine veya her ikisine öncelik verilmesi,
4. Bu prosedürlerin tek bir çalışmada veya bir çalışma programının birden fazla aşamasında kullanılması,
5. Bu prosedürlerin, felsefi dünya görüşleri ve kuramsal bakış açıları kapsamında çerçeve içine alınması,
6. Bu prosedürlerin, çalışmayı yürütmek için planı yönlendiren belirli araştırma deseni ile birleştirilmesi.

Bu araştırmada, karma yöntem araştırması desenlerinden açıklayıcı desen kullanılmıştır. Nicel yöntemle elde edilen verilerin desteklenmesi, örneklendirilmesi ya da açıklanması için nitel yöntemle elde edilen verilere ihtiyaç vardır. Açıklayıcı desen sürecinde; nicel yöntemlerle toplanan verilerin analizinden yola çıkarak nitel veriler toplanır. (Yıldırım & Şimşek, 2013). Deneysel çalışma sonucundan sonra nitel veri eklemek için nedenler şu şekilde sıralanabilir (Creswell & Clark, 2007):

- Katılımcıların deneyin sonuçlarını nasıl gördüklerini anlamak,
- Deneyi gözden geçirmek için katılımcının geribildirimini almak,
- Nicel çıktıları açıklamaya yardımcı olmak,
- Bir deneyden sonra bağımsız değişkenin uzun vadeli, sürekli etkilerini belirlemek,
- Uygulamaların teorik bir modelde nasıl çalıştığını daha derinlemesine anlamak,
- Sonuçların karşılaştırılması temel verilerle yapıldığında bağlamı değerlendirmektir.

Araştırmanın nicel kısmı basit deneysel modellerden tek grup ön test / son test model ile yürütülmüştür. Araştırmanın bağımlı değişkeni problem çözme becerisi, bağımsız

değişkeni ise Scratch ile oyun programlamadır. Nitel araştırma deseni olarak durum çalışması kullanılmıştır. Açıklayıcı desenin bir sonraki adımı ise verilerin bütünlük içinde sunulması yani verilerin yorumlanması sürecidir. Bu tasarımda, nicel verilerin analizinden sonra nitel verilerin analizine geçilmiştir. Bu araştırma gerçekleştirilirken izlenen sıra aşağıda özetlendiği gibidir.

Şekil 2

Açıklayıcı desene göre gerçekleştirilen araştırma süreci

NİCEL ARAŞTIRMA	
DESEN: Basit Deneysel	
1- Verilerin Toplanması:	Problem Çözme Becerisi Ölçeğinin Öğrencilere Uygulanması Ölçeğin Sonundaki Açık Uçlu Sorunun Öğrencilere Uygulanması
2- Verilerin Analizi:	Wilcoxon İşaretili Sıralar Testinin Kullanımı
NİTEL ARAŞTIRMA	
DESEN: Durum Çalışması	
1- Verilerin Toplanması:	Yapılandırılmamış Görüşme Formunun Öğrencilere Uygulanması
2- Verilerin Analizi:	Betimsel Analiz İçerik Analizi



VERİLERİN YORUMLANMASI

Basit deneysel yaklaşımın en önemli özelliği kontrol gruplarını barındırmamasıdır. Basit deneysel model kapsamında, problem çözme becerisi ölçeği ile ön test ve son test olacak şekilde deney grubundan ölçümler alınmıştır. Nitel veriler, nicel çıktıları açıklamaya yardımcı olmak, Scratch ile oyun programlama sürecinin problem çözme süreci gibi teorik bir modelde nasıl çalıştığını daha derinlemesine anlamak ve bağlamı değerlendirmek için toplanmıştır.

Durum çalışması nasıl ve niçin sorularını temel alan, araştırmacının bir olgu ya da olayı derinlemesine incelemesine olanak veren araştırma yöntemidir. Gözlem, görüşme ve doküman yoluyla elde edilen bilgiler içerik analizi, betimsel analiz gibi yöntemlerle incelenebilir (Yıldırım & Şimşek, 2013).

3.2. Çalışma Grubu

Araştırmanın örneklemini Bursa ili Osmangazi ilçesi Şükrü Naili Paşa Ortaokulunda öğrenim gören 109 erkek, 117 kız öğrenci olmak üzere toplam 226 5. ve 6. sınıf öğrencisi oluşturmaktadır.

Nicel verilerin toplanması için basit deneysel modellerden tek grup ön test / son test modelde kullanılan ölçek, seçkisiz olmayan örnekleme yöntemlerinden tabakalı amaçsal örnekleme yöntemi kullanılarak Bilişim Teknolojileri ve Yazılım dersinin zorunlu ders olduğu kademeler olan 5. ve 6. sınıfta öğrenim gören 226 öğrenciye uygulanmıştır. Nitel araştırma yöntemlerinden durum çalışmasında; seçkisiz olmayan amaçsal örnekleme yöntemlerinden ölçüt örnekleme yöntemi kullanılarak bilgi açısından zengin verilerin toplanabileceği 27 öğrenciden görüşme yöntemi ile derinlemesine bilgi toplanmıştır.

Tabakalı amaçsal örnekleme, ilgilenilen belli alt gruplar arasında karşılaştırmalara imkân sağlamak, grupların özelliklerini göstermek ve betimlemek amacıyla tercih edilir. Kolay yolla ulaşılan deneklerle çalışılabilir ve tüm tabakalardan sabit sayıda eleman da seçilebilir. Ölçüt örneklemede ise belirlenen ölçütü karşılayan birimler örnekleme alınır. Nicel

çalışmadan sonra derinlemesine bir izleme çalışması yapılmak istendiğinde kullanılabilir (Büyüköztürk, Çakmak, Akgün, Karadeniz & Demirel, 2017).

3.3. Veri Toplama Araçları

Yapılandırılmamış problemlerin çözümüne yönelik Ge (2001) tarafından öğretim sistemleri alanındaki doktora tezinde geliştirilen, Coşkun (2004) tarafından Türkçe 'ye çevrilen problem çözme becerisi ölçeği 5. ve 6. sınıf öğrencilerinin problem çözme becerilerini belirlemek amacıyla ön test ve son test olarak uygulanmıştır. Belirtilen ölçekte her biri bir problem basamağına ait toplam dört ana madde, her maddeye cevap olabilecek beşer adet cümle yer almaktadır. Her cümle:

- Her zaman
- Sık sık
- Ara sıra
- Pek az
- Hiçbir zaman şeklinde cevaplandırılabilir niteliktedir.

Cevaplar 1-5 arasında değişen değerlerle puanlandırılarak derecelendirilmiştir. Ölçek 4 problem basamağında toplam 20 cümleyi içermektedir. İlk beş madde yorumlama ve problemi tanımlama aşaması ile ilgili, 6-10 arası maddeler çözümler geliştirme ve çözüm sürecini izleme aşaması ile ilgili, 11-15 arası maddeler karar verme ve problem çözme sürecini değerlendirme aşaması ile ilgili maddelerden oluşmaktadır. 16-20 arası maddeler ise problem çözümünde kullanılan özel stratejiler ile ilgili maddelerden oluşmaktadır.

Creswell (2013)'in veri toplama sürecindeki görüşme adımlarına istinaden görüşmeler aşağıdaki dokuz başlıkta derlenmiştir:

1. Görüşmelerle cevaplanacak araştırma sorularına karar verildi.

2. Amaçlı örnekleme yöntemlerinden ölçüt örnekleme yöntemi kullanılarak bilgi açısından zengin verilerin toplanabileceği öğrenciler, Scratch oyun projesinde ve düşüncelerini ifade etmede başarılı öğrenciler içerisinde belirlendi.
3. Yapılandırılmamış açık uçlu sorular ile görüşme formu hazırlandı.
4. Birebir standartlaştırılmış açık uçlu görüşmeler yapılırken kayıt araçları kullanıldı.
5. Görüşme kılavuzu yedi açık uçlu soru ile tasarlandı.
6. Görüşme soruları pilot test yoluyla üç öğrenci üzerinde uygulanarak, öğrencilerin soruları anlayıp anlamadıklarına ilişkin olarak değişiklikler yapıldı.
7. Görüşmenin gerçekleştirileceği uygun ortam belirlendi.
8. Görüşme yerinde, görüşülen kişiden araştırmaya katılması için izin alındı.
9. Görüşme kurallara uygun olarak gerçekleştirildi.

11 haftalık Scratch eğitimi sonunda, ikinci araştırma sorumuz olan “Scratch ile programlama öğretiminde öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımları nelerdir?” sorusuna yanıt bulmak için 27 öğrenciye Tablo 2’de verilen görüşme soruları yöneltilmiştir.

Tablo 2

Problem çözme sürecine ilişkin görüşme soruları (Ge, 2001)

Problem Çözme Süreci	Görüşme Sorusu
Yorumlama ve problemi tanımlama	Scratch’ ta zor bir problemi çözmeye başlamadan önce ne yaparsın?
Çözümler geliştirme ve çözüm sürecini izleme	Scratch’ ta problem üzerinde çalışırken ne yaparsın?
Karar verme ve problem çözme sürecini değerlendirme	Scratch’ ta problem üzerinde çalışmayı bitirdikten sonra ne yaparsın?

Üçüncü araştırma sorumuz olan “Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanındaki yeterliklere ilişkin öğrenci görüşleri nelerdir?” sorusunun yanıtını bulmak için aşağıdaki görüşme soruları 27 öğrenciye yöneltilmiştir.

- Scratch’ ta problemler üzerinde hangi yöntemi (strateji, yol) uygulayarak çalışıyorsun?
- Scratch ile çalışmanın hayal gücünüzü geliştirdiğini düşünüyorsanız, bu duruma uygun örnek verebilir misiniz?
- Scratch programının hoşunuza giden yönleri nelerdir?
- Scratch programının hoşunuza gitmeyen ve size göre eksik yönleri nelerdir?

3.4. Verilerin Toplanması ve Çözülmesi

Görüşmeler iki haftalık bir süreçte gerçekleştirilmiştir. İkinci araştırma ve üçüncü araştırma sorusu için 27 öğrenci ile ses kayıtları alınarak derinlemesine görüşmeler gerçekleştirilmiştir. Görüşmeler gerçekleştirilmeden önce, katılımcılara verdikleri cevapların sadece bu araştırmada kullanılacağı ve görüşmenin amacı ile ilgili bilgiler verilmiştir.

Nicel veriler toplanırken kullanılan problem çözme becerisi ölçeği ön test olarak 17-20 Mart 2015 tarihleri arasında, son test ise 2-5 Haziran 2015 tarihleri arasında internet ortamındaki bir form aracılığıyla öğrencilere uygulanmıştır. Öğrenciler aradaki 11 haftalık sürede haftada iki ders saati olmak üzere 22 ders saati süresince, ilk kez kullandıkları Scratch programında oyun programlamışlardır. Karabak ve Güneş (2013)’in çalışmasında belirtildiği gibi Bilişim Teknolojileri ve Yazılım dersinin öğretim programında Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanının yıllık plandaki süresi 11 haftalık 22 ders saati süresine tekabül etmektedir. Ön testler Scratch ile oyun programlama sürecine başlamadan önce, son testler ise oyun projelerinin bitiminde uygulanmıştır.

Toplanan nitel veriler çözümlenirken ikinci araştırma sorusunda betimsel analiz, üçüncü araştırma sorusunda içerik analizi yöntemi kullanılmıştır. Betimsel analiz, içerik

analizine göre daha yüzeyseldir ve daha çok araştırmanın kavramsal çerçevesinin önceden bilindiği araştırmalarda kullanılır. İçerik analizinde ise amaç, benzer verileri belirli kavramlar ve kategoriler çerçevesinde bir araya getirerek düzenlemek ve yorumlamaktır. Verilerin analizi dört aşamada gerçekleştirilir: (1) verilerin kodlanması, (2) temaların bulunması, (3) kodların ve temaların düzenlenmesi, (4) bulguların tanımlanması ve yorumlanması olmaktadır (Yıldırım & Şimşek, 2013). Araştırma kapsamında bu dört aşamada gerçekleştirilen işlemler aşağıdaki şekilde gerçekleştirilmiştir:

1. **Verilerin kodlanması:** Elde edilen görüşme verileri ilk sütunda öğrenci takma ismi, ikinci sütunda katılımcıların sorulara verdikleri cevaplar, üçüncü sütunda ise ilgili kodların yer aldığı bir Excel tablosu üzerinde kodlanmıştır. Öğrencilerin verdikleri cevapların ilk aşamada, oluşturulan kodun altına alınıp alınmadığına bakılmış, alınamıyorsa yeni bir kod türetilmiştir. Betimsel analizde kullanılan temalar problem çözme literatüründen alındığından verilerin analizi için hâlihazırda mevcut bir yapı vardır ve bu teorik çerçeve verilerin analizini kolaylaştırmıştır. Bu teorik çerçeveye göre kodlama yapılarak ortaya çıkan yeni kodlar listeye dâhil edilmiştir. İçerik analizinde kullanılan kodlar ve temalar verilerden çıkarılan kavramlar ve bu kavramları karşılayacak temalara ulaşacak biçimde tümevarımcı bir analize tabi tutularak ortaya çıkarılmıştır. Kullanılan kodların bir kısmı literatürden alınmış, bir kısmı ise öğrencilerin verdikleri cevaplardan ortaya çıkarılmıştır. Verilerin içerisinde rastgele seçilen kodlar ve katılımcıların cevaplarının hangi koda girdiğine ilişkin bilgi Tablo 3 ve Tablo 4'te gösterildiği gibidir.

Tablo 3

İkinci araştırma sorusu için betimsel analiz örnek tablo

Katılımcı İsimleri	Katılımcıların Sorulara Verdikleri Cevaplar	Kod
Kardelen	Projeyi aklımda tasarlıyorum, yapacaklarımı planlıyorum, senaryoyu kurguluyorum, kuklanın yapacaklarını düşünüyorum.	Kurgulama
Ahmet	Oyunun nasıl bir şey olabileceğini düşünürüm. Kafamın içinde canlandırıyorum. Scratch kuklalarıyla kurgularım. Gerçek hayatla ilişki kurarım.	Kurgulama
Büşra	Hangi kod bloklarını kullanarak yapabilirim diye düşünüyorum. Kâğıt üzerinde senaryoyu çiziyorum.	Tasarlama

Tablo 4

Üçüncü araştırma sorusu için içerik analizi örnek tablo

Katılımcı İsimleri	Katılımcıların Sorulara Verdikleri Cevaplar	Kod
Hakan	Scratch biraz da hayal kurmakla ilgili, aklınıza getirdiğiniz bir hayvanı Scratch programında hareket ettirebilirsiniz. Bir yarasa yapıp onu uçurabiliriz.	Canlandırma
Muhammed	Pikachu, oyunda normalde düz gidebilirdi ama onu yukarıya zıplattık, aşağı yapacaktık ama yapmadık çünkü kendi kendine aşağıya inecekti, canavarı ekledik ama onu koymayabilirdik, yani kuklalarla oyunu zenginleştirmek hayal gücümü geliştirdi.	Kukla Çeşitliliği

Cemre Oyunu tasarlama aşamasında hangi kuklalarla nasıl Kurgulama
tasarlayacağım esnasında zihnimde düşünürken, hayal
gücümü zorladı.

2. **Temaların bulunması:** Yıldırım ve Şimşek (2013) kategori ya da temaları, kodların ya da kavramların kategorize edilerek daha genel bir başlık altında toplanması olarak tanımlamaktadır. Ortaya çıkan temaların sayısı fazla ise bu temaları kapsayacak daha genel bir tema ortaya çıkarılabilir. İkinci araştırma sorusu için betimsel analizde kullanılan temalar literatürdeki (Ge, 2001) problem çözme adımlarına göre organize edilmiş ve üç başlık altında toplanmıştır: (1) yorumlama ve problemi tanımlama, (2) çözümler geliştirme ve çözüm sürecini izleme, (3) karar verme ve çözüm sürecini değerlendirme. Üçüncü araştırma sorusu için içerik analizinde kullanılan temalar görüşme sorularına göre organize edilmiş ve dört başlık altında toplanmıştır: (1) Özel stratejiler (2) Hayal gücü (3) Scratch programının olumlu yönleri (4) Scratch programının olumsuz ve eksik yönleri.
3. **Kodların ve temaların düzenlenmesi:** İkinci araştırma sorusunda, toplamda üç başlık altında dokuz adet kod türetilmiştir. Üçüncü araştırma sorusunda, toplamda dört başlık altında on beş adet kod türetilmiştir. Oluşan bu kodlara bulgular başlığı altından ulaşılabilir. Bu işlem yapılırken dikkat edilen hususlar, kodların ve temaların sade ve anlaşılır olması, kodların düşünce birimlerini açıklayıcı olması ve temaların kodları genel başlıklar altında toplayabilir olması olarak listelenebilir.
4. **Bulguların tanımlanması ve yorumlanması:** Bu aşamada betimsel analiz ve içerik analizi kullanılmıştır. Yıldırım ve Şimşek (2013)'e göre araştırmacının bu son aşamada elde ettiği verileri kodlara ve temalara göre betimlemesi, alıntılara

yer vermesi, örneklendirmesi, açıklaması, yorumlaması ve görsel hale getirerek araştırma problemleri ile bağlantının net olarak ortaya konması gerekmektedir.

Ön test ve son test olmak üzere iki aşamada toplanan nicel verilerin normal dağılım gösterip göstermediğini bulmak için, Kolmogorov-Smirnov testi uygulanmıştır. Parametrik olmayan testlerden Wilcoxon işaretli sıralar testi ile öğrencilerin problem çözme becerileri puanları arasında anlamlı bir farklılık olup olmadığına bakılmıştır. Parametrik olmayan testlerden Mann-Whitney U testi ile öğrencilerin problem çözme becerileri puan ortalamalarının cinsiyete ve sınıf seviyesine göre anlamlı bir farklılık gösterip göstermediğine bakılmıştır.

3.5. Güvenirlilik

Organize edilen veriler iç güvenirliliği sağlamak için bir yardımcı araştırmacıya daha gönderilmiştir. Yardımcı araştırmacıdan verileri kodlaması istenmiştir. Yardımcı araştırmacıya gönderilen ve kodlanmış Excel tablosunun örnek bir kesiti Tablo 5'te verilmiştir. Yardımcı araştırmacı bilgisayar ve öğretim teknolojileri öğretmenliği lisans programı mezunu olup 11 yıldır bilişim teknolojileri öğretmenliği yapmaktadır ve Scratch programını derslerinde kullanmaktadır. Elde edilen kodlar ikinci sütundan başlayacak şekilde, ilk satıra yan yana sırasıyla yazılmıştır. Katılımcıların cevapları ise ilk sütunda alt alta sıralanmıştır. Her veri satırı için yardımcı araştırmacıdan oluşturulan kodlardan yalnızca bir tanesini seçmesi istenmiştir. Yardımcı araştırmacı kodlamayı, katılımcıların verdiği cevap hangi koda uyuyorsa ilgili satır ve sütunun kesiştiği hücreye 1 yazarak gerçekleştirmiştir. Katılımcıların verdikleri cevaplar yalnızca bir koda uyacak şekilde yeniden düzenlenmiştir. Kodlamayı kolaylaştırmak için kodlara ilişkin tanımlamalar ve örneklerin bulunduğu rehber hazırlanmıştır. Yardımcı araştırmacı için hazırlanan rehber Ek 3'te gösterilmiştir.

Tablo 5

Yardımcı araştırmacıya gönderilen Excel tablosunun örneği

Veri	Canlandırma	Kukla Çeşitliliği	Kurgulama	Diğer Görüşler
Oyun yapınca daha iyisini yapmak istiyorsun, daha iyisini yapa yapa bunların en güzelini hayal edip, Scratch üzerinde canlandırmak istiyorsun.				1
Scratch'ta hayal ettiğimi mevcut kuklalar ile gerçekleştiriyorum.		1		
Boş vaktimde Scratch'ta yaptıklarım aklıma geliyor ve evde oturup tekrar yapıyorum. Kediye oynatmak mesela kuşları uçurmak.	1			
Oyunu tasarlama aşamasında hangi kuklalarla nasıl tasarlayacağım esnasında zihnimde düşünürken, hayal gücümü zorladı.			1	

Yardımcı araştırmacı verilerin kodlanması işlemini tamamladığında, araştırmacının güvenilirliğini ortaya koymak üzere Cohen Kappa katsayısından faydalanılmıştır. Kappa testi iki veya daha fazla gözlemci arasındaki uyumun güvenilirliğini ölçen parametrik olmayan bir istatistik yöntemidir. Uyumun karşılaştırıldığı değişken kategorik değişkendir (Cohen, 1960). Bu aşamada, araştırmacının kodları satırlara, yardımcı araştırmacının kodları sütunlara gelecek şekilde matris çıkartılmıştır. Bu işlemler Tablo 6, Tablo 7 ve Tablo 8’de gösterilmiştir.

Tablo 6

Hayal gücü teması altındaki kodların uyumu

Araştırmacı	Yardımcı Araştırmacı					Satır	Toplamları
<u>Kodlar</u>	<u>Canlandırma</u>	<u>Kukla</u>	<u>Çeşitliliği</u>	<u>Kurgulama</u>	<u>Diğer</u>	<u>Görüşler</u>	
Canlandırma	11			1			12
Kukla		10					10
Çeşitliliği							
Kurgulama				2	1		3
Diğer Görüşler				1	1		2
Sütun	11	10		4	2		27
Toplamları							

$$\sum a = 11 + 10 + 2 + 1 = 24$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{12 * 11}{27} = 4,88$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{10 * 10}{27} = 3,7$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{3 * 4}{27} = 0,44$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{2 * 2}{27} = 0,14$$

$$\sum ef = 4,88 + 3,7 + 0,44 + 0,14 = 9,16$$

$$K = \frac{\sum a - \sum ef}{N - \sum ef} = \frac{24 - 9,16}{27 - 9,16} = 0,83$$

Tablo 7

Scratch programının olumlu yönleri teması altındaki kodların uyumu

Araştırmacı	Yardımcı Araştırmacı						
<u>Kodlar</u>	<u>Eğlenceli</u>	<u>Özgüven</u>	<u>Oyun Programlama</u>	<u>Hayal Gücü</u>	<u>Problem Çözme</u>	<u>Diğer Görüşler</u>	<u>Satır Toplamları</u>
Eğlenceli	11		1				12
Özgüven		17	1	1		2	21
Oyun Programlama			6				7
Hayal Gücü				6			6
Problem Çözme					6		6
Diğer Görüşler					1	6	7
Sütun Toplamları	11	17	8	7	7	8	58

$$\sum a = 11 + 17 + 6 + 6 + 6 + 6 = 52$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{12 * 11}{58} = 2,27$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{21 * 17}{58} = 6,15$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{7 * 8}{58} = 0,96$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{6 * 7}{58} = 0,72$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{6 * 7}{58} = 0,72$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{7 * 8}{58} = 0,96$$

$$\sum ef = 2,27 + 6,15 + 0,96 + 0,72 + 0,72 + 0,96 = 11,78$$

$$K = \frac{\sum a - \sum ef}{N - \sum ef} = \frac{51 - 11,78}{58 - 11,78} = 0,84$$

Tablo 8

Scratch programının olumsuz ve eksik yönleri teması altındaki kodların uyumu

	Araştırmacı					Yardımcı Araştırmacı	
<u>Kodlar</u>	<u>Fantezi Kuklalar</u>	<u>Az Kılık</u>	<u>Az Dekor</u>	<u>Az Kukla</u>	<u>3D Eksikliği</u>	<u>Diğer Görüşler</u>	<u>Satır</u> <u>Toplamları</u>
Fantezi Kuklalar	4						4
Az Kılık		2					2
Az Dekor			3				3
Az Kukla				4			4
3D Eksikliği					3		3
Diğer Görüşler						6	6

Sütun	4	2	3	4	3	6	22
Toplamları							

$$\sum a = 4 + 2 + 3 + 4 + 3 + 6 = 22$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{4 * 4}{22} = 0.72$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{2 * 2}{22} = 0.18$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{3 * 3}{22} = 0.4$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{4 * 4}{22} = 0.72$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{3 * 3}{22} = 0.4$$

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam}} = \frac{6 * 6}{22} = 1.63$$

$$\sum ef = 0.72 + 0.18 + 0.4 + 0.72 + 0.4 + 1.63 = 4.05$$

$$K = \frac{\sum a - \sum ef}{N - \sum ef} = \frac{22 - 4.05}{22 - 4.05} = 1$$

Matrislerin çıkartılması işleminden sonra, Cohen'in kappa katsayısı hesaplanmıştır.

Her tablo bir ana temayı gösterecek şekilde hazırlanmıştır. Bu temalar, (1) Hayal gücü, (2) Scratch programının olumlu yönleri, (3) Scratch programının olumsuz ve eksik yönleri olarak sıralanmış ve yukarıdaki tablolarda gösterilmiştir. Koyu renkle verilmiş olan rakamlar eşleşme değerlerini göstermekte, eşleşmeyen durumlarda ise tablodaki ilgili yerlere araştırmacının ne düşündüğüne ilişkin rakamlar eklenmiştir. Sim ve Wright (2005)'a göre aşağıdaki adımlar uygulanarak Cohen Kappa katsayısı hesaplanabilir:

1. Her bir satır ve sütundaki sayılar toplanarak satır ve sütun toplamları hesaplanmıştır.
2. Tablodaki tüm sayılar toplanarak toplam cevapların sayısı (N) elde edilmiştir.
3. Tablolarda kalın harflerle verilen eşleşmeleri gösteren sayılar toplanmış ve $\sum a$ simgesi ile gösterilmiştir.
4. Tablolarda kalın harflerle verilen eşleşmeleri gösteren sayılara ilişkin değerler (ef) şu şekilde hesaplanmıştır:

$$ef = \frac{\text{satır toplamı} * \text{sütun toplamı}}{\text{toplam(N)}}$$

5. Daha sonra her bir kod için oluşan ef değerlerin toplamı hesaplanmıştır.

$$\sum ef = ef1 + ef2 + ef3 + \dots + efn$$

6. Cohen Kappa katsayısı (K) hesaplanırken kullanılan formül ise şu şekildedir:

$$K = \frac{\sum a - \sum ef}{N - \sum ef}$$

Elde edilen k değerini yorumlamak (Landis & Koch, 1977) için aşağıdaki tablo sunulmuştur.

K değeri	Yorum
<0	Şansa bağlı olabilecek uyumdan daha kötü uyum olması
0.01 – 0.20	Önemsiz düzeyde uyum olması
0.21 – 0.40	Zayıf düzeyde uyum olması
0.41 – 0.60	Orta düzeyde uyum olması
0.61 – 0.80	İyi düzeyde uyum olması
0.81 – 1.00	Çok iyi düzeyde uyum olması

Cohen Kappa katsayıları üç ana temada sırasıyla 0.83, 0.84 ve 1 çıkmıştır. Bu sonuçlara göre üç ana temada çok iyi düzeyde bir uyum söz konusudur. Nicel araştırma

kısımındaki ölçek pilot uygulama olarak, 7. sınıfa giden 100 öğrenci üzerinde gerçekleştirilmiş ve yapılan çalışmada güvenirlik katsayısı 0,83 olarak bulunmuştur.



4. Bölüm

Bulgular

Bir önceki bölümde verilerin toplanması ve veri toplama araçlarına ilişkin bilgilere yer verilmiştir. Bahsi geçen süreçte elde edilen nitel ve nicel veriler, bu bölümde 3 ana başlık altında incelenmektedir. Her bir başlık altında kodlara ilişkin bilgilere yer verilmiş, öğrencilerin görüşlerinden alıntılar yapılmış ve araştırmacının konuya ilişkin yorumlarına yer verilmiştir.

4.1. Problem Çözme Becerisi

Bu bölümde araştırma sorularından “Scratch ile programlama öğretiminin öğrencilerin problem çözme becerileri üzerinde anlamlı bir etkisi var mıdır?” sorusunun iki alt sorusuna yanıt aranmaktadır. Ortalamaları karşılaştırılacak veriler aynı gruptadır ve art arda yapılan iki ölçüm sonucu alınmıştır. Elimizde aynı öğrencilerin farklı zamanlarda yapılmış test sonuçlarının oluşturduğu veri çiftleri bulunmaktadır.

Tablo 9

Normallik testi

	Kolmogorov-Smirnov			Shapiro-Wilk		
	İstatistik	N	p	İstatistik	N	p
fark	,092	226	,000	,967	226	,000

Tablo 9’deki normallik testi için Kolmogorov-Smirnov testinin anlamlılık test sonucu $p < 0,05$ olduğu için farklar puan dizisi normal dağılmaktadır. Öğrencilerin uygulama öncesi puanları ile uygulama sonrası puanları arasında anlamlı bir fark olup olmadığını belirlemek için, parametrik olmayan bir test olan Wilcoxon İşaretli Sıralar Testi uygulanmıştır. Aşağıda yer alan Tablo 10 ve Tablo 11’de test sonuçları gösterilmiştir.

Tablo 10

Wilcoxon işaretli sıralar testi: Farklar puan dizisinin betimsel istatistikleri

Son test – Ön test Ölçümü	N	Sıra Ortalaması	Sıra Toplamı
Negatif Sıralar	66a	92,77	6123,00
Pozitif Sıralar	147b	113,39	16668,00
Fark Olmayan	13c		
Toplam	226		

a. son test < ön test

b. son test > ön test

c. son test = ön test

Tablo 10'daki farklar puan dizisinin betimsel istatistiklerine göre; 66 öğrencinin son test puanı ön test puanına göre düşük çıkmış, 147 öğrencinin son test puanı ön test puanına göre yüksek çıkmış, 13 öğrencinin ise ön test ve son test puanları eşit çıkmıştır.

Tablo 11

Wilcoxon işaretli sıralar testi anlamlılık test sonucu

	son test - ön test
Z	-5,859 ^b
p	,000

Tablo 11'deki Wilcoxon işaretli sıralar testinin sonucuna göre; öğrencilerin ön test ve son test puanları arasında istatistiksel olarak anlamlı bir fark gözlenmiştir ($Z = -5.859$, $p = .000 < .005$). Yani, araştırma sorularının altında yer alan hipotezlerde H_0 hipotezi reddedilir ve H_a hipotezi kabul edilir. Fark puanlarının pozitif sıralar lehine olması, Scratch ile yapılan programlama öğretiminin öğrencilerin problem çözme becerileri üzerinde anlamlı bir etkisinin olduğunu göstermektedir.

Tablo 12

Pozitif sıralar ön test ve son test puan ortalamaları betimsel istatistikleri

	N	Minimum	Maximum	Ortalama	Standart Sapma
ön test	147	1,0000	4,8000	3,577211	,6643369
son test	147	2,7000	5,0000	4,014286	,5208002

Tablo 12’de problem çözme becerisi puanları artış göstermiş öğrencilerin ön test ve son test puan ortalamalarının betimsel istatistikleri verilmiştir.

Tablo 13

Mann-Whitney U testi anlamlılık test sonucu: Cinsiyete göre

	fark
Mann-Whitney U	6031,500
Wilcoxon W	12026,500
Z	-,703
p	,482

Tablo 13’deki kız ve erkek öğrencilerin puan ortalamaları arasında fark olup olmadığını sınavan Mann-Whitney U testinin anlamlılık test sonucu $p=0,482>0,05$ olarak bulunmuştur. Araştırma sorularının altında yer verilen hipotezlerde H_0 hipotezi reddedilemez. Scratch ile programlama öğretiminde öğrencilerin problem çözme becerileri cinsiyete göre anlamlı bir farklılık göstermemektedir.

Tablo 14

Mann-Whitney U testi anlamlılık test sonucu: Sınıf seviyesine göre

	fark
Mann-Whitney U	5476,000

Wilcoxon W	12857,000
Z	-1,789
p	,074

Tablo 14'deki 5. ve 6. sınıf öğrencilerinin puan ortalamaları arasında fark olup olmadığını sınavan Mann-Whitney U testinin anlamlılık test sonucu $p=0,074>0,05$ olarak bulunmuştur. Araştırma sorularının altında yer verilen hipotezlerde H_0 hipotezi reddedilemez. Scratch ile programlama öğretiminde öğrencilerin problem çözme becerileri sınıf seviyesine göre anlamlı bir farklılık göstermemektedir.

4.1.1. Etki büyüklüğü. Etki büyüklüğü işaretinden bağımsız olarak değerlendirilir ve her değeri alabilir. Etki büyüklüğü değeri açısından, 1'in üzeri çok büyük, 0,8 büyük, 0,5 orta, 0,2 ise küçük az etki olarak değerlendirilir (Can, 2013). Bu çalışmada etki büyüklüğü aşağıdaki formül ile hesaplanmıştır.

$$d = \frac{Z}{\sqrt{N}} - 0,38972 = \frac{-5,858846}{15,0332964}$$

Etki büyüklüğü ($d=0,39$) değeri bize; Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerinde orta düzeyde bir etkisi olduğunu göstermektedir.

4.2. Yapılandırılmamış Problemlerin Çözüm Süreci

Bu bölümde, araştırma sorularından "Scratch ile programlama öğretiminde öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımları nelerdir?" sorusunun üç alt sorusuna yanıt aranmaktadır. Öğrencilerin problem çözme becerilerini ölçmek için kullandığımız ölçekteki 1-5 arası maddeleri temsil eden "Zor bir problemi

çözmeye başlamadan önce ne yaparsın?” sorusu yorumlama ve problemi tanımlama, 6-10 arası maddeleri temsil eden “Problem üzerinde çalışırken ne yaparsın?” sorusu çözümler geliştirme ve çözüm sürecini izleme, 11-15 arası maddeleri temsil eden “Problem üzerinde çalışmayı bitirdikten sonra ne yaparsın?” sorusu karar verme ve çözüm sürecini değerlendirme aşaması ile ilgilidir.

4.2.1. Yorumlama ve problemi tanımlama. Yorumlama ve problemi tanımlama aşamasında öğrencilerin hangi işlem adımlarını uyguladıklarını öğrenmek için birinci görüşme sorusu sorulmuştur. Yorumlama ve problemi tanımlama aşamasında öğrencilerin uyguladıkları işlem adımlarının frekansları aşağıda belirtilmiştir.

Tablo 15

Yorumlama ve problemi tanımlama

Problem Çözme Adımları	Frekans (n=27)
Kurgulama	20
Tasarlama	7
Diğer görüşler	2

4.2.1.1. Kurgulama. Tablo 15’deki frekanslara göre öğrencilerin (n=20) oyunu zihinlerinde gerçek hayatla ilişki kurarak kurguladıkları ortaya çıkmıştır. Aşağıda öğrenci görüşlerinden bazılarına yer verilmiştir.

Duygu, “Kafamda oyunu tasarlıyorum, kurguluyorum. Örnek verirse, biz yukarıya nasıl zıplıyoruz gibi.” ifadesiyle oyundaki işlem adımlarını zihninde gerçek hayatı referans alarak kurguladığını belirtmiştir. Ahmet, “Oyunun nasıl bir şey olabileceğini düşünürüm. Kafamın içinde canlandırıyorum. Scratch kuklalarıyla kurgularım. Gerçek hayatla ilişki kurarım.” ifadesiyle Duygunun sözlerini teyit ederek, oyunu zihninde Scratch kuklalarıyla gerçek hayatı referans alarak kurguladığını belirtmiştir. Zülal ise, “Probleme odaklanırım.

Birkaç kez oyunu oynarım, oyunu oynayınca aklıma fikir gelir. O an gerekli olan bilgileri hatırlıyorum. Oyunu oynarken problemi birkaç kez analiz ederim. Kod bloklarının dizilişini tasarlıyorum.” ifadesiyle zihninde oyunu kurgulayarak kod bloklarının dizilişini tasarlamaktadır.

Burçak, oyunu zihninde kurgulamadan direkt uygulamaya yönelik kod bloklarını birleştirmeye geçtiğinde zorlanacağını şu sözlerle ifade etmiştir: “Düşünmeden direkt kod bloklarına geçersen karışık olur ve yapmam zorlaşır. Farklı çözüm yollarını kafamda topluyorum. Bu konu hakkında bilgilerimi kafamda düşünürüm. Problemden yapmam gerekenleri tasarlarım.” Kardelen, “Projeyi aklımda tasarlıyorum, yapacaklarımı planlıyorum, senaryoyu kurguluyorum, kuklanın yapacaklarını düşünüyorum.” ifadesiyle oyunu zihninde kuklaların eylemine göre kurguladığını belirtmiştir.

İbrahim, “Düşünüyorum ne yapacağımı kafamda tasarlıyorum. Örneğin konu zıplamak ise zıplayınca x mi y mi artacak onları düşünüyorum.” ifadesiyle oyunu zihninde kurgular iken matematiksel ifadeler ile eşleştirme yapmaktadır. Scratch aynı zamanda matematiksel ifadelerin öğrenilmesini de sağlamaktadır. Muhammet, “Oyunu zihnimde kuklaların özelliklerine göre tasarlarım, örnek verirsem zıplama nasıl olacak diye düşünürüm.” söylemiyle İbrahim’in sözlerini teyit etmektedir. Örnek verirsek, Mario kuklasının yukarı ok tuşuna basılınca zıplaması gerektiği ve hangi eksendeki değerinin zamanla değişeceğini düşünmesi gerekmektedir.

4.2.1.2. Tasarlama. Tablo 15’deki frekanslara göre öğrencilerin (n=7) zihinlerinde kurguladıkları oyunun tasarımını eylemlerin gerçekleşme sırasına göre metne döktükleri görülmektedir. Aşağıda öğrenci görüşlerinden bazılarına yer verilmiştir.

Emir, “Scratch yorum ekle kısmına oyunu tasarlıyorum.” ifadesiyle yorum ekle kısmına açıklama ekleyerek tasarımını metne döktüğünü belirtmiştir. Nisanur, “Sorunu çözmek için ne yapmam gerektiğini maddeler halinde sıralıyorum. Oyunu tasarlamadan hangi

kod bloklarını kullanacağımızı bilemeyiz.” söylemiyle tasarıma uygun eylem adımlarını işlem sırasına göre metne döktüğünü belirtmiştir. Beyza, “Açıklama ekle kısmına kod bloklarına göre senaryoyu yazıyorum.” ifadesiyle kod bloklarını baz alarak oyunun tasarımını metne dökmemektedir.

Melike ise “İlk önce hayalimde nasıl yapabileceğimi düşünüp, planlayıp resme döküyorum. Not defterime oyunu tasarlıyorum.” ifadesiyle Scratch programının görsel zekasını kullanmaya teşvik ettiğini ifade etmiştir. Büşra da “Hangi kod bloklarını kullanarak yapabilirim diye düşünüyorum. Kâğıt üzerinde senaryoyu çiziyorum.” ifadesiyle Melike’nin görüşünü desteklemektedir.

4.2.2. Çözümler geliştirme ve çözüm sürecini izleme. Çözümler geliştirme ve çözüm sürecini izleme aşamasında öğrencilerin hangi işlem adımlarını uyguladıklarını öğrenmek için ikinci görüşme sorusu yöneltilmiştir. Bu aşamada öğrencilerin uyguladıkları işlem adımlarının frekansları aşağıda belirtilmiştir.

Tablo 16

Çözümler geliştirme ve çözüm sürecini izleme

Problem Çözme Adımları	Frekans (n=27)
Kod bloklarını tasarıma göre uygulama	26
Diğer görüşler	1

Tablo 16’deki frekanslara göre öğrenciler (n=26), kod bloklarını yerleştirirken oyunun tasarımını referans aldıklarını ifade etmişlerdir.

4.2.2.1. Kod bloklarını tasarıma göre uygulama. Aşağıdaki ifadelerden de anlaşılacağı üzere öğrencilerin yapılandırılmamış problemlerin çözümüne yönelik olarak tasarım odaklı yaklaşması gerekmektedir.

Duygu, “Kod bloklarını kafamda canlandırdığım oyunun tasarımına göre yerleştiriyorum.” ifadesiyle, Sıla’nın “Sahnedeki eylem sırasına göre kod bloklarını yerleştiririm.” ifadesi birbirini desteklemektedir. Ahmet, “Kuklaların eylemine göre kod bloklarını belirliyorum, kod bloklarını oyunun sırasına göre yerleştiriyorum.” ifadesiyle Duygu ve Sıla’nın ifadelerine katılmaktadır. Muhammet sözleriyle eylemlerin gerçekleşme sırasına dikkat çekmektedir: “Kod bloklarını tasarımın sırasına göre yerleştiririm. Adım adım gerçekleşecek eylemlere uygun kod bloklarını yerleştiririm.”

Burçak, “Kod bloklarını inceleyip işime yarayacak olanları seçerim. Başlangıç kod bloğunu seçerim. İlk önce kontrol menüsünden hangi kod bloğunu ekleyeceğimi seçerim. Diğer kod bloklarını da sahneye sürükleyip bırakarak, en son kod bloklarının sıralamasını tasarıma göre yerleştiririm.” ifadesiyle kontrol menüsündeki kompüstasyonel ifadeler karşılık gelen kod bloklarının önemine dikkat çekmiştir. Ayrıca kod bloklarının sıralamasını tasarıma göre uygulamaktadır. Emir ise, “Kod bloklarını yorum ekle kısmına yazdığım tasarıma göre yerleştiriyorum.” ifadesiyle metne döktüğü zihinsel tasarım sürecine göre kod bloklarının dizilimini ayarlamaktadır.

4.2.3. Karar verme ve çözüm sürecini değerlendirme. Karar verme ve çözüm sürecini değerlendirme aşamasında öğrencilerin hangi işlem adımlarını uyguladıklarını öğrenmek için üçüncü görüşme sorusu yöneltilmiştir. Karar verme ve çözüm sürecini değerlendirme aşamasında öğrencilerin uyguladıkları işlem adımlarının frekansları aşağıda belirtilmiştir.

Tablo 17

Karar verme ve çözüm sürecini değerlendirme

Problem Çözme Adımları	Frekans (n=27)
Programı çalıştırma ve bir sonraki aşamaya geçme	27

Tasarıma göre kod bloklarını kontrol etme	20
Tasarımı kontrol etme	3
Diğer görüşler	2

Tablo 17’deki frekanslara baktığımızda öğrencilerin (n=27) hepsi programı çalıştırıp hata gelmezse bir sonraki aşamaya geçtiklerini ifade etmişlerdir. Hata durumunda, oyunun tasarımına göre (n=20) kod bloklarının yerleşim sırasını kontrol ettiklerini belirtmişlerdir. Buna karşılık üç öğrenci hatayı bulmak için oyunun tasarımını kontrol ettiklerini belirtmişlerdir.

4.2.3.1 Tasarıma göre kod bloklarını kontrol etme. Sıla ifadesiyle hata durumunda kod bloklarının dizilişini tasarıma göre kontrol etmektedir: “Olmazsa sahnedeki olaylar zincirine göre tekrar denerim. Hangi kod bloğu eksikse onu uygularım.” Nergis’ de aynı şekilde, “Hata var ise tasarıma uygun kod bloklarını doğru yerleştirip yerleştirmedigimi kontrol ediyorum.” ifadesiyle tasarıma göre kod bloklarını kontrol etmektedir.

Cemre, “Çalışmazsa büyük ihtimal oyunu doğru tasarlamışım ama kod bloklarını yanlış yerleştirmemden kaynaklanan bir hatadan kaynaklanmıştır. Genellikle kontrol menüsündeki kod bloklarının seçiminde zorlanıyorum.” ifadesiyle tasarımına güvenmektedir ancak tasarımı Scratch kontrol menüsündeki kompütasyonel kod bloklarına uygun olarak detaylandırmadığından doğru kod bloğunu yerleştirememektedir. Muhammed, “Çalışmadıysa senaryoya göre kod bloklarını doğru yerleştirip yerleştirmedigimi kontrol ediyorum. Genelde senaryoyu doğru ama kod bloklarını senaryoya göre yanlış yerleştiriyorum. Tasarıma uygun olarak kontrol dizisindeki koşullu ifadelerin seçiminde zorlanıyorum.” ifadesiyle tasarımı kontrol menüsündeki kompütasyonel ifadelerle (kod bloklarına) uygun olarak detaylandırmadığından dolayı kod bloklarını yanlış yerleştirmektedir. Burçak çözümler geliştirme ve çözüm sürecini izleme aşamasında, “İlk önce kontrol menüsünden hangi kod

bloğunu ekleyeceğimi seçerim.” ifadesiyle kontrol menüsündeki kompütasyonel kod bloklarının önemine dikkat çekmiştir ve Burçak projeleri tam anlamıyla eksiksiz olarak zamanında bitiren nadir öğrencilerdendir. Oyun projelerini tam anlamıyla bitirmek için öğrencilerin tasarımlarını kontrol menüsündeki kompütasyonel ifadelerle uygun olarak detaylandırmaları gerekmektedir.

Gülşen ise, “Senaryoyu doğru yapıyorum ama tasarıma uygun kod bloğunu doğru yerleştiremiyorum. Kod bloklarının içini doğru yerleştirmede hata yapıyorum. x ve y eksenlerini birbirine karıştırıyorum.” diyerek yatay ve dikey eksenleri karıştırdığını belirtmiştir.

4.3. Problem Çözme, Programlama ve Özgün Ürün Geliştirme Alanındaki Yeterlikler

Bu bölümde araştırma sorularından “Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanındaki yeterliklere ilişkin öğrenci görüşleri nelerdir?” sorusunun dört alt sorusuna yanıt aranmaktadır.

4.3.1. Özel stratejiler. Özel stratejiler boyutunda öğrencilerin hangi işlem adımlarını uyguladıklarını öğrenmek için “Scratch’ ta problemler üzerinde hangi yöntemi (strateji, yol) uygulayarak çalışıyorsun?” görüşme sorusu 27 öğrenciye yöneltilmiştir. Özel stratejiler boyutunda öğrencilerin uyguladıkları işlem adımlarının frekansları aşağıda belirtilmiştir.

Tablo 18

Özel stratejiler

Problem Çözme Adımları	Frekans (n=27)
Kurgulama	19
Kod bloklarını tasarıma göre uygulama	12
Arkadaşı ile işbirliğine gitme	3
Diğer görüşler	4

Tablo 18’i incelediğimizde öğrencilerin büyük çoğunluğu, oyunu zihinlerinde kurgulayıp tasarladıklarını ve bu tasarıma uygun kod bloklarını Scratch programında yerleştirdiklerini ifade etmişlerdir. Üç öğrenci de problemin çözüm sürecinde arkadaşı ile işbirliğine gittiğini belirtmiştir. Aşağıda bazı öğrenci görüşlerine yer verilmiştir.

Zülal kendi ifadesiyle zihinsel tasarım sürecini ifade etmiştir: “İlk baş, ne yapacağımı düşünürüm. Puan mı alacak mesela puan bir değişkendir. Değişkene tıklayıp değişken oluştururum. Puanı nasıl alacağını tasarlarım. Oyunu tasarlarım, oyun adımlarını tek tek planlarım. Tasarıma uygun kod bloklarını bulup tek tek sıralarım. Problemi birkaç kez gözümün önünde analiz ediyorum.” Efe, “Oyunu ilk önce analiz ederim. Oyunu tasarlarım, bir sonraki aşamayı kafamda kurgularım.” ifadesiyle oyun sürecini adım adım zihninde kurguladığını belirtmiştir.

Başarılı öğrencilerimizden Burçak, kendi ifadesiyle bilgi işlemsel düşünme sürecini çok iyi tanımlamıştır: “Kafamda problemi şema haline getiriyorum. Bu benim daha rahat tanımlamamı sağlıyor. Benim yaptığım uygulamada kaplumbağa boşlukta yürüyor. İki nokta arasında gidip gelmesi gerekiyor. Kaplumbağanın ne yapması gerektiğini düşünüyorum. Şemanın bir kutusuna kaplumbağanın iki nokta arasında yürümesini istiyorum. İkinci kutuya ise bunu hangi kod blokları ile yapabileceğimi yerleştiriyorum. Üçüncü kutuya bunların arasındaki bağlantıyı düşünüyorum.”

Melike kendi ifadesiyle hayal gücünün önemine değinmiştir: “Hayal gücüm güçlü olduğundan dolayı senaryoyu güzel tasarlayabiliyorum.” Gülşen kendi ifadesiyle gerçek hayatla bağlantı kurmanın önemine değinmiştir: “Bütün kod bloklarını denerim, hiç pes etmem. Hangi oyunu yaparsam onu gerçek hayatta yaşadırısam yapmam daha kolaylaşıyor. Oyun ile gerçek hayat arasında bağlantı kurarım.” Muhammet, “İlk önce problemi tanımlarım, problemi gerçek hayatta nasıl çözüyorsak oyunu tasarlarırken de aynı şekilde çözmemiz

gerekir.” ifadesiyle yapılandırılmamış problem çözme sürecinde gerçek hayat ile ilişki kurmanın önemine değinmiştir.

Problem çözme becerisi ölçeğinin sonunda öğrencilerden genel bir yanıt almak amacıyla “Scratch ile oyun programlamanın size ne gibi katkıları olmuştur? Olumlu olarak gördüğünüz noktaları yazınız.” sorusu sorularak öğrencilerin metin kutusuna açık uçlu cevap bırakmaları sağlanmıştır. Aşağıdaki tabloda verilen cevapların betimsel analizi yapılarak frekansları ortaya çıkarılmıştır.

Tablo 19

Scratch ile oyun programlamanın faydaları

Temalar	Frekans (n=95)
Problem Çözme Becerisi	50
Hayal Gücü	30
Görsel Programlama	20
Resim	3

Betimsel analizdeki maddeleri incelediğimizde, ortaya çıkan temalar Scratch programı ile ilgili literatürdeki araştırmalardaki temalar ile birebir örtüşmektedir. Frekansı en yüksek madde olan problem çözme becerisi nicel ve nitel verilerle detaylıca araştırılmıştır. Frekansı en yüksek ikinci madde olan hayal gücüne ilişkin öğrenci görüşleri; problem çözme, programlama ve özgün ürün geliştirme alanındaki yeterlikler kapsamında aşağıda içerik analizi ile detaylandırılmıştır.

4.3.2. Hayal gücü. Öğrencilerin, Scratch programının hayal güçlerine katkısına ilişkin görüşlerini belirlemek için “Hayal gücünüzde gelişme olduysa, bu duruma uygun örnek verebilir misiniz?” görüşme sorusu 27 öğrenciye yöneltilmiştir. Verilen yanıtlara göre aşağıdaki tablodaki kodlar içerik analizi ile oluşturulmuştur.

Tablo 20

Hayal gücü

Kodlar	Frekans (n=27)
Canlandırma	12
Kukla Çeşitliliği	10
Kurgulama	3
Diğer Görüşler	2

4.3.2.1. Canlandırma. Öğrenciler (n=12) Scratch ile oyun tasarımında; kuklaları Scratch sahnesinde kendi hayallerine göre canlandırmalarının ve fantezi kuklalar ile fantezi eylemlerin gerçekleştirilebilmesinin hayal güçlerini olumlu yönde etkilediğini belirtmişlerdir.

Hakan, “Scratch biraz da hayal kurmakla ilgili, aklınıza getirdiğiniz bir hayvanı Scratch programında hareket ettirebilirsiniz. Bir yarasa yapıp onu uçurabiliriz.” ifadesiyle Scratch programında oyun tasarlamının hayal kurmakla ilişkili olduğunu belirterek, istediği hayvan kuklasının günlük yaşamdaki eylemini Scratch sahnesinde canlandırabileceğini ifade etmiştir. Göktuğ da Hakan ile aynı fikirde olduğunu kendi ifadesiyle belirtmiştir: “Hayal gücüme göre kuklaları kullanarak istediğimi yapabilirim. İstedığımız dekoru ve kuklaları kullanabiliriz. Ejderhayı çölde uçurabiliriz mesela.”

Sıla ise hayal gücünün gelişimine bir başka açıdan yaklaşarak, “Hikâyelerde veya romanlarda resimler yok ama biz Scratch ile çalıştığımızdan beri kuklaları kullanarak bir hikâye veya romanı zihnimize canlandırabiliyoruz. Kuklaları canlandırmak hayal gücümü geliştirdi.” ifadesiyle Scratch programının hayal gücünün günlük hayattaki gelişimine etkisini, kuklaları canlandırmanın katkısına ilişkin güzel bir örnek vererek ifade etmiştir.

Mustafa, “Normal hayatta uçamayız ama Pikachu oyununda Pikachu kuklasını uçurabiliyoruz. Hayalperest oluyoruz, hayallere dalıyoruz çünkü sıradan olmayan şeyler

yapabiliyoruz.” görüşüyle gerçek hayatta yapılması mümkün olmayan eylemlerin Scratch sahnesinde gerçekleştirilmesinin hayal gücünün gelişimine katkısı olduğunu belirtmiş, bu tür fantezi ve sıra dışı eylemlerin sık sık hayal kurmasını sağladığını ifade etmiştir. Mücahit de Mustafa’nın görüşüne katılarak, “Mesela hayvanları da orada konuşturabiliriz, ejderha kuklası var mesela, hayal gücümüzdekini Scratch sahnesinde uygulatabiliriz kuklalarımıza.” ifadesi göstermiştir ki fantezi kuklalar ile fantezi eylemlerin gerçekleştirilebilmesi hayal gücünü olumlu yönde etkilemiştir.

Kağan, “Pikachu kuklasını uçurmak hayal gücümü geliştirdi. Scratch programına hayal gücümü yansıtabiliyorum.” ifadesiyle fantezi eylemlerin Scratch sahnesinde gerçekleştirilmesinin hayal gücünün gelişimini pozitif yönde etkilediğini belirtmiştir. Benzer şekilde İbrahim, “İstediğim şeyi yapma imkânı sağladığı için hayal gücüm gelişti, örnek adamı uçurabiliyorum. Fantezik şeyler yapmamıza imkân sağlıyor.” ifadesiyle Scratch programının fantezi eylemlerin gerçekleştirilebilmesine imkân sağladığını belirtmiştir.

4.3.2.2. Kukla çeşitliliği. Scratch programında oyun tasarımında kullanılan kuklaların; zenginliği, eğlenceli ve fantezi oluşunun öğrencilerin hayal gücünün gelişiminde pozitif yönde etki yaptığını aşağıdaki öğrenci görüşleri (n=10) desteklemektedir.

Büşra, “Hayali şeyler düşünmemi Scratch programındaki kuklalar etkiledi.” ifadesiyle kukla çeşitliliğine vurgu yapmıştır. Melike, “Eğlenceli kuklaları hayal gücümün gelişmesine sebep oldu.” ifadesiyle kuklaların tasarımını eğlenceli bulduğunu ifade etmiştir.

Zülal, “Çok garip kuklalar vardı, beni etkilemiştir bu.” ifadesiyle fantezi kuklaların tasarımı onun hayal dünyasını etkilemiştir. Gürkan da benzer şekilde, “Scratch’ta ki ejderha gibi fantezi kuklaları kullanarak diğer kuklalarla sahnede konuşturabiliriz. Kuklalar benim hayal dünyamı canlandırdı.” ifadesi de göstermektedir ki öğrencilerin birçoğu fantezi ejderha kuklasından etkilenmiştir.

Damla, “Verilen konuyu hayal gücümüze göre yaptık, basit bir şekilde olmadı. Daha farklı ve daha güzel oldu. Kuklaları canlandırarak oyunu çeşitli kuklalarla daha renkli ve ilgi çekici hale getirebiliyoruz.” ifadesiyle hayal gücünü Scratch sahnesine kuklaların çeşitliliği ile yansıttığını belirtmiştir. Muhammet’te ifadesiyle Damla’nın görüşünü desteklemektedir: “Pikachu oyunda normalde düz gidebilirdi ama onu yukarıya zıplattık, aşağı yapacaktık ama yapmadık çünkü kendi kendine aşağıya inecekti, canavarı ekledik ama onu koymayabilirdik, yani kuklalarla oyunu zenginleştirmek hayal gücümü geliştirdi.”

4.3.2.3. Kurgulama. Öğrenciler (n=3) oyun tasarım sürecinin zihinde kurgulanmasının hayal güçlerini olumlu yönde etkilediğini belirtmişlerdir. Aşağıda buna ilişkin öğrenci görüşlerine yer verilmiştir.

Alper, “Oyun yapınca daha iyisini yapmak istiyorsun, daha iyisini yapa yapa bunların en güzelini hayal edip, Scratch üzerinde canlandırmak istiyorsun.” sözleriyle daha güzel bir oyunu ortaya koyma isteğinin arttığını belirterek zihinsel kurgulama sürecine değinmiştir. Cemre de “Oyunu tasarlama aşamasında hangi kuklalarla nasıl tasarlayacağım esnasında zihnimde düşünürken, hayal gücümü zorladı.” sözleriyle Alper’in görüşünü desteklemektedir. Zülal, “Kurgulama, kuklalarla oyunu nasıl yapabilirim. Çok garip kuklalar vardı, beni etkilemiştir bu.” sözleriyle kurgulama sürecinde kuklaların önemine değinmiştir.

Frekansı en yüksek üçüncü madde olan görsel programlama ile ilgili Scratch programına ilişkin öğrenci görüşleri; problem çözme, programlama ve özgün ürün geliştirme alanındaki yeterlikler kapsamında aşağıda içerik analizi ile detaylandırılmıştır.

4.3.3. Scratch programının olumlu yönleri. Öğrenciler açısından, Scratch programının olumlu yönlerini belirlemek için “Scratch programının hoşunuza giden yönleri nelerdir?” görüşme sorusu 27 öğrenciye yönlendirilmiştir. Verilen yanıtlara göre aşağıdaki tablodaki kodlar içerik analizi ile oluşturulmuştur.

Scratch programının olumlu yönleri

Temalar	Kodlar	Frekans (n=27)
Psikolojik Boyut	Eğlenceli	12
	Özgüven	21
Beceri Boyutu	Hayal Gücü	6
	Problem Çözme	6
	Programlama	6
Diğer Görüşler		7

Tablo 21’deki frekanslara göre öğrenciler Scratch programını eğlenceli bulmaktadırlar, oyunun görsel olarak programlanmasında karşılarına çıkan problemleri çözdüklerinde başarı hissi duymaktadırlar ve bu da özgüvenlerinin artmasına sebep olmaktadır. Kendi oyunlarını yaptıkları için gurur duymaktadırlar. Beceri boyutunda ise programlama becerilerinin, problem çözme becerilerinin ve hayal güçlerinin geliştiklerini ifade etmişlerdir.

4.3.3.1. Psikolojik boyut. Hakan, “Oyun yapmak ve onu oynamak eğlenceli geliyor.” ifadesiyle oyun programlamayı eğlenceli bulmaktadır. Melike’de, “Film yapmak, oyun yapmak çok eğlenceli. Resim yapmayı çok seviyorum, bazı kuklaları kendim çiziyorum. Komik ve korkunç kuklaları canlandırmak eğlenceli.” sözleriyle canlandırmanın önemine dikkat çekmiştir. Kağan’da Melike’nin sözlerine katılarak, “Oyun yapmak eğlenceli. Güzel bir kuklayı iyi yönde kullanabiliyoruz, onu canlandırabiliyoruz. Kuklayı canlandırmak güzel çünkü çok uğraşıyoruz ve sonucunda istediğimizin olmasını görmek mutlu ediyor.” sözleriyle oyun programlama sürecini başarı ile neticelendirmeyi eğlenceli bulmaktadır. Muhammet, “Hayalimizdeki düşüncelerimizi Scratch’ta oluşturabiliyoruz, bu eğlenceliydi. Kendi oyunumuzu oluşturduğumuzda eğleniyoruz, çünkü kendi oyunumuz biz onu istiyoruz.”

sözleriyle Scratch programına hayal gücünü yansıtabilmeyi ve kendine oyununu üretmeyi eğlenceli bulmaktadır. Zülal ise, “Mantık yürütmemi geliştirirken aynı zamanda oyun yaparak eğleniyoruz.” sözleriyle düşünme sürecinde eğlendiğini ifade etmiştir.

Beyzanur, “Benim yaptığım oyunu diğer arkadaşlarımın da oynaması hoşuma gitti.” ifadesiyle kendi oyununu arkadaşlarının da oynamasından gurur duymaktadır ve sosyal ortamda Scratch programında başarılı öğrenciler sınıf içi gözlemlerime dayanarak söyleyebilirim ki ön plana çıkmaktadır. Hilal, “Bilmeyen arkadaşlarıma yardım etmek hoşuma gidiyor. Kendim yaptığımda kendimi zeki hissettim. Kendimle gurur duymama sağladı. Belki de başkaları benim oyunumu oynayacak, onların da hoşuna gidecek.” sözleriyle arkadaşlarına yardım etmekten ve oyun üretmekten gurur duymaktadır. Nisanur ise farklı bir bakış açısından yaklaşarak, “Kuşu uçurmak, balığı yüzdürmek güzeldi. Kendi oyunumu yapınca gurur duydum, kendimi yönetmen gibi hissettim. Oyun yapmak film çekmek gibi.” sözleriyle gerçek hayattan bir rol ile kendini eşleştirmiştir.

Alper, “Oyun yapınca kendime güvenim artıyor, arkadaşlarımın hoşuna gidiyor, başarabileceğime inanıyorum.” sözleriyle Scratch programında başarılı olmanın aynı zamanda sosyal ortamda kendisini de ön plana çıkardığını ifade etmiştir. Hakan, “Oyun yapmak çok fazla düşünmemizi sağlıyor. Kendime olan özgüvenim arttı.” sözleriyle problem çözme sürecinin kendisine olan özgüveni arttırdığını ifade etmiştir. Mert, “Kuklaları hareket ettirebiliyorum, kendi sevdiğim oyunları yapabilmek hoşuma gidiyor. Kendi oyunumu yapınca kendime olan güvenim artıyor.” ifadesiyle üretebilmenin özgüvenini arttırdığını ifade etmiştir.

Melike, “Kendi oyunumu yaparak başardım ve başarabilme zevkini hissettim.” ifadesiyle, problem çözme sürecinde önüne çıkan problemleri çözerek sonuca ulaşmasıyla başarabilme hazzına ulaşmıştır. Damla, “Kendi oyunumu yapmak benim için ileride daha güzel büyük oyunlar yapabileceğimi düşünmeme sebep oldu.” sözleriyle başarının daha

büyük başarılarla sebep olacağını belirtmiştir. Öğrencinin bu düşüncede olmasının sebebi olarak, Scratch programının yapılandırılmamış problemleri çözme ortamı sağlaması gösterilebilir.

Kağan, “Oyunu bitirmek için vazgeçmedim, başarmak istedim. Arkadaşımdan önce bitirince sınıfta popülaritem artıyor. Herkes beni alkışlar.” sözleriyle Scratch programında başarılı olmanın öğrencinin sınıftaki popülaritesini arttırdığını belirtmiştir. Bu da aslında bize Scratch programında oyun programlama sürecinin her öğrencinin kolaylıkla tamamlayabileceği bir süreç olmadığını göstermektedir. Aynı zamanda öğrencinin içsel motivasyonunu arttıran bir faktör olarak ortaya çıkmaktadır.

4.3.3.2. Beceri boyutu. Mücahit, “Oyun yapabilmek hoşuma gitti. Çocukların bu yaşlarda en çok sevdiği şeylerdir. Eğitici oyunlar yapmak hoşuma gitti, çünkü bu benim gelişimime de katkı sağladı. Kuklaların fazla oluşu sebebiyle değişik şeyler yapabiliriz.” sözleriyle Scratch programında oyun programlamanın programlama becerisinin gelişimine katkısını belirtmiştir. Zülal, “Birçok işlemi yapabiliyoruz, istediğimizi kod bloklarıyla yapabiliyoruz. Basitçe programlama yapma imkanını bize sağlıyor. Yapamadığın şeyleri bilgisayarda yapmak güzel, bir sürü kuş çeşidini oyunda uçurabiliyoruz.” sözleriyle programlama becerisinin gelişiminde kuklaların önemine değinmiş ve Scratch programının basit yoldan programlamaya giriş yapabilme imkanını verdiğini belirtmiştir. Tunahan, “Farklı kuklalar ile yaratıcı oyunlar yapmak hoşuma gidiyor. Dans oyununda kuklaları konuşturmak hoşuma gitti.” sözleriyle gerçek hayattaki eylemlerin oyun tasarım sürecinde gerçekleştirilmesinin programlama sürecinde kendisini motive ettiğini belirtmiştir. Gürkan, “Kendi animasyon ve oyunlarımızı yapıyoruz, kendi oyunumu yapmak güzel ve nasıl yapıldığını öğreniyoruz. Kuklaları kod blokları ile canlandırmak kolay ve güzel.” sözleriyle oyun programlama sürecinde kuklaların kod blokları ile canlandırılmasını kolay ve ilgi çekici bulmaktadır.

Damla, “Daha fazla yaratıcı olmamızı sağlıyor, bir probleme birden fazla çözüm bulmak için zihnimizi zorluyoruz, beyin fırtınası yapmak zorunda kalıyoruz.” sözleriyle problem çözme sürecinde birden fazla çözüm yolu bulmak için sarf ettiği zihinsel sürecin problem çözme becerisini pozitif yönde etkilediğini ifade etmiştir. Zülal, zihinsel tasarım sürecini kendi ifadesiyle şöyle açıklamıştır:

“Ne yapacağımı düşünüp, çözüm arayıp, sorgulayıp, uygulama becerimi geliştirdi. Kurgumu yapana kadar uğraşmak hoşuma gidiyor. Scratch programında beyin sistematik bir şekilde düşünmeyi öğreniyor, sorgulamayı öğreniyor. Matematik dersindeki x ve y koordinat eksenini Scratch programında mevcut, Türkçe dersindeki hikâye sorgulama stratejik düşünme ile ilgili ama sadece sözcüklere bakıyoruz, Scratch programında da mevcut ama Scratch programında her şeye bakıyoruz.”

4.3.4. Scratch programının olumsuz ve eksik yönleri. Öğrenciler açısından, Scratch programının olumsuz ve eksik yönlerini belirlemek için “Scratch programının hoşunuza gitmeyen ve size göre eksik yönleri nelerdir?” görüşme sorusunu 27 öğrenciye sorduk. Verilen yanıtlara göre aşağıdaki tablodaki kodlar içerik analizi ile oluşturulmuştur.

Tablo 22

Scratch programının olumsuz ve eksik yönleri

Temalar	Kodlar	Frekans (n=27)
Kuklalar	Az Kukla	4
	Az Dekor	3
	Az Kılık	2
	Fantezi Kuklalar	4
Arayüz Tasarımı	3D Eksikliği	3
Nötr		13

Tablo 22' deki frekanslara göre öğrenciler (n=13) Scratch programının olumsuz ve eksik bir yönünün olmadığını ifade etmiştir. Aşağıda öğrenci görüşlerinden bazılarına yer verilmiştir.

Beyzanur, “Bazı kuklaların bir adet kılığı var ve o kuklaya kılık değiştirme veremiyoruz bir adet olduğu için, kendimizin çizmesi de zaman alacak. Sahneye ekleyebileceğimiz daha fazla dekor da olsa iyi olurdu.” ifadesiyle tek kılığa sahip kuklaları sahnede iki boyutlu olarak canlandırma imkanının olmadığını belirtmiştir. Mücahit: “Fantezi kuklaların olması bana değişik geliyor, daha çok kültürümüze uygun normal kuklalar olabilirdi.” ifadesiyle fantezi kuklaları yadsımıştır. Cemre: “Yaratıcılığımızın gelişmesi için daha fazla kukla olabilirdi, üç boyutlu olabilirdi.” ifadesiyle kuklaların derinliğinin olmasının yaratıcılık becerisinin gelişimi ile pozitif yönde ilişkilendirmiştir. Zülal ise aşağıdaki ifadesiyle dizilerin tasarımında açıklama bulunmasının kullanım kolaylığı ile ilişkilendirmiştir.

“Kod bloklarının üstüne fare imlecini götürdüğümüzde, o kod bloğunun ne işe yaradığı konusunda açıklama kutusu çıksa güzel olurdu. Sürekli çalıştığımız için, onu göre göre beynimiz ezberler. Menülerin yani dizilerin üzerinde açıklama olsa güzel olurdu, örnek verirsek algılama menüsünde hangi kod blokları görünür gibi.

Kuklaların biraz daha fazla kılıkları olsa iyi olur.”

Sinan: “Kuklalara yürüme efekti verilmesi için tekrarlı yapıya sahip tek bir kod bloğunun olması.” ifadesiyle hazır fonksiyonların da dizilere eklenmesinin kullanımı kolaylaştıracağını ifade etmiştir.

5. Bölüm

Tartışma ve Öneriler

Problem çözme becerisine ilişkin bulgulardan ve öğrencilerin Scratch ile programlama öğretimine ilişkin görüşlerinden yola çıkarak elde edilen verilerin çözümlenmesiyle ortaya çıkan sonuçlara bu bölümde yer verilmiştir. Aşağıdaki alt başlıklarda araştırmadan elde edilen bulgular ile literatürden elde edilenler karşılaştırılarak çıkarımlarda bulunmaktadır.

5.1. Problem Çözme Becerisi

Kalelioğlu ve Gülbahar (2014)'ın araştırma sonuçlarına göre, öğrencilerin problem çözme becerilerinin puan ortalamaları arasında anlamlı bir farklılık olmadığı bulunmuştur. Araştırma sonucunda anlamlı bir farklılık bulunamamasının eğitim süresinin kısalığından ve örneklemin küçüklüğünden kaynaklanabileceği ifade edilmiştir. Karabak ve Güneş (2013)'in çalışmasında belirtildiği üzere Bilişim Teknolojileri ve Yazılım dersinin öğretim programında Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanının yıllık plandaki süresi 11 haftalık 22 ders saati süresine denk gelmektedir.

Bu araştırmada anlamlı farklılık çıkmasının sebebi olarak 22 ders saati eğitim verilmesi ve örneklem büyüklüğünün yeterli olması gösterilebilir. Scratch ile yapılan programlama öğretiminin sonucunda, öğrencilerin problem çözme becerilerinin olumlu yönde etkilendiği (Akcaoglu & Koehler, 2014; Brown ve diğerleri, 2008; Calder, 2010; Nam ve diğerleri, 2010; Shin & Park, 2014) sonucuna ulaşılmıştır. Ayrıca, problem çözme becerileri sınıf seviyesine ve cinsiyete göre farklılık göstermemektedir.

Ortaya çıkan sonuçlara göre; problem çözme becerisine sahip öğrenciler yetiştirmek için alternatif bir metot olarak Scratch ile programlama öğretiminde oyun tasarımının gerçekleştirilmesi önerilmektedir. Ayrıca, araştırmanın gerçek deneysel desenler ile gerçekleştirilmesi önerilmektedir.

5.2. Yapılandırılmamış Problemlerin Çözüm Süreci

İlgili literatürde programlama konusu ile ilgili yayınlarda 2012 yılından bu yana istikrarlı bir artış görülse de bu yayınlar kodlama öğretiminin pedagojisine yoğunlaşmamaktadır. Öğrencilerin, Scratch ile oyun programlama sürecindeki işlem adımları kodlama öğretiminin pedagojisine yönelik öğretmenlere rehberlik edebilir.

5.2.1. Yorumlama ve problemi tanımlama. Yorumlama ve problemi tanımlama aşamasına baktığımızda öğrencilerin oyunu zihinlerinde gerçek hayatı referans alarak kurguladıkları ve bu kurguladıkları oyunun tasarımını metne döktükleri görülmektedir. Öğrenciler, oyunu zihinlerinde tasarlamadan direkt uygulamaya yönelik kod bloklarını birleştirmeye geçtiğinde zorlanacaklarını ifade etmişlerdir. Gomes ve Mendes (2007) programlama öğrenmenin güçlükleri ve çözümlerini araştırdıkları çalışmalarında öğrencilerin birçok kez, problemi tam anlamıyla anlamadan problemin çözümüne yönelik eyleme geçtiklerini vurgulamıştır. Öğrenciler, Scratch ortamında yapılandırılmamış oyun tasarım görevlerini ilk önce zihinlerinde kurgulayarak ve tasarım sürecini metne dökerek problemi tanımlamaktadırlar.

Öğrenci görüşlerinden de anlaşılacağı üzere; Scratch'ta oyun programlama sürecinde, problemin tanımlanması aşaması yapılandırılmamış problemlerin çözümünde önemli bir süreçtir, çünkü problemin çözüm sürecini kolaylaştırmaktadır. Problemin tanımlanması, problemin yorumlanmasına veya anlaşılmasına bağlıdır (Chi & Glaser, 1983; Jonassen, 1997; Voss & Post, 1988). Problem iyi karakterize edildiğinde (Michaelson, 2015), Scratch ortamında bir çözüm gerçekleştirmek nispeten kolaylaşmaktadır. Öğrenciler, olası çözümleri test etmek ve bilgiyi kodlamak için problemin başlangıçtaki durumunu tasarım odaklı yaklaşımla detaylandırarak kısıtlamaları tanımlamaktadırlar (Chi & Glaser, 1983). Yorumlama ve problemin tanımlanması aşamasında, öğrencilere rehberlik sağlanması

açısından oyundaki kuklaların öğrenciler ile gerçek hayatı referans olarak canlandırılması öğrencinin zihinsel düşünme sürecini hızlandıracak ve kolaylaştıracaktır.

5.2.2. Çözümler geliştirme ve çözüm sürecini izleme. Öğrenciler bu aşamada kod bloklarını oyunun tasarımına göre yerleştirmektedirler. Öğrencilerin referans olarak aldıkları bir tasarım vardır. Her öğrencinin problemin çözümüne yönelik oyun tasarımı farklılık göstermektedir. Öğrenci görüşlerinden de anlaşılacağı üzere; yapılandırılmamış problemlerin, belirli bir karar alma süreci kullanılarak saptanabilen tek bir çözüm yolu yoktur (Chi & Glaser, 1983; Jonassen, 1997). Scratch ile oyun programlama sürecinde her öğrenci kendi temposunda öğrenmekte ve aşamalı olarak kendi bilgisini oluşturmaktadır. (Gomes & Mendes, 2007).

5.2.3. Karar verme ve çözüm sürecini değerlendirme. Öğrencilerin çoğunluğu oyunu doğru tasarlamaktadır, hatalar çoğunlukla kod bloklarını tasarıma uygun birleştirememelerinden kaynaklanmaktadır. Bunun sebebi olarak da tasarımı detaylandırırken kompütasyonel ifadelerle uygun olarak tasarlamadıkları görülmektedir. Buna bağlı olarak da mantıksal ve koşullu ifadelerin yani kompütasyonel ifadelerin karşılığını Scratch programındaki Kontrol menüsünde bulmakta zorlanmaktadırlar. Scratch'ta oyun programlama sürecinde, yapılandırılmamış problemlerin çözüm süreci tasarım süreci olarak düşünülebilir. Öğrencilerin kontrol menüsündeki kompütasyonel ifadelerle uygun olarak oyunu tasarlamaları gerekmektedir. Bu süreç ise bilgi işlemsel süreç olarak tanımlanabilir (Aho, 2012; Barr ve diğerleri, 2011; Michaelson, 2015; Wing, 2006).

İkinci nitel araştırma sorumuzun sonuçlarına göre; kavramsal olarak yapılandırılmamış problemleri çözmek, problem çözümü için sistematik olarak bir araştırma değil de bir tasarım süreci olarak düşünülebilir. Öğrencilerin oyunu zihinlerinde gerçek hayat şartlarına uygun olarak kurgulamaları, kompütasyonel ifadelerle uygun olarak tasarlamaları ve detaylandırmaları problemin çözümünü kolaylaştırmaktadır. Öğrenciler Scratch ile oyun

tasarımında ağırlıklı olarak sırasıyla; oyunu zihinlerinde gerçek hayat şartlarına uygun olarak kurgulamaktadırlar, kurguladıkları oyunun tasarımını metne dökmektedirler, tasarıma uygun olarak kod bloklarını yerleştirmektedirler, programı çalıştırıp bir sonraki aşamaya geçmektedirler ve hata durumunda ilgili bölümdeki kod bloklarını tasarıma göre kontrol etmektedirler. Öğrencilerin hata durumunda, ilgili bölümü tasarıma göre kontrol etmeleri bize; oyunu kompüstasyonel ifadelerle uygun tasarımlarının önemini göstermektedir (Aho, 2012; Barr ve diğerleri, 2011; Michaelson, 2015; Wing, 2006).

Öğrencilerin problem çözme adımlarını incelediğimizde; Scratch ortamında yapılandırılmamış problemleri çözebilmeleri ve oyun projesini başarı ile tamamlamaları için tasarım odaklı yaklaşım uygulanması öğretmenlere önerilmektedir. Öğretmenlere, Scratch Kontrol menüsündeki kompüstasyonel ifadelerle uygun olarak oyunun tasarımını detaylandırmalarında öğrencilere rehberlik etmeleri önerilmektedir.

5.3. Problem Çözme, Programlama ve Özgün Ürün Geliştirme Alanındaki Yeterlikler

5.3.1. Özel stratejiler. Öğrenciler, bilgi işlemsel düşünme becerisinin gelişimi için oyun programlama sürecinde; oyunu zihinlerinde gerçek hayat şartlarına uygun olarak kurgulayıp tasarıma uygun olarak kod bloklarını yerleştirmektedirler ve yapılandırılmamış problemlerin çözümüne yönelik arkadaşları ile işbirliğine geçmektedirler (Barr ve diğerleri, 2011; Wing, 2006). Öğrenci görüşlerinden de anlaşılacağı üzere her öğrencinin bilgi işlemsel düşünme süreci farklılık göstermektedir. Bu durum, günümüzde bireylerin farklı düşünme biçimlerinin farkında olmasının ve problem çözümüne giden yolda bilgisayar mantığıyla kararlar almasının; yani, bilgi işlemsel düşünmesi gerekliliğini ortaya koyduğunu söylemek mümkündür. Katılımcıların programlama öğretiminde kullanılan Scratch yazılımının çok yönlü düşünmeye ve alternatif çözüm yollarının oluşturulmasına katkıda bulunabileceğine ilişkin görüşleri, Scratch yazılımının bilgi işlemsel düşünmeyi bu açılarından destekleyebileceğini göstermesi açısından önemlidir.

5.3.2. Hayal gücü. Öğrenciler Scratch programındaki kukla zenginliğinin ve fantezi kuklalar kullanılarak fantezi eylemlerin gerçekleştirilmesinin hayal güçlerini olumlu yönde etkilediğini ifade etmişlerdir. Scratch ile oyun tasarımında öğrencilerin fantezi kuklaları kullanarak fantezi eylemler gerçekleştirilmesi hayal güçlerinin gelişmesine katkıda bulunmuştur (Burke & Kafai, 2010; Giannakos ve diğerleri, 2013; Wilson & Moffat, 2010). İlgili literatür incelendiğinde Scratch ile programlama öğretiminin katılımcıların hayal gücüne olan etkisi ile ilgili araştırmaların fazla detaylandırılmadığı görülmektedir. Scratch programındaki fantezi kuklaların sayıca fazla olması öğrencinin hayal gücünün gelişimine pozitif yönde etki edeceği düşünülmektedir.

Öğrencilerin hayal güçlerinin gelişmesi için, Scratch programında fantezi kuklalar kullanılarak fantezi eylemlerin gerçekleştirilmesi önerilmektedir. Scratch programı iki boyutlu görsel bir programlama aracı olduğundan dolayı, çalışmanın üç boyutlu görsel bir programlama aracı olan Alice programı kullanılarak uygulanması önerilmektedir.

5.3.3. Scratch programının olumlu yönleri. Öğrenciler çoğunlukla Scratch programında oyun programlamayı eğlenceli bulmuşlardır (Fesakis & Serafeim, 2009; Genç & Karakuş, 2012; Wilson & Moffat, 2010). Oyun programlamanın başarı hissini arttırdığını ve kendilerine olan özgüvenlerinin arttığını ifade etmişlerdir (Maloney ve diğerleri, 2004). Öğrenciler, yapılandırılmamış oyun tasarım görevlerinin çözümüne yönelik uğraşırken başarı hissini etkisiyle özgüvenleri artmakta ve bilgi işlemsel düşünme becerisinin gelişimine pozitif etki yapmaktadır (Barr ve diğerleri, 2011). İçsel bir motivasyona sahip olmayan öğrenciler pek başarılı olamaz. Bu araştırmada, Scratch ile oyun programlamanın, öğrenci katılım ve motivasyonunu artırdığı (Adams & Webster, 2012; Claypool, 2013), eğitim sürecini daha eğlenceli bir hale dönüştürdüğü belirlenmiştir. Scratch ortamında oyun programlamanın, öğrencilerin derse motivasyonunu ve katılımını sağlamak için güçlü bir araç olarak kullanımı ortaya çıkmıştır.

Birçok öğrencinin programlamaya çalışmak için yeterli motivasyonu yoktur (Gomes & Mendes, 2007), fakat Scratch programı eğlenceli ve motive edici özellikleriyle programlamayı teşvik etmektedir. Öğrenciler, oyun tasarım sürecinde koşullu ifadeleri ve döngüleri (kompütasyonel ifadeleri) kullanmak zorunda kalmışlardır. Dolayısıyla oyun tasarımı projeleri, öğrencileri programlama deyimlerini öğrenmeye zorlamaktadır. Öğrenciler Scratch ile oyun programlama sürecinin ayrıca görsel programlama becerilerinin gelişmesine (Adams, 2010; Adams & Webster, 2012; Claypool, 2013; Lewis, 2010; Malan & Leitner, 2007; Meerbaum-Salant ve diğerleri, 2013) katkı sağladığını ifade etmişlerdir.

5.3.4. Scratch programının olumsuz ve eksik yönleri. Öğrencilerin çoğunluğu (n=13) Scratch programının olumsuz ve eksik bir yönünün olmadığını ifade etmiştir. Kuklaların 3 boyutlu olmaması önemli bir eksiklik olarak gösterilebilir. Ayrıca kuklaların yeterli sayıda kılığının olmaması öğrenciyi kuklayı canlandırma aşamasında sınırlamaktadır. Fantezi kuklalar çoğu öğrencinin hayal gücünün gelişimine sebep olurken bazı öğrenciler ise fantezi kuklaların bulunmasını yadsımıştır. Öğrenciler ayrıca yeterli sayıda dekorun ve kuklanın bulunmadığını ifade etmiştir. İlgili literatür incelendiğinde, Scratch programının olumsuz ve eksik yönlerine yönelik çalışmalara rastlanmamıştır.

Kaynakça

- Adams, J. C. (2010). Scratching middle schoolers' creative itch. G. Lewandowski, S. Wolfman, T. J. Cortina & E. L. Walker (Ed.), *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 356-360). New York: ACM.
- Adams, J. C., & Webster, A. R. (2012). What do students learn about programming from game, music video, and storytelling projects? L. S. King, D. R. Musicant, T. Camp & P. Tymann (Ed.), *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 643-648). New York: ACM.
- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-design and learning (GDL) After-school program. *Computers & Education*, 75, 72-81.
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom. J. Lohmann (Ed.), *ASEE Annual Conference & Exposition* (pp. 9285-9300). Washington DC: American Society for Engineering Education.
- Burke, Q., & Kafai, Y. B. (2010). Programming & storytelling: Opportunities for learning about coding & composition. N. Parés & M. Oliver (Ed.), *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 348-351). New York: ACM.
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2017). *Bilimsel araştırma yöntemleri*. Ankara: Pegem Akademi.

- Calder, N. (2010). Using scratch: An integrated Problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- Can, A. (2013). *SPSS ile bilimsel araştırma sürecinde nicel veri analizi*. Ankara: Pegem Akademi.
- University of Pittsburgh (1983). *Problem solving abilities*. Pittsburgh: L. R. A. D. Center.
- Claypool, M. (2013). Dragonfly: Strengthening programming skills by building a game engine from scratch. *Computer Science Education*, 23(2), 112-137.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37-46.
- Coşkun, M. (2004). *Coğrafya öğretiminde proje tabanlı öğrenme yaklaşımı*. (Yayınlanmamış Doktora Tezi), Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Creswell, J. W. (2013). *Qualitative inquiry and research design: Choosing among five approaches*. London: Sage Publications.
- Creswell, J. W., & Clark, V. L. P. (2007). *Designing and conducting mixed methods research*. Thousand Oaks, California: Sage Publications.
- Dahotre, A., Zhang, Y., & Scaffidi, C. (2010). A qualitative study of animation programming in the wild. G. Succi, M. Morisio & N. Nagappan (Ed.), *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 275-284). New York: ACM.
- Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. P. Brézillon, I. Russell & J.-M. Labat (Ed.), *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 258-262). New York: ACM.

- Funke, J. (2010). Complex problem solving: A case for complex cognition? *Cognitive Processing*, 11(2), 133-142.
- Ge, X. (2001). *Scaffolding students' Problem-solving processes on an Ill-structured task using question prompts and peer interactions*. (Doctoral dissertation), The Pennsylvania State University, Pennsylvania.
- Genç, Z., & Karakuş, S. (2012). Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında scratch kullanımı. Z. Genç (Ed.), *5th International Computer & Instructional Technologies Symposium* (pp. 981-987). Elazığ: Fırat Üniversitesi.
- Giannakos, M. N., Jaccheri, L., & Proto, R. (2013). Teaching computer science to young children through creativity: Lessons learned from the case of Norway. M. v. Eekelen, E. Barendsen, P. Sloep & G. v. d. Veer (Ed.), *Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research* (pp. 103-111). Heerlen: Open Universiteit.
- Gomes, A., & Mendes, A. J. (2007). Learning to Program-difficulties and solutions. C. S. Furtado & M. d. G. Rasteiro (Ed.), *Proceedings of the International Conference on Engineering Education* (pp. 411). Maryland: International Network on Engineering Education and Research.
- Gültekin, K. (2006). *Çoklu ortamın bilgisayar programlama başarısı üzerine etkisi*. (Yayımlanmamış Yüksek Lisans Tezi), Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Hong, N. S. (1998). *The relationship between Well-structured and Ill-structured problem solving in multimedia simulation*. (Doctoral dissertation), The Pennsylvania State University, Pennsylvania.

- Jonassen, D. H. (1997). Instructional design models for Well-structured and Ill-structured Problem-solving learning outcomes. *Educational Technology Research and Development, 45*(1), 65-94.
- Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology Research and Development, 48*(4), 63-85.
- Jonassen, D. H., & Kwon II, H. (2001). Communication patterns in computer mediated versus Face-to-face group problem solving. *Educational Technology Research and Development, 49*(1), 35-51.
- Kafai, Y. B., Fields, D. A., Roque, R., Burke, W., & Monroy-Hernández, A. (2012). Collaborative agency in youth online and offline creative production in scratch. *Research and Practice in Technology Enhanced Learning, 7*(2), 63-87.
- Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education, 13*(1), 33-50.
- Karabak, D., & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi, 2*(3), 163-169.
- Kordaki, M. (2012). Diverse categories of programming learning activities could be performed within Scratch. *Procedia-Social and Behavioral Sciences, 46*, 1162-1166.
- Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *International Biometric Society, 33*(1), 159-174.
- Lee, Y.-J. (2011). Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today, 34*(2), 26-31.
- Lewis, C. M. (2010). How programming environment shapes perception, learning and goals: Logo vs. scratch. G. Lewandowski, S. Wolfman, T. J. Cortina & E. L. Walker (Ed.),

- Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 346-350). New York: ACM.
- Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. I. Russell, S. Haller, J. D. Dougherty & S. Rodger (Ed.), *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (pp. 223-227). New York: ACM.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: A sneak preview *Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing* (pp. 104-109). Washington DC: IEEE Computer Society.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4).
- Mason, J. (2006). Mixing methods in a qualitatively driven way. *Qualitative Research*, 6(1), 9-25.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). Habits of programming in scratch. G. Rößling, T. Naps & C. Spannagel (Ed.), *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education* (pp. 168-172). New York: ACM.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239-264.
- Michaelson, G. (2015). Teaching programming with computational and informational thinking. *Journal of Pedagogic Development*, 5(1), 51-66.
- Milli Eğitim Bakanlığı (2012). *Bilişim teknolojileri ve yazılım dersi (5, 6, 7 ve 8. Sınıflar) öğretim programı.*

<http://ttkb.meb.gov.tr/program2.aspx/program2.aspx?islem=1&kno=196> 'den alınmıştır.

- Nam, D., Kim, Y., & Lee, T. (2010). The effects of Scaffolding-based courseware for the scratch programming learning on student problem solving skill. S. L. Wong (Ed.), *Proceedings of the 18th International Conference on Computers in Education* (pp. 723-727). Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
- Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education*, 59(2), 385-398.
- Rosenbaum, E. (2009). Jots: Cultivating reflective learning in scratch. P. Paolini & F. Garzotto (Ed.), *Proceedings of the 8th International Conference on Interaction Design and Children* (pp. 284-285). New York: ACM.
- Scaffidi, C., & Chambers, C. (2012). Skill progression demonstrated by users in the scratch animation environment. *International Journal of Human-Computer Interaction*, 28(6), 383-398.
- Scratch. (2018). <https://scratch.mit.edu> internet adresinden 14.01.2018 tarihinde erişildi.
- 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.
- Sim, J., & Wright, C. C. (2005). The kappa statistic in reliability studies: Use, interpretation, and sample size requirements. *Physical Therapy*, 85(3), 257-268.
- Trilling, B., & Fadel, C. (2009). *21st century skills: Learning for life in our times*. San Francisco: John Wiley & Sons.

- Voss, J. F., & Post, T. A. (1988). The nature of expertise. In M. T. H. Chi, C. R. Glaser & M. J. Farr (Eds.), *On the solving of Ill-structured problems* (pp. 261-285). New York: Psychology Press.
- Wilson, A., & Moffat, D. C. (2010). Evaluating scratch to introduce younger school children to programming. J. Lawrence & R. Bellamy (Ed.), *Proceedings of the 22nd Annual Psychology of Programming Interest Group* (pp. 7). Leganés, Spain: Universidad Carlos III de Madrid.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Yıldırım, A., & Şimşek, H. (2013). *Sosyal bilimlerde nitel araştırma yöntemleri*. Ankara: Seçkin Yayıncılık.

Ekler

Ek 1: Problem Çözme Becerisi Ölçeği

	Her zaman	Sık sık	Ara sıra	Pek az	Hiçbir zaman
* Zor bir problemi çözmeye başlamadan önce ne yaparsın?					
1. Problemin benden tam olarak ne istediğini anlayıp anlamadığımı düşünürüm.	()	()	()	()	()
2. Daha önce benzer bir problem üzerinde çalışıp çalışmadığımı hatırlamaya çalışırım.	()	()	()	()	()
3. Problemi çözmek için bana gereken bilgiler üzerine düşünürüm.	()	()	()	()	()
4. Problemden bana gerekmeyecek bilgiler olup olmadığına bakarım.	()	()	()	()	()
5. Problemin sınırları üzerine düşünmeye çalışırım.	()	()	()	()	()
* Problem üzerinde çalışırken ne yaparsın?					
6. Ulaşılabilecek bütün bilgileri ve sınırlarını listelerim.	()	()	()	()	()
7. Verilen bilgilerden çözüme ilişkin olanları teşhis etmeye çalışırım.	()	()	()	()	()
8. Kafamda ya da bir kağıt üzerinde, problemi anlamama yardımcı olacak bir şekil oluştururum.	()	()	()	()	()
9. Problem üzerinde çalışırken tüm adımları tek tek planlarım	()	()	()	()	()
10. İlerlediğim her bir adımda probleme tekrar dönüp bakmaya devam ederim.	()	()	()	()	()
* Problem üzerinde çalışmayı bitirdikten sonra ne yaparsın?					
11. Makul olup olmadığını görmek için problem çözmeye yöntemime tekrar bakarım.	()	()	()	()	()
12. Çözümümü destekleyecek veya doğrulayacak delilleri bulmaya çalışırım.	()	()	()	()	()
13. Çözümler üzerine düşünürüm ve başka alternatifler olup olmadığını görmeye çalışırım.	()	()	()	()	()
14. Problemin çözümüne farklı açılardan bakmaya çalışırım.	()	()	()	()	()
15. Sonucumu veya hipotezimi, kendime "eğer..... olsaydı, ne olurdu?" şeklinde sorular sorarak test ederim.	()	()	()	()	()
* Problemler üzerinde hangi yöntemi uygulayarak çalışıyorsun?					
16. Problemi anlamamı sağlayacak bir şekil çizerim.	()	()	()	()	()
17. Öncelikle bir hipotez oluşturur ve sonra onu test ederim (deneyim)	()	()	()	()	()
18. Bu problemi çözmeye yarayacak gerekli adımları seçerim.	()	()	()	()	()
19. Problemleri veya hedefleri öncelik sırasına göre sıralar ve en önemli olan bir tanesinde odaklanırım.	()	()	()	()	()
20. Bir problem çözmeye modeli takip ederim.	()	()	()	()	()

Ek 2: Görüşme Formu

SCRATCH İLE PROGRAMLAMA ÖĞRETİMİNİN PROBLEM ÇÖZME BECERİSİ ÜZERİNDEKİ ETKİSİNE İLİŞKİN ÖĞRENCİ GÖRÜŞME FORMU

Tarih: ___/___/___ Saat(Başlangıç/Bitiş): ___/___

GİRİŞ

Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisini araştırmak ve bu sürece ilişkin öğrenci görüşlerini değerlendirmek amacıyla siz öğrencilerle görüşmeler yapıyorum. Görüşmemize geçmeden önce, yaptığım tüm görüşmelerde verilen bilgiler sadece bu araştırmada kullanılacak ve kişisel bilgiler kesinlikle gizli tutulacaktır. Görüşmenin 15 dakika süreceğini tahmin ediyorum ve izin verirseniz görüşmeyi kaydetmek istiyorum. Bu şekilde hem zamanı daha etkili kullanabiliriz hem de sorulara vereceğiniz yanıtların kaydını daha ayrıntılı tutma fırsatı elde edebilirim.

Bu araştırmaya katılmayı kabul ettiğiniz için şimdiden teşekkür ederim. Eğer sizin bana görüşmeye başlamadan önce sormak istediğiniz bir soru varsa, önce bunu yanıtlamak istiyorum.

SORULAR

1. Scratch' ta zor bir problemi çözmeye başlamadan önce ne yaparsın?
2. Scratch' ta problem üzerinde çalışırken ne yaparsın?
3. Scratch' ta problem üzerinde çalışmayı bitirdikten sonra ne yaparsın?
 - Hata oyunu doğru tasarlamamandan mı yoksa kod bloklarını tasarıma uygun birleştirememenden mi kaynaklanmaktadır?
4. Scratch' ta problemler üzerinde hangi yöntemi (strateji, yol) uygulayarak çalışıyorsun?

5. Hayal gücünüzde gelişme olduysa, bu duruma uygun örnek verebilir misiniz?
6. Scratch programının hoşunuza giden yönleri nelerdir?
7. Scratch programının hoşunuza gitmeyen ve size göre eksik yönleri nelerdir?



Ek 3: İçerik Analizi Kod Tablosu

Bu rehber analiz.xlsx dosyasındaki cümlelerin kodlanması işlemini gerçekleştirmek amacıyla hazırlanmıştır. Kodlamalar yapılırken, her bir cümle aşağıda verilen kodlardan yalnızca birine uyacak şekilde düzenlenmiştir. Excel dosyasındaki cümlelerin bulunduğu satırdaki ilgili olan kod hangisiyse “1” rakamı yazılmalıdır.

Scratch Programının Hayal Gücünün Gelişimine İlişkin Açıklamalı Kod Tablosu

Kod	Tanım	Örnek
Canlandırma	Tek tek resimleri veya hareketsiz cisimleri gösterim sırasında hareket duygusu verebilecek biçimde düzenleme ve filme aktarma işi.	Hayal gücüme göre kuklaları kullanarak istediğimi yapabilirim. İstedığımız dekoru ve kuklaları kullanabiliriz. Ejderhayı çölde uçurabiliriz mesela.
Kukla Çeşitliliği	Scratch programındaki kuklaların çeşidinin çok olması.	Hayali şeyler düşünmemi Scratch'taki kuklalar etkiledi. Oyunu tasarlama aşamasında
Kurgulama	Zihinsel kurgulama, zihinsel tasarım, zihinde canlandırma ile ilişkili kavramdır.	hangi kuklalarla nasıl tasarlayacağım esnasında zihnimde düşünürken, hayal gücümü zorladı.

Scratch Programının Olumlu Yönlerinin Açıklamalı Kod Tablosu

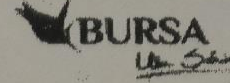
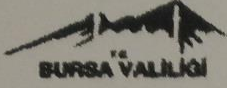
Kod	Tanım	Örnek
-----	-------	-------

Eğlenceli	Eğlendiren, hoş giden.	Eğlenceli, zihni çalıştırarak güzel tasarımlar yapılmasını sağlıyor.
Özgüven	İnsanın kendine güvenme duygusu.	Oyun yapınca kendime güvenim artıyor, arkadaşlarıma hoşuna gidiyor, başarabileceğime inanıyorum.
Oyun Programlama	Scratch programında kod bloklarını, dekorları ve kuklaları kullanarak oyunu yaratma.	Oyun yapabilmek. Ne istiyorsak onu yapıyoruz. Kuklaları canlandırmak hoşuma gitti. Kuklalara istediğimizi yaptırıyoruz.
Hayal Gücü	Zihnin hayal yaratma yetisi, düş gücü, imgelem, muhayyile.	Hayal gücümü geliştirdi, ilginç şeyler düşünmeme yardımcı oldu. Oyun yapmak hayali şeyler düşünmeme sebep oldu.
Problem Çözme	Yeni olay ya da durumlar karşısında var olan ilişkileri ortaya çıkarma, yeni ilişkiler kurma ve güdülen amaca göre belli bir sonuç elde etme işi.	Daha fazla yaratıcı olmamızı sağlıyor, bir probleme birden fazla çözüm bulmak için zihnimizi zorluyoruz, beyin fırtınası yapmak zorunda kalıyoruz.

Scratch Programının Olumsuz/Eksik Yönlerinin Açıklamalı Kod Tablosu

Kod	Tanım	Örnek
Fantezi Kuklalar	Gerçek dışı hayali kuklalar.	Bence insan kuklalarında daha güzel ve düzgün kuklalar olabilirdi, çok saçma ve kötü kuklalar var.
Az Kılık	Kuklaların kılık sayısının yetersiz olması.	Kuklaların biraz daha fazla kılıkları olsa iyi olur.
Az Dekor	Sahnenin dekor sayısının yetersiz olması.	Daha fazla dekor olabilirdi.
Az Kukla	Kuklaların sayıca yetersiz olması.	Daha fazla kukla olabilirdi.
3D Eksikliği	Dekor ve kuklaların x, y, z eksenlerinin olmaması.	Yaratıcılığımızın gelişmesi için üç boyutlu olabilirdi.

Ek 4: İzin Yazısı



T.C.
BURSA VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : 86896125/605/3855520
Konu: Özkan VATANSEVER
Araştırma İzni.

10/04/2015

ULUDAĞ ÜNİVERSİTESİ REKTÖRLÜĞÜNE
(Genel Sekreterlik)

İlgi : a) M.E.B. Araştırma, Yarışma ve Sosyal Etkinlik İzinleri konulu 07/03/2012 tarihli ve 2012/13 sayılı Genelgesi
b) 16/03/2015 tarih ve 8750 sayılı yazınız

Üniversiteniz Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı yüksek lisans öğrencisi Özkan VATANSEVER'in "Scatch ile Programlama Öğretiminin Ortaokul 5 ve 6. Sınıf Öğrencilerinin Problem Çözme Becerileri Üzerindeki Etkisinin İncelenmesi" konulu tez çalışmasını ilimiz İnegöl ilçesi Şükrü Naili Paşa Ortaokulu 5 ve 6. sınıf öğrencilerine uygulama isteği ile ilgili onay ekte gönderilmiştir.

Bilgilerinizi, ilgilinin çalışmasının tamamlanmasından sonra İl Millî Eğitim Müdürlüğümüze çalışmanın sonucu ile ilgili bilgi verilmesini arz ederim.

Mustafa KAHYA
Vali a.
Millî Eğitim Müdür Yardımcısı

EK:
Makam Onayı (1 Sayfa)

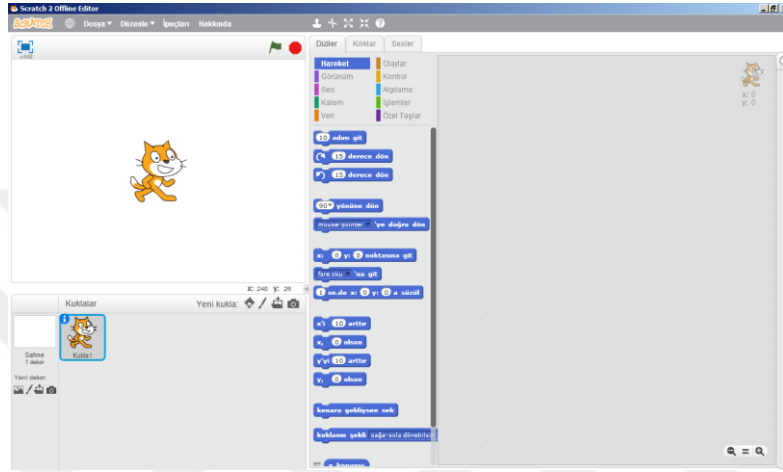
Murat TEMURÇİN
Mamur
Güvenli Elektronik İmzalı
Aslı ile Aynıdır.
10 Nisan 2015

Yeni Hükümet Konağı İl Millî Eğitim Müdürlüğü
Web: <http://bursa.meb.gov.tr>
E-posta: bursamem@meb.gov.tr

Ayrıntılı Bilgi: Engin SEYEMEN VHKİ
Tel: (0 224) 256 70 00
Tel: (0 224) 215 25 39

Ek 5: 5. Sınıf Etkinlikler

Tarih	Etkinlikler
16-20 Mart	Problem çözme becerisi ölçeği ön test olarak uygulandı. Problem çözme sürecindeki aşamalar öğrencilerle tartışıldı.
23-27 Mart	Scratch 2.0 ve Scratch paylaşım sitesi http://scratch.mit.edu tanıtıldı.



Dans edelim animasyonu yapıldı. Animasyon sınıf ortamında bir kız, bir erkek öğrenci ile canlandırıldı.



Senaryo:

Erkek 1 sn. de kızın yanına git

Dans edelim mi? diye sor ve bekle

Eğer yanıt=evet ise

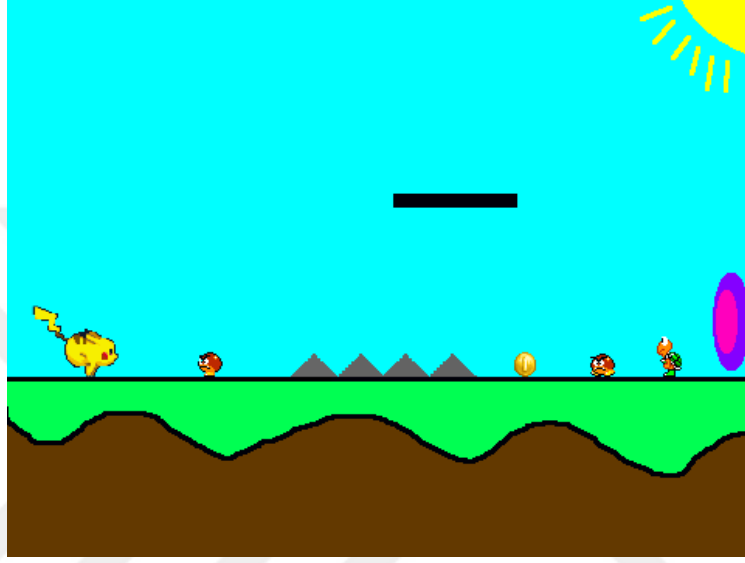
Dans et

Hayır ise

Geri dön

30-3 Nisan

Yedi seviyeden oluşan Pikachu oyunu yapımına başlandı.



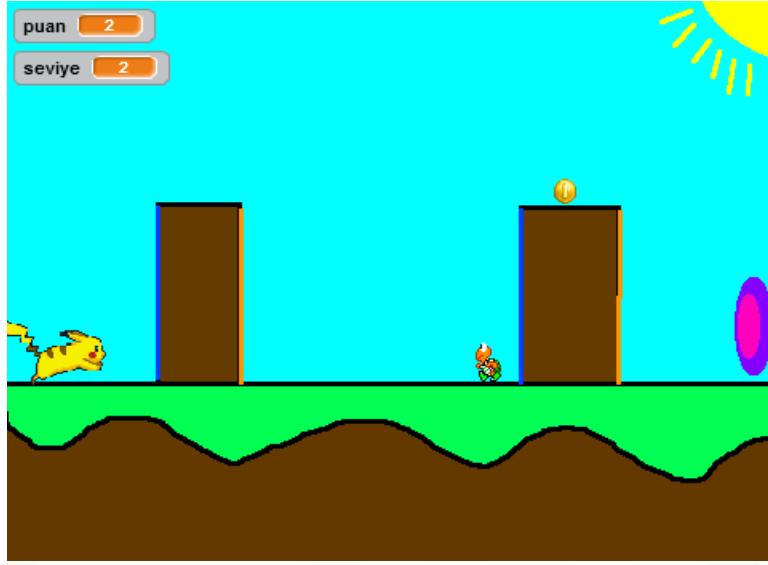
6-10 Nisan

1. Seviye



13-17 Nisan

2. Seviye



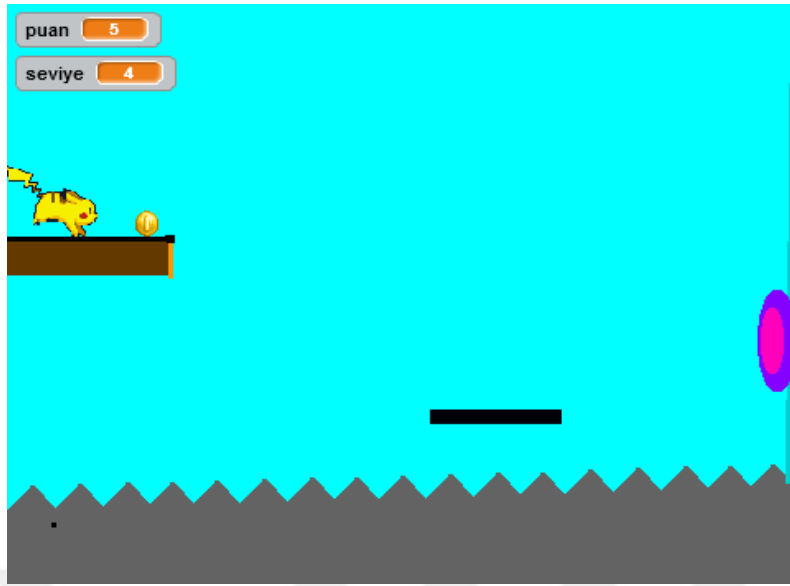
20-24 Nisan

3. Seviye



27-1 Mayıs

4. Seviye">



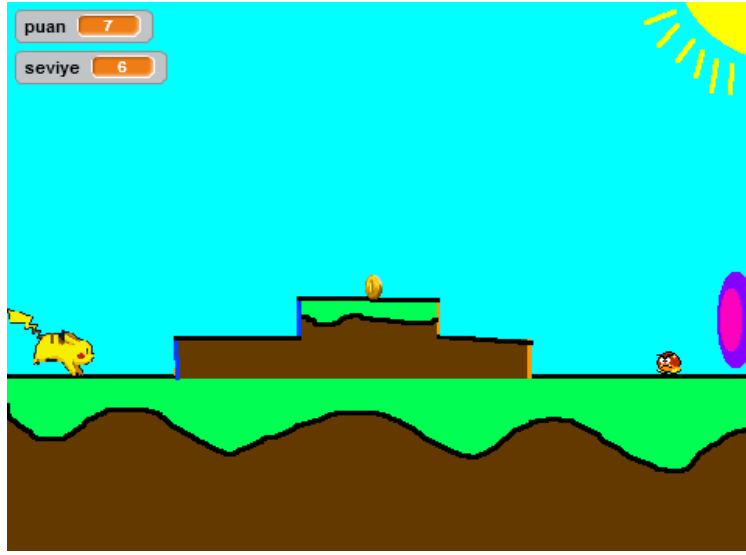
4-8 Mayıs

5. Seviye



11-15 Mayıs

6. Seviye">



18-22 Mayıs

7. Seviye



25-29 Mayıs

Öğrencilerin oyun projesindeki hataları ve eksiklikleri gidermeleri sağlandı.

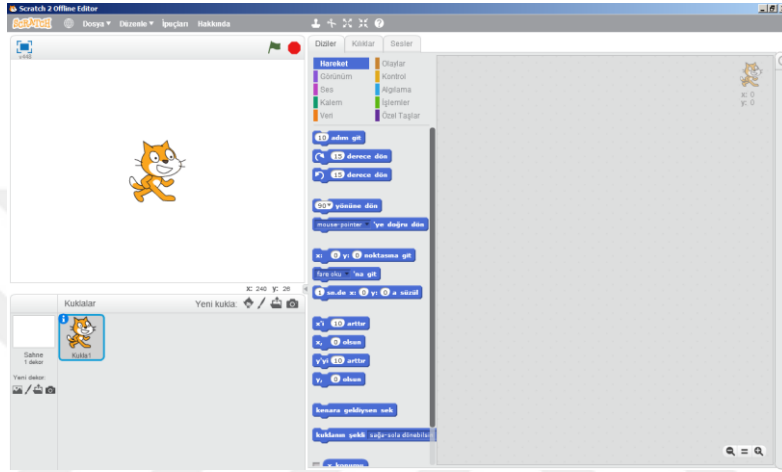
1-5 Haziran

Problem çözme becerisi ölçüğü son test olarak uygulandı.

Görüşmeler yapıldı.

Ek 6: 6. Sınıf Etkinlikler

Tarih	Etkinlikler
16-20 Mart	Problem çözme becerisi ölçęęi son test olarak uygulandı. Problem çözme sürecindeki aşamalar öğrencilerle tartışıldı.
23-27 Mart	Scratch 2.0 ve Scratch paylaşım sitesi http://scratch.mit.edu tanıtıldı.



Dans edelim animasyonu yapıldı. Animasyon sınıf ortamında bir kız, bir erkek öğrenci ile canlandırıldı.



Senaryo:

Erkek 1 sn. de kızın yanına git

Dans edelim mi? diye sor ve bekle

Eğer yanıt=evet ise

Dans et

Hayır ise

Geri dön

30-3 Nisan

Altı seviyeden oluşan Mario oyunu yapımına başlandı.



6-10 Nisan

1. Seviye

puan 0

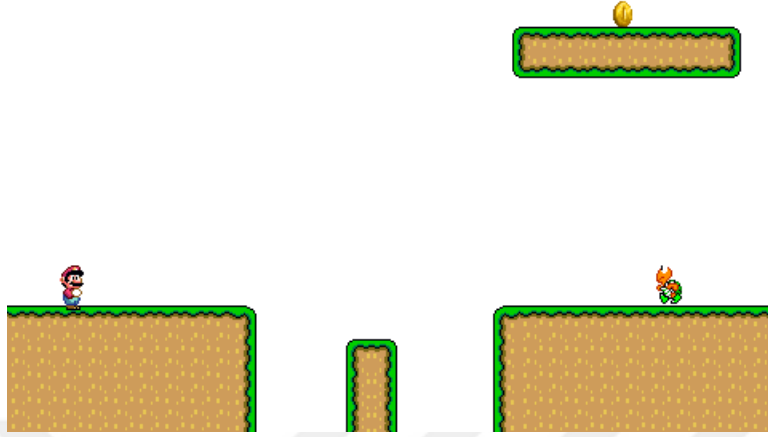
seviye 1



13-17 Nisan

2. Seviye

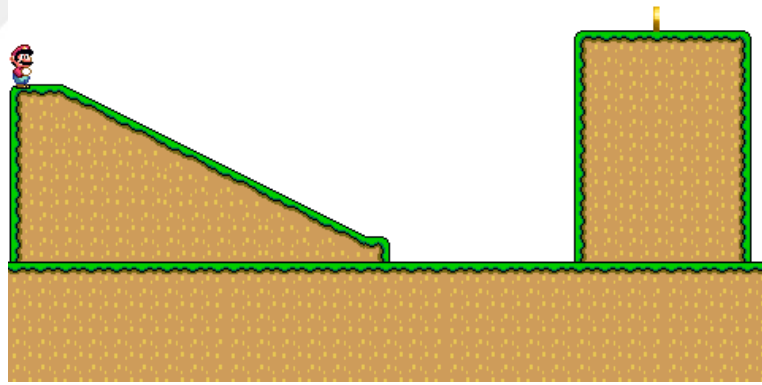
puan 2
seviye 2



20-24 Nisan

3. Seviye

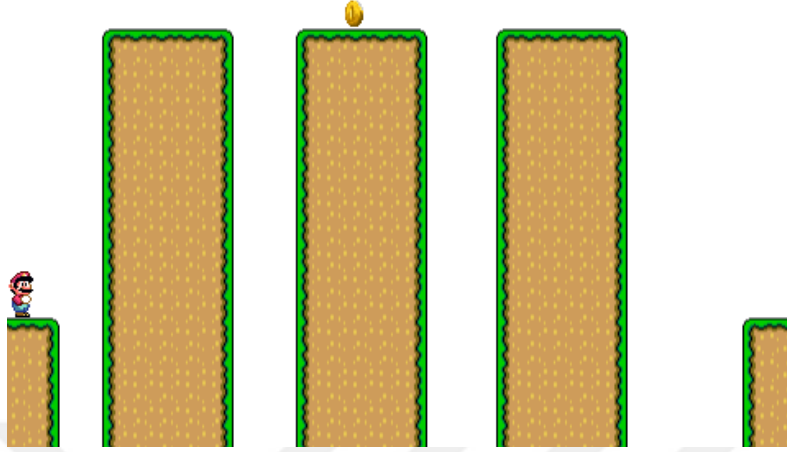
puan 4
seviye 3



27-1 Mayıs

4. Seviye

puan 5
seviye 4



4-8 Mayıs

5. Seviye

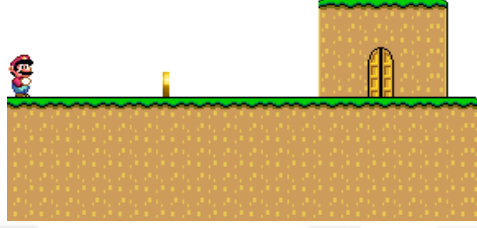
puan 6
seviye 5



11-15 Mayıs

6. Seviye

puan 8
seviye 6



- 18-22 Mayıs Öğrencilerin oyun projesindeki hataları ve eksiklikleri gidermeleri sağlandı.
- 25-29 Mayıs Öğrencilerin oyun projesindeki hataları ve eksiklikleri gidermeleri sağlandı.
- 1-5 Haziran Problem çözme becerisi ölçeği son test olarak uygulandı.
Görüşmeler yapıldı.
-

Özgeçmiş

Doğum Yeri ve Yılı	: Kırcalı - 1983		
Öğrenim Gördüğü Kurumlar	: Başlama	Bitirme	Kurum Adı
	Yılı	Yılı	
Lisans	: 2000	2004	Çanakkale 18 Mart Üni.
Yüksek Lisans	: 2012	2018	Uludağ Üniversitesi
Bildiği Yabancı Diller ve Düzeyi	: İngilizce – Orta		
Çalıştığı Kurumlar	: Başlama ve Ayrılma	Kurum Adı	
	2004	MEB	
Yurt Dışı Görevleri	:		
Kullandığı Burslar	:		
Aldığı Ödüller	:		
Üye Olduğu Bilimsel ve Mesleki			
Topluluklar	:		
Editör veya Yayın Kurulu			
Üyeliği	:		
Yurt İçi ve Yurt Dışında			
Katıldığı Toplantılar	:		
Katıldığı Yurt İçi ve Yurt Dışı			
Bilimsel Toplantılar	:		

Vatansever, Ö., Gündüz, A., & Öncü, S. (26-28 Nisan 2013). *Ortaokul 5. Sınıf Öğrencilerinin Dyned İngilizce Dil Eğitim Yazılımının Kullanımını Olumsuz Yönde Etkileyen Faktörlerin Belirlenmesi*. Paper presented at the International Conference On Innovation And Challenges In Education, Kütahya.

Vatansever, Ö. & Baltacı-Göktalay, Ş. (6-8 Nisan 2017). *Scratch İle Programlama Öğretiminin Ortaokul 5. ve 6. Sınıf Öğrencilerinin Problem Çözme Becerisi Üzerindeki Etkisinin İncelenmesi*. Uluslararası Avrasya Sosyal Bilimler Kongresinde sunuldu, Antalya.

Yayımlanan Çalışmalar :

Diğer Profesyonel Etkinlikler :

08/02/2018

Özkan VATANSEVER

