

**T.C.
SAKARYA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM
DALI**

**JQUERY TABANLI BLOK PROGRAMLAMA ÖĞRETİMİNİN
PROGRAMLAMAYA YÖNELİK TUTUMA ETKİSİNİN DEĞERLENDİRİLMESİ
(BAŞISKELE ÖRNEĞİ)**

YÜKSEK LİSANS TEZİ

İBRAHİM BAŞTUĞ

DANIŞMAN

DOÇ. DR. MÜBİN KIYICI

HAZİRAN 2019

T.C.
SAKARYA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM
DALI

JQUERY TABANLI BLOK PROGRAMLAMA ÖĞRETİMİNİN
PROGRAMLAMAYA YÖNELİK TUTUMA ETKİSİNİN DEĞERLENDİRİLMESİ
(BAŞISKELE ÖRNEĞİ)

YÜKSEK LİSANS TEZİ

İBRAHİM BAŞTUĞ

DANIŞMAN

DOÇ. DR. MÜBİN KIYICI

HAZİRAN 2019

BİLDİRİM

Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü, Tez-Proje Yazım Kılavuzu'na uygun olarak hazırladığım bu çalışmada:

- Tezde yer verilen tüm bilgi ve belgeleri akademik ve etik kurallar çerçevesinde elde ettiğimi ve sunduğumu,
- Yararlandığım eserlere atıfta bulunduğumu ve kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir deęiřtirmede bulunmadığımı,
- Bu tezin tamamını ya da herhangi bir bölümünü başka bir tez çalışması olarak sunmadığımı

beyan ederim.


İmza
İbrahim BAŐTUĐ

JÜRİ ÜYELERİNİN İMZA SAYFASI

‘Jquery Tabanlı Blok Programlama Öğretiminin Programlamaya Yönelik Tutuma Etkisinin Değerlendirilmesi (Başiskele Örneği)’ başlıklı bu yüksek lisans tezi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

Başkan Prof. Dr. Cem ÇUHADAR (İmza)

Üye (Danışman) Doç. Dr. Mübin KIYICI (İmza)

Üye Dr. Öğr. Üyesi Onur İŞBULAN (İmza)

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

26.07/2019

(İmza)

Prof. Dr. Ömer Faruk TUTKUN
Enstitü Müdürü ✓

ÖN SÖZ

Gelişen teknolojiler ve değişen dünya, bilişim teknolojilerinin her yerde ve pek çok işte kullanımına ve sonuçta da bazı becerilerin önemini öne çıkarmaktadır. Bu becerilerden biri bilgisayar programlama becerisidir. Literatür tarandığında bu becerinin kazanılmasının klasik yöntemlerle oldukça zor olduğu ortaya çıkmaktadır (Başer, 2013). Bunu çözmek için ortaya atılmış pek çok yöntemden biri görsel programlama dili adı verilen Scratch ve benzeri yazılımların kullanımınıdır. Ancak bu yazılımların da sınırlılıkları olduğu ifade edilmektedir (Du, Wimmer ve Rada, 2016).

Hem öğrencilere programlama becerilerini kazandırmayı, hem de kodları doğrudan kullanmak zorunda kalmadan, tanıma imkânı sunarak onları ileri seviye programlama dillerine hazırlamak için tasarladığım JQuery Tabanlı Blok Programlama (JTBP) yazılımının öğrenci görüşleri üzerinden değerlendirmesi ve programlamaya yönelik tutumun değiştirilmesinde etkililiğini araştırdım.

Bu çalışmayı planlama ve yürütmem de sağladığı katkılardan dolayı değerli hocam ve tez danışmanım, Doç. Dr. Mübin KIYICI'ya teşekkürü bir borç bilirim.

Daima yanımda olan ve manevi desteğini kelimelerle ifade edemeyeceğim hayat arkadaşım, sevgili eşim Ayşe BAŞTUĞ'a ve kendisine yeterince zaman ayıramadığım için tezimin bitmesini sabırsızlıkla bekleyen kızım Aybüke BAŞTUĞ'a en içten teşekkürlerimi borç bilirim.

ÖZET

JQUERY TABANLI BLOK PROGRAMLAMA ÖĞRETİMİNİN PROGRAMLAMAYA YÖNELİK TUTUMA ETKİSİNİN DEĞERLENDİRİLMESİ (BAŞISKELE ÖRNEĞİ)

İbrahim BAŞTUĞ, Yüksek Lisans Tezi

Danışman: Doç. Dr. Mübin KIYICI

Sakarya Üniversitesi, 2019.

Programlama konusu eğitimin birçok basamağında öğrencilere öğrenmesi zor gelen konular arasında yer almaktadır. Ortaokul düzeyinde programlama eğitimi “Bilişim Teknolojileri ve Yazılım” dersi kapsamında öğrencilere verilmektedir. Dersi etkili olarak yürütebilmenin yollarından biri Scratch ve benzeri görsel programlama dillerini kullanmaktır. Fakat görsel programlama dillerinin algoritma öğretimi konusunda başarılı olduğu halde programlama öğretimi konusunda birtakım sınırlılıkları mevcuttur. Bu sorunu çözmek için hem Scratch gibi blok mantığının kullanıldığı, hem de bu kodların karşılığı olan “jQuery” kodlarını içeren bir yazılım araştırmacı tarafından geliştirilmiştir.

Bu araştırma, JQuery tabanlı blok programlama (JTBP) sistemini öğrencileri ileri seviye programlama dillerine hazırlamak amacıyla geliştirmek ve bu yazılımı 6. Sınıf öğrencilerinin programlamaya yönelik tutumlarını değiştirmede etkililiği ve öğrenci görüşleri üzerinden değerlendirmek üzere gerçekleştirilmiştir.

Çalışma bulguları, geliştirilen JTBP yazılımının öğrencilerin programlamaya yönelik tutumlarını, özellikle kendi özgün tasarımlarını yaptıktan sonra büyük oranda olumlu yönde değiştirdiğini göstermiştir. Öğrenciler programlamayı daha çok sevmeye başlamışlardır. Oyun tasarımı ve bilişim konusunda kendilerini geliştirme isteklerinde artış gözlenmiştir. Ancak programlamaya yönelik motivasyon ve programlamayı önemli görme durumlarında belirgin bir değişim gözlenmemiştir. JTBP yazılımına yönelik öğrenci görüşleri incelendiğinde öğrenciler yazılımın hoşlarına gittiğini ve yazılımı heyecanlı buldukları belirtmişlerdir.

Anahtar Kelimeler: Programlamaya yönelik tutum, JQuery, JavaScript, Görsel programlama dilleri.

ABSTRACT

EVALUATION OF THE EFFECT OF JQUERY-BASED BLOCK PROGRAMMING ON ATTITUDE TOWARDS PROGRAMMING (BAŞISKELE)

İbrahim BAŞTUĞ, Master Thesis

Supervisor: Assoc. Prof. Dr. Mübin KIYICI

Sakarya University, 2019.

The subject of programming is among the subjects that are difficult to learn in many stages of education. The programming education at secondary level is given to the students within the scope of “Information Technologies and Software” course. One of the ways to carry out the lesson effectively is to use Scratch and similar visual programming languages. However, although visual programming languages are successful in teaching algorithms, there are some limitations in teaching programming. In order to solve this problem, a software was developed by the researcher which includes block logic such as Scratch and JQuery codes corresponding to these blocks.

This research was carried out to develop the JQuery based block programming (JTBP) system in order to prepare students for advanced programming languages and to evaluate this software on the effectiveness of changing 6th grade students' attitudes towards programming and to evaluate the software by students' opinions.

The study findings showed that the developed JTBP software significantly changed the students' attitudes towards programming, especially after making their own original designs. Students begin to like programming more. An increase was observed in their desire to develop themselves in game design and informatics. However, there was no significant change in motivation towards programming and the importance of programming. When the students' opinions about JTBP software were examined, the students stated that they liked the software and found the software exciting.

Keywords: Attitudes towards programming, JQuery, JavaScript, Visual programming languages.

İÇİNDEKİLER

BİLDİRİM.....	i
JÜRİ ÜYELERİNİN İMZA SAYFASI.....	ii
ÖN SÖZ.....	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
TABLolar LİSTESİ	x
ŞEKİLLER LİSTESİ.....	xii
BÖLÜM I	1
GİRİŞ.....	1
1.1. Problem Durumu	1
1.2. Araştırmanın Amacı ve Problemler.....	3
1.3. Alt problemler	3
1.4. Araştırmanın Amacı ve Önemi.....	4
1.5. Sayıtlılar	5
1.6. Sınırlılıklar.....	5
1.7. Tanımlar	5
1.8. Kısaltmalar	6
BÖLÜM II	7
ARAŞTIRMANIN KURAMSAL ÇERÇEVESİ VE İLGİLİ ARAŞTIRMALAR.....	7
2.1. Programlama nedir	7
2.2. Programlama dilleri.....	8
2.3. Dinamik programlama dilleri	9

2.3.1. Dinamik diller	9
2.3.2. Dinamik dil özellikleri	9
2.4. JavaScript	10
2.5. JQuery	11
2.6. Programlama eğitimi	14
2.7. Görsel programlama dilleri uygulamaları	16
2.7.1. Scratch.....	18
2.7.2. Alice	20
2.7.3. Code.studio.org	21
2.7.4. App inventor	22
2.7.5. Greenfoot.....	22
2.7.6. Diğer.....	23
2.8. Programlama yönelik tutum	24
2.9. Bilişim teknolojileri ve yazılım dersi (BTY)	28
2.10. İlgili araştırmalar.....	30
2.11. Alan yazın taramasının özeti.....	33
BÖLÜM III.....	35
YÖNTEM.....	35
3.1. Araştırma Modeli	35
3.2. JTBP Yazılımının Geliştirme Süreci.....	36
3.2.1. Tasarım öncesi analiz.....	37
3.2.2. Uygulama aracı tasarımı	39
3.2.3. Pilot Uygulama	43
3.3. Katılımcılar.....	44
3.4. Veri Toplama Aracı.....	45

3.5. Verilerin Toplanması ve Çözümlemesi	47
BÖLÜM IV	49
BULGULAR	49
4.1. Programlamaya yönelik tutumla ilgili bulgular.....	49
4.1.1.Uygulama öncesi programlamaya yönelik tutum	49
4.1.1.1. Uygulama öncesi programlamaya yönelik ilgi.....	49
4.1.1.2. Uygulama öncesi programlamayı faydalı ve önemli görme	51
4.1.1.3. Uygulama öncesi programlamaya yönelik motivasyon	54
4.1.1.4. Uygulama öncesi programlamaya yönelik olumsuz tutum	55
4.1.2.Uygulama sonrası programlamaya yönelik tutum	58
4.1.2.1. Uygulama sonrası programlamaya yönelik ilgi.....	58
4.1.2.2. Uygulama sonrası programlamaya yönelik motivasyon	60
4.1.2.3. Uygulama sonrası programlamayı önemli ve faydalı görme	61
4.1.2.4. Uygulama sonrası programlamaya yönelik olumsuz tutum.....	62
4.1.2.5. Uygulama sonrası programlama öğrenme süreci	63
4.1.3.Tasarım sonrası programlamaya yönelik tutum.....	64
4.1.3.1. Tasarım sonrası programlamaya yönelik ilgi	64
4.1.3.2. Tasarım sonrası programlamaya yönelik motivasyon.....	66
4.1.3.3. Tasarım sonrası programlama önemli ve faydalı görme	68
4.1.3.4. Tasarım sonrası programlama yönelik olumsuz tutum	69
4.1.3.5. Tasarım sonrası programlama öğrenme süreci.....	70
4.1.4.Programlamaya yönelik tutum görüşlerinin karşılaştırması	72
4.1.4.1. Programlamaya ilgi duyma	72
4.1.4.2. Programlamayı önemli ve faydalı görme	75
4.1.4.3. Programlamaya yönelik motivasyon	76

4.2. JTBP sistemine yönelik tutumla ilgili bulgular	77
4.2.1.Uygulama sonrası JTBP ye yönelik tutum.....	77
4.2.1.1 Uygulama sonrası JTBP’ye yönelik olumlu tutum	77
4.2.1.2 Uygulama sonrası JTBP’ye yönelik olumsuz tutum	79
4.2.2.Tasarım sonrası JTBP’ye yönelik tutum	82
4.2.2.1. Tasarım sonrası JTBP’ye olumlu tutum.....	82
4.2.2.2. Tasarım sonrası JTBP’ye yönelik olumsuz tutum.....	84
4.2.2.3. Tasarım sonrası JTBP ile ilgili öğrenci önerileri.....	87
4.2.3.JTBP ye yönelik tutum görüşlerinin karşılaştırılması.....	88
BÖLÜM V	90
SONUÇ, TARTIŞMA VE ÖNERİLER	90
5.1. Sonuç ve Tartışma	90
5.1.1.Programlama yönelik tutum.....	90
5.1.2.JTBP sistemine yönelik tutum	92
5.2. Öneriler.....	92
KAYNAKLAR.....	94
EKLER	104

TABLÖLAR LİSTESİ

Tablo 1. Programlama Dilleri ve Sınıfları	8
Tablo 2. Çalışma Örnekleme.....	44
Tablo 3. Uygulama Öncesi Programlamaya Yönelik İlgi	49
Tablo 4. Uygulama Öncesi Programlamayı Faydalı Ve Önemli Görme.....	51
Tablo 5. Uygulama Öncesi Programlamaya Yönelik Motivasyon.....	54
Tablo 6. Uygulama Öncesi Programlamaya Yönelik Olumsuz Tutum.....	56
Tablo 7. Uygulama Sonrası Programlamaya Yönelik İlgi	58
Tablo 8. Uygulama Sonrası Programlamaya Yönelik Motivasyon.....	60
Tablo 9. Uygulama Sonrası Programlamayı Önemli ve Faydalı Görme.....	61
Tablo 10. Uygulama Sonrası Programlamaya Yönelik Olumsuz Tutum.....	62
Tablo 11. Uygulama Sonrası Programlama Öğrenme Süreci.....	63
Tablo 12. Tasarım Sonrası Programlamaya Yönelik İlgi	64
Tablo 13. Tasarım Sonrası Programlamaya Yönelik Motivasyon	67
Tablo 14. Tasarım Sonrası Programlama Önemli ve Faydalı Görme	68
Tablo 15. Tasarım Sonrası Programlama Yönelik Olumsuz Tutum.....	70
Tablo 16. Tasarım Sonrası Programlama Öğrenme Süreci Öğrenci Görüşleri.....	71
Tablo 17. Programlamaya İlgi Duyma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm1	72
Tablo 18. Programlamaya İlgi Duyma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm2	73
Tablo 19. Programlamaya İlgi Duyma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm3	73
Tablo 20. Programlamaya İlgi Duyma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm4	73

Tablo 21. Programlamaya İlgili Duyuma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm5	74
Tablo 22. Programlama Önemli ve Faydalı Görme İle İlgili Öğrenci Görüşlerinin Karşılaştırılması.....	75
Tablo 23. Programlamaya Yönelik Motivasyon Görüşlerinin Karşılaştırılması.....	76
Tablo 24. Uygulama Sonrası JTBP'ye Yönelik Olumlu Tutum	77
Tablo 25. Uygulama Sonrası JTBP'ye Yönelik Olumsuz Tutum	80
Tablo 26. Tasarım Sonrası JTBP'ye Olumlu Tutum.....	82
Tablo 27. Tasarım Sonrası JTBP'ye Yönelik Olumsuz Tutum.....	85
Tablo 28. Tasarım Sonrası JTBP İle İlgili Öğrenci Önerileri.....	87
Tablo 29. Öğrencilerin JTBP Sistemine Yönelik Tutumlarının Karşılaştırılması	88

ŞEKİLLER LİSTESİ

Şekil 1. w3scholls örnek ekranı	39
Şekil 2. JTBP yazılımı arayüzü kod bölümü	40
Şekil 3. JTBP yazılımı arayüzü ekran bölümü	41
Şekil 4. Arılar senaryosu ekranı	42
Şekil 5. Yıldız savaşları senaryo ekranı.....	43
Şekil 6. Pilot uygulamada karşılaşılan hosting hata ekranı	44

BÖLÜM I

GİRİŞ

1.1. Problem Durumu

Bilgiye ulaşmanın ve paylaşmanın kolay ve hızlı hale geldiği günümüzde, bilgiyi etkili kullanmak, analiz etmek, ilişkiler kurmak ve değerlendirmek önem kazanmıştır. Günlük yaşamda bilgisayarların sağlık, turizm ve benzeri pek çok alanda vazgeçilmez hale gelmesinde yazılım geliştirme unsurunun etkisi büyüktür. Dünya genelinde yazılım sektörünün öneminin gittikçe artmasıyla beraber bu alanda eğitim görmüş insan ihtiyacı da artmaktadır. (Karabak ve Güneş, 2013).

Öğrencilerin öğrenimlerini yazılım alanında devam ettirmeseler dahi programlama mantığını erken yaşlarda öğrenmelerinin diğer alanlardaki başarılarına da katkıda bulunacağı düşünülmektedir (Karabak ve Güneş, 2013). Programlama becerisi, düşünme becerilerini hedef alan yapısıyla bilişsel gelişime büyük katkı sunmaktadır (Kert ve Uğraş, 2009). Bilgisayar programlama öğrenirken öğrenciler, problem çözme (Pea ve Kurland, 1984), mantıksal düşünme, soyutlama, yaratıcılık, takım çalışması gibi yüksek zihinsel beceriler kazanmaktadır (Lau ve Yuen, 2009; Molina-García, Rodríguez-Elías, Glasserman-Morales ve Rodríguez-Pérez, 2016; Bers, Flannery, Kazakoff ve Sullivan, 2014). Günümüzde programlama becerisi ortaokul hatta ilkokul öğrencileri için bile kazandırılması gereken beceriler içinde yer almaktadır (Yolcu, 2018). Fakat programlama dersi ülkemizde 2012 yılında eklenen ortaokul düzeyinde Bilişim Teknolojileri ve Yazılım (BTY) dersi müfredatına kadar meslek liselerini haricinde yalnızca lisans seviyesinde verilmekteydi.

Günümüzde yazılım geliştirmenin önemi göz önüne alındığında ortaokul 5. ve 6. sınıflarda zorunlu olan “Bilişim Teknolojileri ve Yazılım” dersinin konusu içinde olan programlama konusu daha da önem kazanmaktadır. Programlama dersi başlangıç seviyesinde öğrenciler için zor olarak algılandığı belirtilmektedir (Feldgen ve Clúa, 2004; Fesakis ve Serafeim, 2009). Rodger ve diğerleri (2009) meslek seçiminin ortaokul seviyesinden itibaren

başladığını, öğrencilerin bilgisayar alanında çalışanlarla ilgili algılarının masa başında oturan asosyal insanlarla sınırlı olduğunu ve bununda bilgisayar alanına yönelmeyi olumsuz etkilediğini belirtmektedir. Programlama öğrenmenin zor olduğu gerçeğine vurgu yaparak (Feldgen ve Clúa, 2004; Fesakis ve Serafeim, 2009) bu sürece olumsuz durumlarda eklenince bilgisayar programlamaya karşı olumsuz tutum oluşabildiği ifade etmişlerdir (Akande, 2009'dan aktaran Başer, 2013).

Smith (1968) tutumu, kişiye atfedilen ve kişinin bir objeye ilgili düşünce, duygu ve davranışlarını oluşturan eğilim olarak tanımlamaktadır (aktaran Başer, 2013). Tutum, bir nesne veya olguya karşı hazır olma durumudur (Gökdaş, 2008). Tutumlar, beklenen davranış algıları ile birlikte birinin bilgisayar programlama öğrenmek gibi belirli bir aktiviteye nasıl tepki verebileceğini belirledikleri için önemlidir (Jay ve Willis, 1992). Tutum nasıl öğreneceğimizi ve davranışlarımızı etkilemektedir (Maio ve Haddock, 2009'dan aktaran Özyurt ve Özyurt, 2015). Tutum doğrudan gözlenememekle beraber (Arkonaç, 2005'ten aktaran Gökdaş, 2008) yine de öğrencilerin başarısını etkileyebilmektedir (Başer, 2013). Örneğin, bilgisayarlara yönelik olumlu tutumların, gençlerin ve yaşlı yetişkinlerin bilgisayar kullanımını ve bilgisayarla ilgili sonraki becerileri anlamalarını arttırdığı gösterilmiştir (Utting, Cooper, Kölling, Maloney ve Resnick, 2010).

Programlama konusu zor olduğu (Feldgen ve Clúa, 2004) gibi gelecek kaygısıyla öğrencilerin programlamaya karşı azalan ilgi ve tutumlarını olumlu olarak değiştirmek için çocuklara eğlenceli gelecek bir ortamın ve etkinliklerin olması gereklidir (Kert ve Uğraş, 2009). Son yıllarda, gelişmiş ülkeler özelinde hem yazılımı sevdirmek hem de lisans seviyesine gelmeden önce öğrencilere yazılım geliştirme mantığını öğretmek amacıyla çeşitli araçlar geliştirilmiştir (Karabak ve Güneş, 2013).

Çocukların kendi istekleriyle katıldıkları ve eylemin merkezinde oldukları oyun, kalıcı bilgilerin edinilebileceği bir eylemdir (Erekmekçi ve Fidan, 2012). Oyun geliştirme sürecinde problem çözme eylemine anında geri bildirim verdiğinden bilgisayar oyunlarının tasarımı ve oynanması aşırı derecede motive edici olduğundan dolayı değişik yetenek ve stratejilerin geliştirilmesinde öğrenciler sürece aktif olarak katılım göstermektedirler (Collins ve diğerleri, 1996'dan aktaran Hong, Fadjo, Chang, Geist ve Black, 2010).

Günümüzde her alanda yer edinmiş olan yazılımları geliştirecek insan ihtiyacına paralel programlama eğitimleri önem kazanmaktadır. Milli Eğitim Bakanlığı 5. ve 6. sınıflarda bilişim teknolojileri ve yazılım dersini 2012 yılında zorunlu hale getirmiş ancak içerik konusunda öğretmenlerin özgür bırakılmış olması sadece ana hatları belirlenmesiyle sınırlı kalmıştır (Karabak ve Güneş, 2013). Alanyazın incelendiğinde Scratch ve benzeri programlama eğitimi için kullanılan yazılımlara yönelik çalışmalar ve müfredat önerileri olduğu görülmektedir. Fakat doğrudan kod içermeyen blok tabanlı programlama dillerinin, öğrencileri ileri seviyeli programlama dillerine hazırlayamadığı görülmüştür (Du, Wimmer ve Rada, 2016).

1.2. Araştırmanın Amacı ve Problemler

Bu araştırma, JQuery tabanlı blok programlama (JTBP) sistemini öğrencileri ileri seviye programlama dillerine hazırlamak amacıyla geliştirmek ve bu yazılımı 6. Sınıf öğrencilerinin programlamaya yönelik tutumlarını değiştirmede etkililiği ve öğrenci görüşleri üzerinden değerlendirmek üzere gerçekleştirilmiştir.

Scratch ve benzeri yazılımların blok yapısına sahip olan, hem de kod yapılarının var olduğu yeni bir sistemin tasarımı, öğrencilerin programlama deneyimlerini bir adım daha öteye taşıyacağı ve öğrencileri ileri seviye dillere hazırlayacağı düşünülmüştür. Bu amaçla araştırmacı JQuery kodlarının blok mantığıyla kullanılabilirdiği bir ortam tasarlamıştır. Araştırmacı bu şekilde geliştirilen bir ortamla öğrencilerin programlama mantığı öğrenmenin yanı sıra her platformda çalışabilecek kodlar tasarlayabilmesini amaçlamıştır.

1.3. Alt problemler

1. Öğrencilerin programlamaya yönelik tutumları ne düzeydedir?
2. JTBP sistemine yönelik öğrenci görüşleri nelerdir?
3. Öğrencilerin tasarımlarını tamamlamasından sonra programlamaya yönelik tutumlarında değişim var mıdır?

4. Öğrencilerin tasarımlarını tamamlamasından sonra JTBP sistemine yönelik tutumlarında değişim var mıdır?

1.4. Araştırmanın Amacı ve Önemi

Oyun tasarlayarak öğrenmek öğrencinin kendi öğrenme stiline uygun şekilde öğrenmesini sağlamaktadır (Feldgen ve Clúa, 2004; Fesakis ve Serafeim, 2009). Scratch 8 yaşından 16 yaşına kadar olan bireylerin kendileri için anlamlı olan oyun ve benzeri projeler üstünde çalışırken programlamayı öğrenmelerini sağlamaktadır (Maloney, Resnick, Rusk, Silverman, ve Eastmond, 2010). Fakat bazı araştırmacılar bu tarz yazılımların algoritmayı öğretmekten öte gidemediğini gündeme getirmektedirler. Du, Wimmer ve Rada (2016) çalışmalarında code.org'un öğrencilerin tutumlarını değiştirme konusunda olumlu sonuçlar verdiğini ancak öğrencilerin programlama becerilerine katkıda bulunamadığını ortaya koymaktadır. Scratch ve benzeri yazılımlar editör bağımlıdır ve ileri düzey özellikler eklemeyi desteklememektedir (Good ve Howland, 2015). Öte yandan sosyal ağ, oyun ve diğer kullanım alanlarında geliştirilen yazılımlar için web ve mobil cihazlar tasarım ve geliştirme platformu haline gelmektedir. Bu durum JavaScript gibi eski olduğu halde web tasarımında kullanılan dillerin öne çıkmasına sebep olmuştur (Mikkonen, Taivalsaari, 2007). Bu sebeple her platformda çalışabilen uygulamalar tasarlamak için JavaScript kütüphaneleri kullanan oldukça popüler olarak kullanılan JQuery'i (Ocariza, Pattabiraman ve Zorn, 2011) öğrenmek önemlidir.

Günümüzde programlama eğitimini özellikle ortaokul düzeyinde öğrencilere daha etkili şekilde vermek için birçok yazılım üretilmiştir. Bu yazılımlar görsel özellikleri ve blok yapıları sayesinde öğrencilerin noktalama işaretleri kullanmak zorunda kalmadan, kod ezberlemeden programlamayı öğrenmelerini sağlamaktadır (Kelleher ve Pausch, 2003; Karabak ve Güneş, 2013). Fakat bazı araştırmacılar bu tarz yazılımların algoritmayı öğretmekten öte gidemediğini gündeme getirmektedirler. Du, Wimmer ve Rada (2016) çalışmalarında code.org un öğrencilerin tutumlarını değiştirme de olumlu sonuçlar doğurduğunu fakat öğrencilerin programlama becerilerine katkıda bulunamadığını ortaya koymaktadır. Scratch ve benzeri yazılımlarla ortaya çıkan çalışmaların yalnızca yine bu yazılımların içinde çalışması (Halma, 2005'ten aktaran Alghamdi, 2017), kişinin daha ileri

düzey özellikler eklemek istediğinde yazılımın sınırlarını aşamaması gibi durumlar bulunmaktadır (Good ve Howland, 2015). Bunu aşabilmek için hem scratch ve benzeri yazılımların blok yapısına sahip olan, hem de kod yapılarının var olduğu yeni bir sistem tasarımı ortaya çıkması öğrencilerin programlama deneyimlerini bir adım daha öteye taşıyabilecektir. Java ve benzeri programlama dillerinde olduğu her platformda çalışabilen yazılımlar tasarlarırken kod yapıları ve noktalama işaretlerinin karmaşasından öğrencilerin korkup programlamaya yönelik olumsuz tutum geliştirmelerini önlemek gerekmektedir. Bunu yaparken scratch benzeri blok mantığı kullanılan böyle bir ortamda öğrenci blokların hangi kodlara denk geldiğini görecektir. Böylelikle doğrudan kod yazmadan kısa animasyonlar tasarlanabilecektir. Bu amaçla araştırmacı JQuery kodlarının blok mantığıyla kullanılabilirdiği bir ortam tasarlamıştır. Araştırmacı bu şekilde geliştirilen bir ortamla öğrencilerin programlama mantığı öğrenmenin yanı sıra her platformda çalışabilecek kodlar tasarlayabilmesini amaçlamıştır.

1.5. Sayıtlar

Görüşme yapılırken öğrencilerin görüşme sorularına içtenlikle cevap verdikleri kabul edilmiştir.

1.6. Sınırlılıklar

Çalışma Kocaeli ilinin Başiskele ilçesinde 2018-2019 eğitim öğretim yılında Levent Kırca Oya Başar Ortaokulunda öğrenim gören 6. Sınıf öğrenciyle sınırlıdır.

1.7. Tanımlar

Kn: Katılımcı öğrenciler k1-k26 arasında numaralandırılmıştır. k1 bir numaralı katılımcı öğrenci manası taşımaktadır.

Web 2.0: 2004'te kullanılmaya başlayan, toplumsal iletişim sitelerini, vikileri, iletişim araçları ve internet kullanıcıları tarafından ortaklaşa oluşturulan sistemdir.

WWW: Açılımı 'World Wide Web' olan ve Türkçe' de 'Dünya Çapında Ağ' olarak karşılık bulan internet ortamında bilgi sunumu, aktarımı ve dağıtımını sağlayan servistir.
Moodle: E-öğrenme platformudur.

1.8. Kısaltmalar

JTBP: JQuery Tabanlı Blok Programlama

MEB: Milli Eğitim Bakanlığı

BTY: Bilişim Teknolojileri ve Yazılım

BÖLÜM II

ARAŞTIRMANIN KURAMSAL ÇERÇEVESİ VE İLGİLİ ARAŞTIRMALAR

2.1. Programlama nedir

Çok farklı şekillerde programlama için tanımlar bulunmaktadır. Bunlardan biri problemi tespit etmek, analiz etmek, çözümü tasarlamak ve gerçek kodlarla çözümü yazıp test etmektir (Alghamdi, 2017). Programlama akıllı olmayan bir bilgisayara net bir şekilde görevlerin yollandığı bir iletişim becerisidir (Hromovič, 2006'dan aktaran Saeli, Perrenet, Jochems, ve Zwaneveld, 2011). Hayatımızda bir sorunun çözümü için plan yaptığımız gibi bilgisayarla ilgili sorunların çözümü içinde önce bir plan yaparız. İşte bu planın bilgisayar diliyle ifade edilmiş haline programlama denir (Yolcu, 2018). Sorunun makineye anlatılması için komut dizinleri haline getirilmesi gerekir. Bu komut dizinlerinin derlenmesi ve çalıştırılmasına programlama denir (Sinap, 2017). Bir diğer ifade ile programlama, bilgisayara ne yapılması gerektiğini anlatan komut, kelime ve aritmetik işlemlerdir (Kesici ve Kocabaş, 2007'den aktaran Yolcu,2018). Bilgisayar programlama, bilgisayarın belirli bir görevi yerine getirmesi, sorunları çözmesi ve insan etkileşimini sağlaması için gerekli talimatların geliştirilmesini içeren bir süreç olarak tanımlanmaktadır (Balanskat ve Engelhardt, 2014)

Programlama hem bilgisayar biliminin temel bir becerisi ve bilişsel görevleri desteklemek için önemli bir araç hem de bilişimsel yetkinliklerin bir göstergesidir (Grover ve Pea, 2013). Programlama öğrencilerin üst düzey düşünme becerilerinin geliştirilmesi ve algoritmik problem çözme becerilerinin geliştirilmesini de sağlar (Fessakis, Gouli ve Mavroudi, 2013; Kafai, Pepler ve Chiu, 2007). Bilgisayar programları bilgisayara ne yapılacağını adım adım anlatır. Bilgisayar programı yazmak net ve dikkatli düşünmenizi gerektirdiği için, mantık, matematik, proje tabanlı öğrenme, problem çözme ve eleştirel düşünme becerilerini geliştirmekte programlama öğrenimi önemlidir (Sáez-López, Román-González ve Vázquez-Cano, 2016).

2.2. Programlama dilleri

Bilgisayarda yapılacak bir işin yapılması amacıyla bilgisayar komutları kullanılarak oluşan yapıya program, bu komut yapılarından oluşan sisteme de programlama dilleri denir. Bilgisayar biliminde önemli bir rolü olan programlama bireylerin kendilerini ifade etmelerine (Yolcu, 2018), üst düzey düşünme becerileri kullanılan bir süreç olduğundan kişinin çözüme kendi karakterini eklemesine imkân tanır (Armoni ve Gal- Ezer, 2014'ten aktaran Yolcu 2018).

Yazılımın en önemli öğelerinden biri olan programlama dili, programcının yapmak istediği şeyleri yani bir algoritmayı bilgisayara anlatması olarak tanımlanabilir (Karabak ve Güneş, 2013). Programlama dilleri her biri farklı amaçla geliştirilmiş C#, Java vb. gibi kendine has komutları ve kuralları bulunan yapılardır (Coşar, 2013). Programlama dillerinin düşük seviye makine dillerinden C++, Java gibi yüksek seviye dillere kadar pek çok seviyesi mevcut olmakla birlikte, bunlara ek olarak “script diller” olarak adlandırılan başka uygulamalarla yorumlanan “JavaScript” gibi diller de bulunmaktadır (Yanfei, 2009'dan aktaran Alghamdi, 2017). Programlama dillerinin serüveni 1943'te ortaya çıkan ENIAC Coding System ile başlamış, 1950'li yıllarda FORTRAN, LISP, ALGOL, CABOL, BASIC gibi dillerle devam etmiş ve 1970'li yıllarda ise Pascal, C bu dilleri takip etmiştir. 1980'lerden sonra teknolojinin gelişmesinin hız kazanmasıyla üst seviye programlama dilleri olan Delphi, Java, C++, C# dilleri ortaya çıkmıştır (Eryılmaz, 2003'ten aktaran Sinap, 2017).

Günümüzde pek çok programlama dili kullanılmaktadır. Bu diller genel olarak bazı sınıflara ayrılmakta olup şu şekilde sıralanabilir (Karabak ve Güneş, 2013) :

Tablo 1
Programlama Dilleri ve Sınıfları

Programlama dili sınıfları	Programlama dilleri
Yordamsal (Procedural) Diller	(Algol, C, Pascal, Matlab...)
Nesneye dayalı (Object Based) Diller	(Ada, Smalltalk, C++, Java, C#....)
Fonksiyonel (Functional) Diller	(Lisp, Haskell,...)
Mantıksal (Logic) Diller	(Prolog,...)

2.3. Dinamik programlama dilleri

2.3.1. Dinamik diller

World Wide Web'in artan popülaritesiyle beraber tasarımcılar dinamik dillere geri dönmektedir (Taivalaari, 2009). Web pek çok farklı sayfayı birbirine bağlayabilen esnek, merkezi bir kontrolü olmayan, canlı ve sürekli gelişen yapısıyla web uygulamalarına ve doğal olarak dinamik dillere geçişi artırmaktadır (Taivalaari, Mikkonen, Anttonen ve Salminen, 2011). WEB'de en yaygın kullanılan dinamik dillerden bazıları JavaScript, Perl, PHP, Python ve Ruby'dir (Mikkonen ve Taivalaari, 2007). Bu dillere yaşanan talebin gerekçesi uygulamaların kullanım ömrünün azalması ve uygulama geliştirme sürecinin hızıdır (Taivalaari, 2009). Bu talebin sonucunda bu alanda eğitim ihtiyacı ortaya çıkmaya başlamıştır (Anttonen, Salminen, Mikkonen ve Taivalaari, 2011).

Dinamik programlama dili terimi, geleneksel programlama dillerinde sadece derleme sırasında mevcut olan, bazı ortak çalıştırma özelliklerini paylaşan programlama dili sınıfını tanımlamaktadır (Mikkonen ve Taivalaari, 2007). Dinamik dillerin bu yerleşik özellikleri, çalışmakta olan programın sınıflarını, nesnelere ve diğer tanımlarını yeni kod yükleyerek veya oluşturarak genişletme özelliğini içerebilmektedir (Mikkonen ve Taivalaari, 2007).

2.3.2. Dinamik dil özellikleri

Dinamik bir dil, aşağıdaki özelliklerden bir veya daha fazlasına sahiptir. Bunlar;

1) Dinamik yazma: C, C++ ve Java gibi geleneksel nesneye dayalı programlama dillerinde, değişken ve parametre türleri derlenmeden önce açıkça atanmalıdır. Ancak çoğu dinamik dilde, kullanımdan önce değişkenlerin bildirilmesi gerekmez. Ayrıca, tür bilgileri gerekli olmadığı sürece çalışma zamanına kadar belirlenmez (Mikkonen ve Taivalaari, 2007).

2) Yorumlama: Geleneksel nesneye dayalı programlama dillerinde, kod yürütmeden önce ikili bir gösterime ya da başka bir ara formda derlenir. Dinamik dillerle, kaynak kodu çalışma zamanında okunur, ara gösterime veya makine koduna dinamik olarak çevrilir ve ardından derhal yürütülür. Son kullanıcının bakış açısından, tüm bu aşamalar kesintisiz ve otomatik olarak gerçekleşir (Mikkonen ve Taivalaari, 2007).

3) Çalışma zamanı: Geleneksel programlama dilleri kullanılırken, kod yapıları, örneğin dinamik olarak bağlanmış kütüphaneler (DLL'ler) veya diğer eklenti bileşenleri çalışma süresinde değişmemekteyken dinamik dillerle, sınıf hiyerarşileri ve programın diğer yapısal ve davranışsal yönleri çalışma zamanında değiştirilebilir. Örneğin, yeni işlevler ve değişkenler çalışırken sınıflara ve nesnelere eklenebilir (Mikkonen ve Taivalsaari, 2007).

2.4. JavaScript

JavaScript Netscape tarafından geliştirilmiş olan nesneye dayalı bir programlama dilidir (Sarısakal ve Uysal, 2001; Çay ve İşcan, 2002). Adına rağmen, JavaScript Java programlama diliyle tek ortak noktası hem Java hem de JavaScript'in C programlama diliyle paylaştığı söz dizimsel benzerliktir. (Çay ve İşcan, 2002). Bir HTML belgesinin tüm öğelerine erişim sağlamak için oluşturulmuş olan JavaScript Web tasarımında kullanılan temel teknolojilerden en eskisidir (Nixon, 2014). JavaScript, animasyonlu ve etkileşimli web sitelerinin oluşturulmasını desteklemek için hemen hemen tüm web tarayıcılarında bulunan teknolojiler topluluğu olan dinamik HTML'nin (DHTML) temel bir yapı taşıdır (Goodman, 2006'dan aktaran Mikkonen ve Taivalsaari, 2007). Ancak JavaScript HTML içine gömülü çalışmakta olduğundan öncelikle HTML öğrenmek gerekmektedir (Çay ve İşcan, 2002; Mahmoud, Dobosiewicz, Swayne, 2004).

Web tarayıcısının içine entegre edilen kitaplıklarla çalıştırılabildiği gibi ve temel JavaScript kodları da kitaplık gerekmeden çalışabilmektedir (Mikkonen ve Taivalsaari, 2007). JavaScript'te değişkenlerin kullanımlarından önce tanıtılması gerekmediği ve değişken türlerinin yürütme sırasında dinamik olarak karar verildiği tipik bir dinamik dildir (Mikkonen ve Taivalsaari, 2007). Web formlarındaki e-posta adresinin geçerliliğini kontrol etmek (Nixon, 2014), basit komut görevleri ve animasyonlar için kullanılmasına rağmen genel amaçlı bir programlama dilidir (Mikkonen ve Taivalsaari, 2007). Ancak, Ajax gibi Web 2.0 teknolojilerinin yaygınlaşmaya başlaması ile kitaplıklar oluşturuldukça, JavaScript hakkındaki olumsuz algı giderek düzelmektedir (Mikkonen ve Taivalsaari, 2007). JavaScript, CSS ile birleştirildiğinde, sayfanın yeniden yüklenmesine gerek olmadan değişen dinamik web sayfalarının tasarlanmasını sağlamaktadır (Nixon, 2014).

Bununla birlikte, farklı tarayıcı tasarımcılarının seçtiği yöntemler arasındaki bazı farklılıkların bulunması JavaScript kullanımını zora sokabilmektedir (Nixon, 2014). Yazılan kod farklı tarayıcılarda çalışmayabilmektedir.

Daha önce de belirtildiği gibi, JavaScript bir HTML belgesindeki çeşitli öğeler üzerinde dinamik kontrol sağlamak için geliştirilmiş bir programlama dilidir. Başlangıçta “Asenkron Java Script ve XML” anlamına gelen Ajax gittikçe daha fazla tasarımcı tarafından web sayfaların arka planına erişmek için JavaScript ile birlikte kullanılmaktadır (Nixon, 2014). Ajax, sayfaların aynı sayfa üzerinde yeniden yüklenmeden görüntülenmesini sağlayan, ekran üzerindeki butonlara tıklanıldığında ekranda bazı öğelerin güncellenmesini sağlayan Web 2.0 teknolojisidir (Nixon, 2014).

Günümüzde WEB hayatımızın ayrılmaz bir parçası haline geldiği için (Taivalsaari, Mikkonen, Ingalls, ve Palacz, 2008) kullandığımız pek çok uygulama da WEB üzerinde çalıştırılmak üzere tasarlanmaktadır. Bu durum yeni nesil programcıların JavaScript, Perl, PHP, Python ve Ruby gibi dillerle dönmesine ve bu dillerin ilgi görmesine neden olmuştur (Mikkonen ve Taivalsaari, 2007). Son birkaç yıl içinde özellikle Ajax, Adobe Integrated Runtime (AIR), Google Web Toolkit (GWT), Google Gears, JavaFX, Microsoft Silverlight, Ruby on Rails gibi sistemler tasarımcılar açısından gelecek vaat etmeye başlamıştır (Taivalsaari ve diğerleri, 2008). Bu sebeple her platformda çalışabilen uygulamalar tasarlamak için JavaScript öğrenmek önemlidir.

2.5. JQuery

JQuery, John Resig tarafından 2006'da daha az yaz, daha fazlasını yap sloganıyla oluşturulmuş hızlı ve özlü bir JavaScript Kütüphanesidir (Tutorialspoint, 2015). Kütüphane hızla tanındı ve web uygulamaları oluşturmak için kullanılan en popüler teknolojilerden biri haline gelmiştir (Król ve Szomorova, 2015). Uygulama geliştirme dünyasında etkili olan JQuery kütüphanesi, basit sözdizimi sayesinde bir zamanlar karmaşık olan işleri basit ve eğlenceli hale getirmektedir (Lindley, 2009). JQuery’i her yerde, Twitter ve Facebook gibi oldukça popüler sitelerde, Yahoo! veya Google gibi arama motorlarında ve şık efektlerle dolu web sitelerinde görülmektedir (Beighley, 2010). Günümüzde çoğu JavaScript web uygulaması temel tarayıcı API lerinin üstünde kullanışlı işlevler sağlamak

için kütüphaneler kullanmaktadır ve bu kütüphaneler genellikle genel dinamik dil özelliklerini yoğun olarak kullanmaktadırlar (Andreasen ve Møller, 2014). Birçok geliştirici zarafeti, netliği sayesinde hızlı bir şekilde JQuery kütüphanelerini tasarımlarına eklemiştir (Lindley, 2009). İyi bir geliştirme çerçevesi olarak JQuery, geliştiricilerin normalde 20 satır kodla yapılan işi 3 satırda yapmasına imkan veren bir JavaScript kütüphanesidir (Bibeault ve Katz, 2008). Bir araştırma ilk 10 milyonda ki web sitelerinin %58 nin JQuery kütüphanelerini kullandığını ve bunun piyasada JavaScript kütüphanelerinin %93.4 ünü oluşturduğunu göstermiştir (W3Techs, 2014'ten aktaran Andreasen ve Møller, 2014). Ocariza, Pattabiraman ve Zorn (2011) popüler web uygulamalarının tasarımında yaygın olarak kullanılan JavaScript kütüphanelerinin hatalarını inceledikleri çalışmalarında JQuery kütüphanesinin en popüler JavaScript kütüphanesi olduğunu belirtmektedirler.

Kütüphaneyi kullanmaya başlayan tasarımcılar projelerinize zengin ve etkileşimli deneyimler ekleyebilmektedir (Lindley, 2009). JQuery düzenli olarak gelişmekte ve yeni sürümleri ortaya çıkmaktadır. Bunun yanında binlerce eklentisi bulunmaktadır (Andreasen ve Møller, 2014). Kütüphane, çok fazla özellikte dolu olduğundan web sitelerini geliştirmek için kullanılan araçların çoğunda olduğu gibi, JQuery kütüphanesinin genişliğini ve derinliğini tam olarak anlamak mümkün olmayabilmektedir (Lindley, 2009).

Birçok web sitesinin kütüphane kodu uygulamadan daha büyük bir düzendedir (Andreasen ve Møller, 2014). İstemci tarafı web uygulamaları, web içeriğini değiştirmek için Belge Nesne Modeli (dom) olarak bilinen zengin, ancak düşük seviyeli bir API kullanılarak JavaScript'te yazılır (Lerner, Elbert, Li ve Krishnamurthi, 2013). Belgenin bölümlerini seçip bunlara ekleme, kaldırma veya değiştirme işlemleri yapılabilmektedir (Lerner ve diğerleri, 2013). JQuery kütüphanesi, Orijinal veya değiştirilmiş programı kullanmak, kopyalamak, değiştirmek ve dağıtmak (satış dahil) için sınırsız haklar veren Açık Kaynak GNU Genel Kamu Lisansı ve MIT (Lisans X11) altında mevcuttur. Lisans koşullarına saygı duyulması ve sadece yazar hakkında bilgi verilmesini halinde, bütün komut dosyalarının bir programlama tabanı biçiminde kullanılması mümkündür (Król ve Szomorova, 2015).

Lerner ve arkadaşları (2013) çalışmalarında, JQuery'yi üst seviye yapılarını doğrudan ortaya koyan, ham DOM programlarından farklı, kendi başına bir dil olarak savunmaktadırlar.

JQuery'nin avantajları arasında, standart XHTML belge yapısında kolayca deęişiklikler yapmaya imkân tanıyan birçok olasılık vardır (Król ve Szomorova, 2015). JQuery JavaScript Kütüphanesi kullanılarak web sitelerinin oluşturulma sürecini iyileştirebilmekte ve etkileşimleri ve dinamik içerik sunum biçimleri geliştirilerek çekicilięi artırabilmektedir (Verens 2012; Bennet ve dięerleri, 2014; David 2011; McFarland,2012; Li ve Li 2013'ten aktaran Król ve Szomorova, 2015). JQuery kısa ve okuması kolaydır. Benzersiz zincirleme kod yazımı modeli pek çok operasyonu tek sayfada yapmanızı sağlamaktadır (Bibeault ve Katz, 2008). JQuery'in ile belge üzerinde yapılabilecek işlemler üç geniş gruba ayrılır: sayfadaki ilk düęüm grubunu seçme, belgedeki bu düęümlere gitme ve düęümleri deęiştirme (Lerner ve dięerleri, 2013). Bu işlevler, programcının benzer veya ilgili birçok düęümü kolayca ve düzgün bir şekilde işlemlerini sağlar (Lerner ve dięerleri, 2013).

Örneęin “\$ (“#araba”).addClass (“myclass”).fadeIn ().load (“ajax_url”);” Bu kod yazım yöntemiyle bir satırda 3 farklı operasyonu tek satırda yapmak mümkündür.

JQuery, hızlı web geliştirme için HTML belge gezme, olay işleme, animasyon ve Ajax etkileşimlerini basitleştirir. JQuery, daha az kod yazarak çeşitli görevleri basitleştirmek için tasarlanmış bir JavaScript kütüphanesidir. JQuery tarafından desteklenen önemli temel özelliklerin şunlardır (Tutorialspoint, 2015) :

1. DOM manipölasyonu: JQuery, Sizzle adı verilen tarayıcılar arası açık kaynak kodlu bir seçici motor kullanarak DOM öğelerini seçmeyi, pazarlık etmeyi ve içeriğini deęiştirmeyi kolaylaştırdı.
2. Handling Olay işleme: JQuery, HTML kodunu olay işleyicileriyle karıştırmaya gerek kalmadan, bir bağlantıyı tıklayan bir kullanıcı gibi çok çeşitli olayları yakalamanın zarif bir yolunu sunar.
3. AJAX Desteęi: JQuery, AJAX teknolojisini kullanarak duyarlı ve zengin özelliklere sahip bir site geliştirmek için size çok yardımcı olur.
4. Animasyonlar: JQuery, web sitelerinizde kullanabileceğiniz birçok dahili animasyon efektiyle birlikte gelir.
5. Hafif: JQuery çok hafif bir kütüphanedir - yaklaşık 19 KB boyutundadır (küçültülmüş ve sıkıştırılmış).

6. Çapraz Tarayıcı Desteği: JQuery çapraz tarayıcı desteğine sahiptir ve IE 6.0+, FF 2.0+, Safari 3.0+, Chrome ve Opera 9.0+ sürümlerinde iyi çalışır.
7. En Son Teknoloji: JQuery, CSS3 seçicileri ve temel XPath sözdizimini destekler (Tutorialspoint, 2015).

2.6. Programlama eğitimi

Bazı öğretmenler ilkokul öğrencilerinde kodlamanın küresel yararlarını ve avantajlarını göz ardı ederek, programlamanın kariyer olanakları ile ilgili olduğunu düşünse de programlamanın önemini ortaya koyan çalışmalar arttıkça ilköğretim ve lise seviyesinde programlama konularının eğitime bütünleştirilmesi ilgili adımlar atılmaktadır (Sáez-López ve diğerleri, 2016). 2011'de Estonya'da kodlamayı müfredata entegre etmeye yönelik ulusal bir girişim yapılmıştır (Sáez-López ve diğerleri, 2016; Demirer ve Sak, 2016; Mironova, Amitan ve Vilipöld, 2017). Birleşik Krallık 2014 yılından itibaren, ilk ve ortaokullarda çocuklara bilgisayar programlama öğretilmesi, mantıklı düşünme gibi becerileri kazandırmak için ulusal bir çalışma uygulamaktadır (Alghamdi, 2017). Ayrıca, Finlandiya da programlama eğitimi 6-7 yaşlarında başlamaktadır (Akpınar ve Altun, 2014). Son yıllarda, özellikle ortaöğretimde, Bilgisayar Bilimi'ne olan ilgi büyük ölçüde artmıştır (Mironova ve diğerleri, 2017).

Kuzey İrlanda için gözden geçirilmiş müfredat, gençlerin potansiyellerini geliştirmelerini ve yaşamları boyunca bilinçli ve sorumlu seçimler ve kararlar vermelerini sağlamak amacıyla Eylül 2007'den beri uygulanmaktadır (CCEA, 2007, s.4'ten aktaran Gibson ve Bradley, 2017). Programlama temel kavramları bilişim alanında bir kariyer yapma yapmalarına bakılmaksızın birey için değerli olacak beceriler kazanılmasını sağlamaktadır (Balanskat ve Engelhardt, 2014; Bers ve diğerleri, 2014). Papert (1980) bu görüşü ayrıca, özellikle analiz ve hata ayıklama alanlarında derin bir programlama anlayışının, bilgisayar ve bilgi teknolojisi ile ilgisi olmayanlar da dâhil olmak üzere birçok söylem alanında önemli eğitimsel faydalarla sonuçlanacağını iddia ederek bu görüşü desteklemektedir (aktaran Gibson ve Bradley, 2017). Programlama dilleri düşünme becerilerini hedef alan yapılarıyla bilişsel gelişime büyük katkı sunmaktadır (Kert ve Uğraş, 2009). Fluck ve arkadaşları (2016), programlama kavramlarının öğrenilmesi

önündeki engellerin kaldırılmasının sadece bireyler olarak öğrencilere fayda sağlamayacağını, sonunda ekonomiye fayda sağlayacağını ileri sürmektedir. Örneğin, bu teknoloji tarafından yönlendirilen bir dünyada rekabet gücünü koruyacak bilgisayar bilimcilerin gelişimi yoluyla ortaya çıkabilir (Gibson ve Bradley, 2017). Bunun yanı sıra programlama öğrenmenin öğrencilerin üst düzey düşünme becerileri kazanmalarına da yardımcı olacağı ve bu yolla diğer derslerde de başarılarına olumlu etki edeceği düşünülmektedir (Lau ve Yuen, 2009; Karabak ve Güneş, 2013).

Feldgen and Clúa (2004) öğrencilerinin çoğunun programlamanın iş alanında ne kadar önemli bir yetenek olduğunun farkında olmadıklarını, kazançlı bir iş geleceği için öneminin farkında olsalar öğrenmek için daha fazla güdülenmiş olacaklarını ifade etmişlerdir. Bilgisayar programlama işin bir parçası olarak günümüzde kod yazma ve etkileşimi şeklinde birçok iş alanında kullanılmaktadır. Günümüz şirketlerinin %70 inin bilgisayar yeterliliğini orta ve üst yönetim pozisyonları için gerekli gördüklerini ve bilgisayar okuryazarlığının iş alanları için öneminin çok arttığını belirtmektedirler (Charters, Lee, Ko ve Loksa, 2014).

Estonya da programlama eğitimi ve Scratch konusunda öğretmenlerin yeterli olmadığı, programlama eğitimi verilebilmesi için önce öğretmenlerin eğitimden geçirilmesi gerektiği bildirilmektedir (Mironova ve diğerleri, 2017). Estonya'da yeni başlayanlar için programlama dersi 2011 yılından itibaren orta öğretim okullarının müfredatına dahil edilmiştir. Bu ders Tallinn Teknoloji Üniversitesi Enformatik Bölümünün, öğrencilere temel programlama eğitimi verme çabalarının sonucunda ortaya çıkan tecrübeler üzerinden üretilmiştir. Kursun gelişiminde yazarlar ve eğitimciler bilgisayar eğitiminde zamanın eğilimlerine ayak uydurmaya çalışmıştır (National Science Foundation, (t.y.) aktaran Mironova ve diğerleri, 2017).

Günümüze kadar lisans ve lise düzeyinde öğretilen programlama becerisi günümüzde teknolojinin gelişmesi ile ortaokul hatta ilkokulda öğrencilere kazandırılması beklenen beceriler içerisinde yer almaktadır (Yolcu, 2018). Gelecekte öğrencilerin sahip olması beklenen beceriler 21. yüzyıl becerileri olarak adlandırılmaktadır (Sayın ve Seferoğlu, 2016'dan aktaran Yolcu, 2018). Yolcu (2018) bu becerilerden bazılarını; yaratıcılık, inovasyon, kritik düşünme, problem çözme, iletişim, işbirlikli çalışma, bilgi ve teknoloji okuryazarlığı, küresel yetkinlikler olarak sıralamaktadır. Programlama becerisi bilgi ve

teknoloji okuryazarlığı kapsamında değerlendirilse de bu becerilerin başında gelmektedir (Yolcu, 2018).

Nasıl programlanacağını öğrenme sürecinde öğrenciler matematiği ve değişkenler, döngüler veya şartlar gibi bilişimsel kavramları öğrenmektedirler. Bu uygulamalar, ilköğretimde proje bazında öğrenme yoluyla matematik, dil, sanat eğitimi, fen bilimleri ve sosyal bilimler alanlarında uygulanabilmektedir (Sengupta, Kinnebrew, Basu, Biswas, ve Clark, 2013). Sengupta ve arkadaşları (2013), bilgisayar biliminin matematik ve fen bilimleri derslerinde nasıl ve niçin entegre edilmeleri gerektiğini göstermek için, bilişimsel düşünce kavramlarının bilimsel araştırma ile ortak etkilerini belirlemek için geliştirdikleri çerçevenin, bir ortaokul fen dersinde uygulamanın etkinliğini göstermişlerdir.

Öğretmenler ve öğrenciler, programlamak için kavramların yüksek düzeyde soyutlanmasından dolayı programlamanın çok karmaşık olduğu algısına sahiptir. Scratch'in yaratıcıları, farklı türdeki projeleri farklı bağlamlarda eğlenceli, anlamlı ve sosyal bir programlama dili ile kapsamanın mümkün olduğuna inanıyor (Resnick, Maloney, Monroy-Hernández, Rusk, Eastmond, Brennan, Millner, Rosenbaum, Silver, Silverman ve Kafai, 2009). Papert (1980), programlama dillerinin “alçak taban” (başlaması kolay) ve “yüksek tavan” (karmaşık projeler) özelliğinde olması gerektiğini savundu (aktaran Sáez-López ve diğerleri, 2016).

2.7. Görsel programlama dilleri uygulamaları

Görsel programlama birçok sorunu çözerek ilkokulda kodlamaya izin verir. Bloklarla (Scratch gibi) görsel programlama, yalnızca şekilleri nedeniyle mantıklı olan biçimlerde yerine getirilebilir, bu nedenle derleyiciden hata mesajları almak mümkün değildir. Bu, giriş seviye programlama dersi için mükemmel bir rahatlama ve öğrenciyi geleneksel metin tabanlı dillerin verdiği gönül yarasından kurtarır (Wilson ve Moffat, 2010'dan aktaran Sáez-López ve diğerleri, 2016). Programlama öğrenimi için başlangıç öğrenme ortamlarından en popülerleri olan Alice ve Scratch öne çıkmaktadır (Silla, Przybysz ve Leal, 2016; de Kereki ve Manataki, 2016). Bunun haricinde Greenfoot, Processing (Silla, Przybysz ve Leal, 2016) ve AppInventor (de Kereki ve Manataki, 2016) kullanılmaktadır.

Bu programlama dilleri genelde tamamen görsel olup, kullanıcıların blokları sürükleyerek programlar oluşturdukları "sürükle ve bırak programlama" ya imkan tanır. Oluşan çoklu blok yapıları, tüm komut dosyalarına veya programlara karşılık gelir. Bu diller özellikle genç öğrenciler için faydalıdır çünkü yazım gerektirmez. Sözdizimi hatalarını en aza indirir ve endüstri standardı metinsel tabanlı dillere kıyasla daha çekici ve görseldirler (de Kereki ve Manataki, 2016).

Teknolojinin evrimiyle programcı ihtiyacı ve bulunabilirliği arasındaki boşluk artmakta olduğundan bilgisayar programlama öğrenmek bir gerekliliktir (Hussain, Fergus, Al-Jumeily, Pich ve Hind, 2015). Genç öğrencilerin kodlamayı öğrenmelerine yardımcı olan girişimler ve çevrimiçi kaynaklar artmaktadır. Programlama eğitiminde görsellik günümüzde yaygınlaşmaktadır. Bu amaçla geliştirilen yazılımların sayısı ve çeşitliliği hızla artmaktadır. Algoritma ve programlama mantığını animasyonlarla süsleyen, basitleştirilmiş ortam tasarımları geleneksel yöntemde yaşanan sıkıntılardan kurtulmaya yardımcı olmaktadır (Coşar, 2013). İngiltere gibi bazı ülkelerin bilgisayar bilimi dersini zorunlu ders yaptığı bir dönemde programlama, her bireyin erişebileceği bir hale getirilmelidir (Good ve Howland, 2015; Hussain, Fergus, Al-Jumeily, Pich ve Hind, 2015). Yapısal olarak tüm programlama dilleri birbirine benzediğinden basit yapıya sahip bir programlama dili öğrenerek öğrenilen bilgiler diğer programlama dillerine uyarlanabilmektedir (Arabacıoğlu, 2006'dan aktaran Coşar, 2013). Okul ortamında öğrencilerin öğreneceği bir programlama dili çok da karışık olmamalıdır (Good ve Howland, 2015).

Bu amaç doğrultusunda yapılan çalışmalar sonucu blok tabanlı programlama araçları olarak isimlendirilen klasik programlama dillerinden farklı olan yapılar ortaya çıkmaktadır (Yolcu, 2018). Bu programlama araçlarının temel özelliği görselleştirilmiş blok kod yapısı, ara yüzünün kolay olması, çevrimiçi paylaşım imkânlarının bulunmasıdır (Resnick ve diğerleri, 2009). Blok tabanlı uygulamaların çoğunlukla programlamaya yeni başlayan öğrenciler açısından etkili olduğu sonucuna varılan pek çok çalışma bulunmaktadır (Kalelioğlu ve Gülbahar, 2014). Bu çalışmalarda öğrencilerin programlama motivasyonu, programlamanın temel yapılarını kavrama, bilgi-işlemsel düşünme, özgüven, algoritma becerisi gibi konularda blok tabanlı eğitimin olumlu değişimler sağladığı ifade edilmektedir. (Kukul ve Gökçearsan, 2014; Howland ve Good, 2015; Yükseltürk ve Altıok, 2016).

2.7.1. Scratch

Scratch, MIT Media Lab'daki Yaşam Boyu Anaokulu grubu tarafından oluşturulan bir programlama dilidir. Scratch programlama ortamı ve dili, öğrenmesi son derece hızlı olan ve kullanıcıları yıllarca meşgul tutmak için yeterli derinlik ve çeşitlilikte bir sistem olarak ortaya çıkmaktadır (Maloney ve diğerleri, 2010, s. 14). Scratch programlamanın temel taşı olan algoritmanın çocuklara öğretimi için oldukça kolay, eğlenceli ve ücretsiz bir yazılımdır (Kert ve uğraş 2009). Bu arayüzde çocuklar kodlama, kod oluşturma için renkli bloklarla etkileşime girdiklerinden geleneksel programlama dillerinden daha kolaydır (Sáez-López ve diğerleri, 2016).

Çocuklar Scratch'ı kendilerine sunulan diğer blok programlama çeşitlerine tercih etmektedir. Scratch'ın yaratıcılığı teşvik etmede ve multimedya ürünlerinin sosyal paylaşımında çok başarılı olduğu anlaşılıyor (Weintrop ve diğerleri, 2015; Wilson ve Moffat, 2010'dan aktaran Sáez-López ve diğerleri, 2016). Bu uygulama, sınırlı sayıda programlama bilgisi olan veya hiç öğrenmeyen, öğrenme için erişilebilir bir başlangıç noktası sağlamak için genç öğrencilerin katılımını sağlamayı amaçlamaktadır (Good, 2011'den aktaran Sáez-López ve diğerleri, 2016).

Scratch programlama dili, sekiz farklı kategoride (hareket, görünüm, ses, kalem, kontrol, algılama, operatörler ve değişkenler) gruplandırılmış 100'den fazla programlama bloğu sunar (Sáez-López ve diğerleri, 2016). Scratch, kendi etkileşimli hikayelerinizi, oyunlarınızı ve animasyonlarınızı oluşturabileceğiniz, ücretsiz, açık kaynaklı, medya açısından zengin, bu ürünleri çevrimiçi bir toplulukta dünyadaki diğer genç programcılarla paylaşmanızı sağlayan (Alghamdi, 2017) bir programlama ortamıdır (Good, 2011'den aktaran Sáez-López ve diğerleri, 2016). Scratch açık kaynak kodlu olduğundan dolayı siteye yüklenen projeler kendi bilgisayarlarınıza indirilip incelenebilmekte ve diğer üyelerle işbirliği ile bu projeler geliştirilip yeni projeler oluşturulabilmektedir (Karabak ve Güneş, 2013).

Scratch, etkileşimli projeleri programlayıp paylaşarak öğrencilerin önemli matematiksel ve bilgisayarla ilgili kavramların öğrenilmesini sağlamanın yanı sıra 21. yy becerileri olan, yaratıcı düşünme, sistematik düşünme ve işbirlikli çalışmanızı sağlamaktadır. (Resnick ve diğerleri, 2009). Bu program aynı zamanda öğrencilerin animasyonlar, oyunlar, arayüzler

ve bilişimsel kavramların ve bilişimsel uygulamaların anlaşılmasını genişletebilecek sunumlarla ilgili programlar oluşturmalarına ve geliştirmelerine izin verir (Good, 2011'den aktaran Sáez-López ve diğerleri, 2016). Çocuklar, programı programlama ortamının kendi web adresinden ücretsiz olarak indirilebilir ve hazırladıkları projeleri bu siteden paylaşabilirler. Yapılan projeler dil düğmesi ile desteklenen 42 dilden birine çevrilebildiği için çocuklar herhangi bir dil engeline takılmadan iletişimlerini sürdürebilirler (Karabak ve Güneş, 2013). Scratch sayesinde öğrencilerin programlama kavramlarını görsel ve sezgisel bir dilde kullanmaları amaçlanmaktadır, çünkü yönetim, ürünle sonuçlanan farklı renk ve komutlardan oluşan bloklar yerleştirerek gerçekleştirilir.

Bu görsel ortam, kullanıcıların ilköğretim okulunda çeşitli düzeylerde eğitim ortamlarında araştırma yapmalarını ve oluşturmalarını sağlayan sezgisel bir sürükle ve bırak yöntemi sunar (Good, 2011'den aktaran Sáez-López ve diğerleri, 2016). Scratch çocukların oyunlar, hikayeler ve animasyonlar tasarlayarak programlama öğrenebildikleri bir araçtır. Bu araçlar programlama kavramlarının öğretilmesinde etkili araçlardır (Alghamdi, 2017).

Çocuklar, Scratch kullanarak yaratıcı ve işbirliğine dayalı düşünmeyi öğrenir. (Sáez-López ve diğerleri, 2016). Scratch yazılımı kolayca oyun, animasyon ve simülasyon tasarlamaya imkan tanıyan, noktalama işaretleri içermeyen, basitçe kod bloklarını süreklemek yoluyla tasarım yapmaya imkan tanıyan basit arayüzü oldukça başarılı bulunan bir yazılımdır (Karabak ve Güneş, 2013). Bunun yanısıra çocuklar için kendi istedikleri şekilde oyun tasarımı yapmaya imkân tanıyan arayüzü öğrencilerin oyun yoluyla daha etkin öğrenmelerine de imkân tanımaktadır. Çünkü oyun yoluyla öğrenme, öğrencilerin derin öğrenmeler elde etmelerine imkân tanıyan en etkili öğrenme yöntemlerinden biridir (Erekmeççi ve Fidan, 2012).

Bu program ile koordinatlar, değişkenler ve rasgele sayılar gibi bazı matematiksel kavramları ve tekrarlama, koşullu ifadeler gibi temel bilişimsel kavramlarını öğrenebilirler. Öğrenciler, sadece matematiksel ve bilişimsel kavramları öğrenmekle kalmaz aynı zamanda yaratıcı düşünme, analiz etme, sistematik deney gibi öğrenme kabiliyetlerini de geliştirirler (Monroy-Hernández ve Resnick, 2007).

Scratch ile çocuklar kendi tasarımlarını yaparak, gerçek hayatta karşılaştıkları problemler karşısında yaratıcı çözümler ortaya koyabilirler (Karabak ve Güneş, 2013). Scratch çocukların etkileşimli hikâyeleri, oyunları ve animasyonları programlayıp paylaşabildiği

çevrimiçi bir topluluktur. (Monroy-Hernández ve Resnick, 2007). Kullanıcılar belirli bir programlama dili bilmek zorunda değildirler. Kod bloklarını doğru sırada bir araya getirerek program yazılmış olmaktadır. Böyle kolay ve anlaşılabilir bir arayüze sahip olduğu için çocuklar ilkokul öğrencileri bile kolaylıkla kullanabilirler. Scratch en çekici özelliği kolaylıkla oyun üretilebilmesidir (Nikiforos, Kontomaris ve Chorianopoulos, 2013). Öğrenciler ürettikleri oyunları kolaylıkla Scratch topluluğuyla paylaşabilirler. Topluluğun üyeleri de bu oyunları gözden geçirip iyileştirebilirler. Topluluk üyelerinin göden geçirme ve iyileştirme avantajının programlama öğretiminde kullanılması Resnick tarafından önerilmiştir (Brennan ve Resnick, 2012). Var olan bir oyunun gözden geçirilip iyileştirilerek bir oyun üretilmesi yeni baştan yenisinin yapılmasına tercih edilmektedir (Brennan ve Resnick, 2012).

2.7.2. Alice

Alice, 3 boyutlu sanal dünyalar, genellikle kısa animasyonlu filmler veya oyunlar oluşturmak için kullanılan bir programlama sistemidir. Alice'te kullanıcılar, grafik komut satırlarını sürükleyip bırakarak ve açılır menülerden parametreleri seçerek programlar oluşturmaktadır (Kelleher ve Pausch, 2003). Alice isimli üç boyutlu etkileşimli animasyon aracı ise üç aşamada Carnegie Mellon Üniversitesi bünyesinde ve Randy Pausch yönetiminde geliştirilmiştir. Kullanıcıların kod öğelerini sürükleyip bırakarak programlar oluşturması sözdizimi hatası yapma olasılığını ortadan kaldırmaktadır (Kelleher ve Pausch, 2003; Kelleher, Pausch, ve Kiesler, 2007). Alice çalıştırıldığında, kullanıcılar programlarının yürütülmesini izlemekte ve hatalarını animasyon yardımıyla görmektedir. Ayrıca, Alice 2'nin kolej düzeyinde kullanılması öğrencilerin daha sonra Java'da öğretilen bilgisayar bilimleri derslerinde başarılı olmalarına yardımcı olmaktadır (Kelleher, Pausch, ve Kiesler, 2007).

Powers, Ecott ve Hirshfield (2007) çalışmalarında bilgisayar bilimine giriş dersinde Alice kullandıklarında öğrencilerin sandıkları kadar zorlanmadıklarını ifade ettiklerini bildirmişlerdir. Öğrenciler C++ veya java ya geçtiklerinde noktalama işaretleriyle uğraşmaktan dolayı kolaylıkla yılgınlığa düştüklerini ve bilgisayar bilimi için gerekli becerilere sahip olmadıklarını düşünmeye başladığını ifade etmişlerdir (Powers, Ecott ve

Hirshfield, 2007). Powers, Ecott ve Hirshfield (2007), Alice kullanırken öğrencilerin kazandıkları özgüvenlerinin java veya c++ kullanırken kaybolduğunu belirtmektedirler. Saint Joseph Üniversitesi ve Ithaca Kolejinde yürütülen çalışmalarda, Alice ile “Bilgisayar Bilimi1” dersi eğitimi alan öğrenciler kursu beğendiklerini, özgüvenlerinin yüksek olduğunu, uygulamayla haftalık 6-9 saat arası süre geçirdiklerini ifade ederken eğitimciler öğrencilerin uygulamaya katılım oranının yüksek olduğunu belirtmişlerdir (Cooper, Dann ve Pausch, 2003).

2.7.3. Code.studio.org

Para amaçlı olmayan bir çevrimiçi öğrenme platformu olan Code.org (Du, Wimmer ve Rada, 2016), öğrencilere döngü, olay, şart, algoritma, dijital vatandaşlık ve daha fazlasını hem ev hem sınıf ortamında öğrenme imkanı tanımaktadır (Code.org, 2014’ten aktaran Karaoğlu, 2018). Code.org platformu, Scratch’ın halefidir. Yaratıcılık ve bilişimsel düşünme gibi beceri ve yeteneklerin erken dönemlerde geliştirilmesine olanak sağlamak için tasarlanmıştır (Molina-García ve diğerleri, 2016). Code.org ilham verici videolar ve gençlerin programlama bloklarını bir araya getirerek basit programlar oluşturmalarını sağlayan kısa dersler sunmaktadır (de Kereki ve Manataki, 2016). Code.org organizasyonu ABD Başkanı Barack Obama, Microsoft Başkanı Bill Gates, Facebook’un kurucusu Mark Zuckerberg gibi alanında başarılı girişimcilerle ilham verici videolar çekmişlerdir. Bu videolar 15 bin okul tarafından fark edilmiş ve büyük destek toplamıştır (Demirer ve Sak, 2016). Codecademy çevrimiçi etkileşimli platform, Java ve Python gibi yaygın olarak kullanılan programlama dilleri için kodlama sınıfları sunmaktadır (de Kereki ve Manataki, 2016). Code.org platformu, Amerika Birleşik Devletleri’nde bir girişimi olan ve 180’den fazla ülkeye dersler vererek, her seviyedeki okullarda programlama öğrenmeyi tanıtmak olan “The Hour of Code” programını başlatmıştır (Kumar, 2014’ten aktaran Molina-García ve diğerleri, 2016).

2.7.4. App inventor

App Inventor Google tarafından geliştirilmiş kod yazmadan uygulama geliştirmeyi sağlamak amacıyla üretilmiş ücretsiz bir yazılımdır (Ericson ve McKlin, 2012; Coşar 2013). Temelde bir yapboz mantığıyla işlem yapan App Inventor, açılan pencere içerisinde sunulan nesnelere birleştirilerek, kod yazmadan uygulama geliştirmeyi sağlamaktadır (Coşar, 2013). Öğrenciler sürükle ve bırak mantığıyla mobil cihazlar için programlama yapmaktan zevk aldılar. Buna mole mash isimli uygulama ve sürükle bırak sınavı da dahildir (Ericson ve McKlin, 2012). App Inventor, android cihazlar için mobil uygulamalar oluşturulmasını sağlamaktadır. Google pilot programının bir parçası olarak, App Inventor San Francisco Üniversitesi'nde oluşturuldu ve öğrenciler aygıtları için uygulamalar oluşturabildiğinden ilgi toplamıştır (Wolber, 2011'den aktaran Du, Wimmer ve Rada, 2016).

2.7.5. Greenfoot

Greenfoot COSMOS projesinin oyun programlama kursları kapsamında 2009 yılında python programından uyarlanmıştır (Smith ve Sullivan, 2012). Python kullanırken öğrenciler kodlarının nasıl çalışacağını bilmeden kodlarını yazmaktaydı (Smith ve Sullivan, 2012). Greenfoot'da standart Java sözdizimini kullanıldığından çocuklar için oldukça zor olabilmektedir (Utting ve diğerleri, 2010).

10 yaşındaki öğrenciler ile yapılan uygulamalarda sözdizimi hataları ile ilgili sorunlar yaşandığı, bu sorunun öğrenciler tarafından düzeltilemediği, ancak 13 yaş gurubu öğrencilerinin bunun üstesinden gelebildiği bildirilmiştir. Buna rağmen temel programlama kavramlarının kazandırılması için 10 yaş gurubunda Greenfoot 'un başarılı olduğu bildirilmiştir (Utting ve diğerleri, 2010).

Maloney ve diğerleri (2010) Greenfoot'un tıpkı Alice gibi hedeflediği yaş grubunun Scratch'ın hedeflediği yaş gurubundan büyük olduğunu ifade etmektedir. Bu sebeple java veya java kavramlarının verilebildiği daha gelişmiş bilgisayar bilimi dersleri için daha uygun olduğu ifade edilmiştir (Maloney ve diğerleri, 2010).

2.7.6. Diğer

Çocuklara programlama eğitimi için üretilmiş yazılımların ilki “Logo”, Lisp ile ilgili programlama kavramlarının öğretilmesinde kullanılmaktaydı (Pea, 1983’ten aktaran Alghamdi, 2017).

JAWAA, programlama mantığını web üzerinde kolayca animasyonlar oluşturarak öğretmek için oluşturulmuş bir betik dilidir. Java’da yazılan program, kullanıcıların animasyon yazabilecekleri ve daha sonra Java’yı destekleyen herhangi bir web tarayıcısında gösterebilecekleri bir arabirim sağlar. Animasyonlar, kullanıcılar tarafından programlama deneyimi çok az olan veya hiç olmayan insanlar tarafından kolayca öğrenilebilen basit bir betik dilinde yazılmıştır. Daha ileri düzey kullanıcılar için, dizi, yığın, sıra, grafik veya ağaç gibi bir veri yapısının animasyonunu hızlı bir şekilde üretmek üzere programlarına JAWAA komutları eklenebilir. Yani, bir web sayfasından çağrılan ve animasyonu üreten bir JAWAA çıkış dosyası üretmek için herhangi bir programlama dilinde yazılmış bir programa JAWAA komutları eklenebilir (Rodger, 2002).

Nicolas Guibert ve Patrick Girard, Teaching and Learning with a Programming by Example System adlı çalışmalarında üniversite öğrencilerine zorunlu programlama kavramlarını öğretmek için soyut tanımlar yerine, somut örnekler kullanan MELBA (Metaphor-Based Environment to Learn to Built Algorithms), adlı sistemi tasarlamışlardır. MELBA sisteminde kullanılan mecazi dil ile programlama mantığı öğreticisinde kullanılan uygulama dili birbirlerine benzemektedir (Guibert and Patric, 2003’ten aktaran Coşar, 2013).

Small Basic: Small Basic, programlama öğretimine yeni başlayanlar için özellikle çocuklar için geliştirilmiş olan kolay, anlaşılır ve eğlenceli bir programlama dilidir. Small Basic’in amacı, programlama dil öğretiminin zorluklarını aşarak daha gelişmiş programlama dillerinin öğretimine geçiş sağlamaktır (Coşar, 2013).

Yerli yazılımlardan birisi Arabacıoğlu ve arkadaşları tarafından geliştirilen “PROGMAN”dır. Bu yazılım ile Türkçe bir programlama mantığı geliştirmiştir. Temel programlama komutları kullanıcıya düğmeler halinde sunulur. Bir kod penceresi vardır, bu pencerede seçilen ve işlenen kodlar gösterilir. Programın en önemli özelliği kullanılan

komutlar, deęişkenleri listeleyebilmesi ve kısmen de olsa kodları çalıştırabilmesi sonucunda çıktıyı gösterebilmesidir (Durak, 2009).

Good ve Howland (2015) çalışmalarında "Herkes için programlama" bağlamında, "doęal" bir programlama dili (natural programming language) tasarlama motivasyonu ile Neverwinter Nights 2 (NWN2) elektron araç takımını kullanmışlardır. Araç seti, uzman becerileri olmayan gençlerin hızlı ve kolay bir oyun oluşturma süreci başlatmasına ve motive edici özelliklere sahip olmasına olanak tanımaktadır. Çünkü oluşturulan oyunlar profesyonel olarak geliştirilmiş 3D ticari oyunlara benzemektedir. Oyun yaratma, gençlerin teknolojinin üreticisi olmasını sağlar, onları bilgisayarla ilgili kavramlara sokar ve programlama becerilerini geliştirmelerine yardımcı olur. (Good ve Howland, 2015).

2.8. Programlama yönelik tutum

Smith (1968) tutumu, kişiye atfedilen ve kişinin bir objeye ilgili düşünce, duygu ve davranışlarını oluşturan eğilim olarak tanımlamaktadır (aktaran Başer, 2013). Tutum, bir nesne veya olguya karşı hazır olma durumudur (Gökdaş, 2008). Tutumlar, beklenen davranış algıları ile birlikte birinin bilgisayar programlama öğrenmek gibi belirli bir aktiviteye nasıl tepki verebileceğini belirledikleri için önemlidir (Jay ve Willis, 1992). Tutum nasıl öğreneceğimizi ve davranışlarımızı etkilemektedir (Maio ve Haddock, 2009'dan aktaran Özyurt ve Özyurt, 2015). Tutum doğrudan gözlenememekle beraber (Arkonaç, 2005'ten aktaran Gökdaş, 2008) yine de öğrencilerin başarısını etkileyebilmektedir (Başer, 2013). Örneğin, bilgisayarlara yönelik olumlu tutumların, gençlerin ve yaşlı yetişkinlerin bilgisayar kullanımını ve bilgisayarla ilgili sonraki becerileri anlamalarını arttırdığı gösterilmiştir (Utting, Cooper, Kölling, Maloney ve Resnick, 2010).

Yapılan araştırmalar son yıllarda bilgisayar bilimlerini seçen öğrencilerin hızla azaldığını göstermektedir (Shashaani, 1997; Hoegh ve Moskal, 2009; Heersink ve Moskal, 2010; Başer, 2013; Charters ve diğerleri, 2014). Programlama eğitimi üst düzey zihinsel beceriler gerektirdiğinden ilk defa bu eğitimi alan bireylerde olumsuz bir bakış açısına sebep olmaktadır (Bayman ve Mayer, 1988'den aktaran Yolcu, 2018). Ortaya konan sorunların

biri öğrencilerin yazılım geliştirme işine sıcak bakmamaları ve ilgi duymamalarıdır. (Charters ve diğerleri, 2014). Bu algılar kişilerin programlamaya yönelik olumsuz tutum geliştirmelerine ve bireylerin programlamayı gerekli bir yeterlilik olarak görmelerini ve öğrenmelerini engellemektedir (Charters ve diğerleri, 2014).

Programlama giriş dersleri eğitiminde C++, Java gibi diller yaygın olarak kullanılmaktadır. Ancak bu diller özellikle altyapısı olmayan öğrenciler için oldukça zordur (Mahmoud ve diğerleri, 2004). Bazıları programlamanın çok zor olduğuna (Mahmoud ve diğerleri, 2004; Feldgen ve Clúa, 2004; Fesakis ve Serafeim, 2009; Yiğit, 2016) ve gerekli nitelikler olan mantıksal düşünme ve zekaya sahip olmadıklarına (Beckwith, Inman, Rector ve Burnett, 2007'den aktaran Charters, 2014), bazıları ise programcıların asosyal ve teknolojiyle takıntılı, iletişim becerileri olmayan, tuhaf insanlar olduğuna inanmaktadırlar (Charters ve diğerleri, 2014; Rodger ve diğerleri, 2009; Fisher ve Margolis, 2002). Bazı öğrenciler bilgisayar biliminden sıkıcı (Kelleher, Pausch and Kiesler, 2007), kafa karıştırıcı ve ustalaşmanın çok zor olduğunu düşündüklerinden sıklıkla vazgeçerler (Wilson ve Moffat, 2010'den aktaran Sáez-López ve diğerleri, 2016). Geleneksel metotlarla programlama eğitiminin öğrencileri cesaretlendirmek yerine daha da geri çektiği, (Mahmoud ve diğerleri, 2004) iki yıllık bir eğitimden sonra bile programlama algısının düşük olduğu bildirilmiştir (Kurland ve diğerleri 1989'dan aktaran Saeli ve diğerleri, 2011). Programlama konusunun lisans seviyesinde ki öğrenciler için bile ne kadar zor olduğu ve lisans seviyesinde ki öğrencilerin bile tutumlarının programlamaya yönelik olarak iyi olmadığı ortaya konmuştur (Başer, 2013). Farklı seviyelerde olan öğrenciler için programlama dillerini öğrenmek zor olabilmektedir. İster üniversite öğrencisi olsun ister ilkökul öğrencisi eğer öğrenci daha önce programlama eğitiminin temellerini almadıysa bazı zorluklarla karşılaşacaktır (Alghamdi, 2017).

Amerika Birleşik Devletleri, Kanada ve diğer ülkelerdeki üniversiteler, bilgisayar programlama giriş kurslarında % 50'ye ulaşan geri çekilme, başarısızlık ve D dereceli ders notları bildirmişlerdir (Hermann ve diğerleri, 2003'ten aktaran Mahmoud, Dobosiewicz, Swayne, 2004). Bilgisayar bilimine giriş derslerinin birçok öğrenciyi daha fazla öğrenmeye teşvik etmekten çok daha fazla sınırlendirmesinin daha muhtemel olduğu bildirilmiştir (Forte, 2013'ten aktaran Mahmoud, Dobosiewicz, Swayne, 2004). Programlama giriş derslerinin başarısızlık oranının yüksek olmasının sebebi, aynı anda problem çözme, analiz,

tasarım, programlama dilinin yapı ve noktalama işaretlerinin ve hata ayıklamasının uzmanlığı gibi pek çok beceriyi öğrenme zorunluluğudur (Gajraj ve diğerleri, 2010). Programlama öğrenmek; programlama kavramını, döngü ve koşul ifadeleri gibi temel programlama yapılarını öğrenmeyi, problem çözme ve hataları gidermeyi içeren karmaşık bir süreçtir (Du Boulay, 1989; Tüzün, 2007'den aktaran Coşar, 2013). Bunun haricinde yazılım geliştirmek için kullanılan yaygın programlama dillerinin yapılarında barındırdığı noktalama işaretleri ve tasarım ortamı da öğrencilerin programlamaya yönelik tutum ve algılarında negatif yönelimlerin oluşmasına sebep olduğu görülmektedir (Charters ve diğerleri, 2014). Öğrenciler tek gördükleri hata mesajları olduğunda, çıktı alamadıklarında ve hata mesajlarını yorumlayamadıklarında hayal kırıklığına uğramaktadırlar (Mahmoud, Dobosiewicz, Swayne, 2004).

Programlama karmaşık ve soyut bir yapıdadır. Eğitimciler geleneksel yöntemle ders işlerken bilgiyi tahtada verip birkaç örnekle açıklayarak anlaşılmasını beklemektedir (Garner, 2003'ten aktaran Yolcu, 2018). Öğrencilerin çoğu sorunun kendilerinde olduğuna, kendilerinin arkada kaldıklarına inanmakta ve sonunda vazgeçmektedirler (Mahmoud, Dobosiewicz, Swayne, 2004). Bu durum insanların programlamaya yönelik tutumlarının olumsuz etkilenmesine ve bireylerin programlamayı öğrenmelerine yönelik cesaretlerini kırmaktadır (Charters, Lee, Ko ve Loksa, 2014). Bazı öğrenciler dayanabildiği kadar dayanmakta bazıları ise ilk yarıyıldan sonra kolayca pes ettiği halde birçok öğrenci dilin kendisiyle ilgili bir sorun olmadığını, öğrenme sürecinde kullanılan araçların öğrenme sürecini zorlaştırdığını düşünmektedir (Mahmoud, Dobosiewicz, Swayne, 2004).

İş alanından eğlence sektörüne hayatımızın her alanına girmiş olan bilgisayarlar ve onları kontrol etmeye yarayan yazılımlarının önemi gün geçtikçe artmaktadır. Buna paralel olarak yazılımları geliştirmeyi bilen insan ihtiyacı da hızla artmaktadır (Karabak ve Güneş, 2013). İşyerinde programlama artık günümüzün genel kariyerleri arasında yerini alırken, Amerika da 90 milyona yakın insan excel ve web araçlarıyla, grafik dillerle, visual basic ve hatta java gibi profesyonel dillerle programlama yapmaktadırlar (Charters ve diğerleri, 2014). Günümüz şirketlerinin %70'inin bilgisayar yeterliliğini orta ve üst yönetim pozisyonları için gerekli gördüklerini ve bilgisayar okuryazarlığının iş alanları için öneminin çok arttığını belirtilmektedir (Chu, Huber, Mastel-Smith ve Cesario, 2009). Günümüzde kariyer değiştiren yetişkinlerin daha kazançlı işlere sahip olabilmek için programlama

bilmeleri gerekmektedir. Milyonlarca kişi için programlama teknik bir iş olduğu halde bazı işçiler aslında günlük iş aktivitelerinde farkında olmadan programlamayı kullanmaktadırlar (Kölling ve Utting 2013'den aktaran Charters, 2014).

Bilgisayar programlama işin bir parçası olarak günümüzde kullanılmaya devam ettiği halde bilgisayar bilimi ve teknoloji alanında kariyer yapma konusunda olumsuz tutum var olmaya devam etmektedir (Charters, 2014). Günümüz gençliği, son kullanıcı programlama yeteneklerini oyun tasarım ortamları kullanarak farkında olmadan kazansalar da programlamaya yönelik olumsuz tutum içinde olabilmektedir (Petre ve Blackwell, 2007'den aktaran Charters ve diğerleri, 2014). Bu sorunu gidermek için birçok yazılım geliştirilmiştir (Karabak ve Güneş, 2013). Bunu değiştirmek için eğlenceli ve basit bir ortam tasarımı yapma ihtiyacı ortaya çıkmıştır (Kert ve Uğraş, 2009). Programlamaya yönelik olumsuz tutumu azaltmak için araştırmacılar ve eğitimciler, ortaokul ve lise seviyesi öğrencileri hedef alan bilgisayar kampları ve eğitsel oyunları kullanmaktadır (Utting ve diğerleri, 2010). Programlamaya yönelik tutum, algı ve ilginin alanyazında görülen negatif etkilerden arındırılması için meslek seçimlerinin oluşmaya başladığı ortaokul seviyesinden itibaren çalışmalar yapılması gerekliliği ortaya çıkmıştır (Rodger ve diğerleri, 2009). Bu sorunların çözümü için eğitimciler ve aileler beraberce çalışmalı ve bir çözüm bulmalıdırlar. Bu çözümlerden biri öğrencilerin erken yaşta programlama eğitimi almalarının sağlanmasıdır (Alghamdi, 2017). Scratch, Alice, ve Greenfoot gibi çocukların programlama algılarını geliştirmeye yönelik eğlenceli programlama ortamları da bulunmaktadır (Charters ve diğerleri, 2014).

Programlama eğitimi daha basit ve daha az söz dizimi kuralı isteyen bir yapıyla yürütülebilseydi öğrenciler programın anlamsal yönlerine daha fazla yoğunlaşıp daha az hata yapabilirlerdi (Mannila ve diğerleri, 2006'dan aktaran Saeli ve diğerleri, 2011). Öğrencilerin algoritmik düşüncelerini sağlamak pek çok problemi çözerdi (Futschek, 2006'dan aktaran Saeli ve diğerleri, 2011). Algoritmik düşünmeyi öğretmek için bilgisayara bile ihtiyaç duyulmamaktadır (Curzon ve McOwan, 2008'den aktaran Saeli ve diğerleri, 2011). Gelişmiş ve gelişmekte olan ülkelerin bir kısmında programlama eğitimine verilen önem programlama eğitiminin ortaokul seviyesinden itibaren verilmesine sebep olmuştur (Karabak ve Güneş, 2013). Programlamaya yönelik olumsuz tutumu

azaltmak için arařtırmacılar ve eđitimciler, ortaokul ve lise seviyesi öđrencileri hedef alan bilgisayar kampları ve eđitsel oyunları kullanmaktadır (Utting ve diđerleri, 2010).

Günümüzde çocuklar dijital yerliler olarak dođmakta daha konuşmaya başlamadan tablet bilgisayar ve akıllı telefonları kullanmaya başlamaktadırlar (D. Al-Jumeily ve diđerleri, 2015'ten aktaran Alghamdi, 2017). Bu sebeple çocukların hayatı öğrenmelerinde ki deđişlikle beraber hem öğretmenler hem de çocuklar daha hazırlıklı olmalıdırlar (Alghamdi, 2017). Teknolojik gelişmeler sırasıyla önce bilgisayar teknolojilerini, sonra da programlama dillerini etkileyerek ortaya çıkan ihtiyaçları karşılayacak yeni programlama dillerinin gelişmesini sağlamıştır (Yolcu, 2018). Günümüzde programlama becerisi her bireyin sahip olması gereken niteliklerden biri haline gelmiştir (Yolcu, 2018). Bu durum programlama eđitiminin erken yař gruplarına kadar inmesi gerekliliđini ortaya koymuştur. Bu sebeple pek çok ülke programlama eđitimini müfredatlarına almaya başlamıştır (Alghamdi, 2017). Artan bu ihtiyacı karşılamak için üniversitelerde bölümler bulunmaktadır. Öğretmenler öğrencilerine programlama kavramlarını öğretebilmek için desteđe ihtiyaç duyacaklardır (Alghamdi, 2017). Bazı öğretmenler programlamanın çeřitli alanlarda küresel faydaları ve avantajlarını görmezden gelerek teknoloji řirketlerinde eđitim ve kariyer fırsatlarıyla ilgili olduđunu algılıyor. Bazı öğretmenler, Java veya C++ gibi programlama dilleriyle yalnızca biliřim deneyimlerinin sınırlı olduđuna inanmaktadır. Bu nedenle küçük sınıflarda biliřimle öğrencileri tanıştırmayı asla düşünemeyebilirler (Maya ve diđerleri, 2015'ten aktaran Sáez-López ve diđerleri, 2016).

Oyun geliştirme sürecinde problem çözme sürecine anında geri bildirim olduđundan bilgisayar oyunlarının tasarımı ve oynanması aşırı derecede motive edicidir, öğrenciler farklı yetenek ve stratejilerin geliştirirler (Collins ve diđerleri, 1996'dan aktaran Hong, Fadjo, Chang, Geist ve Black, 2010).

2.9. Biliřim teknolojileri ve yazılım dersi (BTY)

Türkiye'de ortaöđretim kurumlarında bilgisayar eđitimi, 1984 yılında kurumlara 1100 bilgisayarın alınmasıyla başlamış ve yıllar içerisinde güncellenerek devam etmektedir (Akgün ve Akgün, 2011'den aktaran Karabak ve Güneř, 2013). 2012-2013 eđitim öğretim

yılında deęişen eęitim sistemi ile birlikte 2007 yılından itibaren seçmeli olarak verilen Bilişim Teknolojileri dersi ortaokullarda 5. ve 6. Sınıflarda zorunlu, 7. ve 8. sınıflarda yine seçmeli ders olarak Bilişim Teknolojileri ve Yazılım (BTY) adı ile verilmektedir. Haftalık ders saati iki olan BTY dersinin, MEB tarafından belirlenen öğrenme alanlarının başlıkları 2012 yılında şu şekildedir (MEB, 2013'den aktaran Karabak ve Güneş, 2013).

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

2017 yılında müfredat tekrar deęiştirildi. Bu defa içerikte bazı deęişiklikler oldu. 5. sınıflarda ve 6. sınıflarda içerik ayrı ayrı yapılandırıldı. 5. ve 6. Sınıflarda ünite başlıkları 2017 müfredatına göre aşağıdaki gibidir (MEB, 2017) :

1. Bilişim teknolojileri
2. Etik ve güvenlik
3. İletişim, araştırma ve iş birliği
4. Ürün oluşturma
5. Problem çözme ve programlama

Her iki sınıf seviyesinde aynı ünite başlıkları olsa da içerikte farklılıklar mevcuttur. Çalışmanın kapsamı programlama olduğundan aşağıda programlama ile ilgili kısım sunulmuştur (MEB, 2017).

BT.6.5.1. Problem Çözme Kavramları ve Yaklaşımları

BT.6.5.1.2. Sabitleri ve deęişkenleri problem çözümünde kullanır.

BT.6.5.1.5. Problemin çözümü için bir algoritma geliştirir.

BT.6.5.1.7. Farklı algoritmaları inceleyerek en hızlı ve doğru çözümü seçer.

BT.6.5.1.10. Matematik ve bilgisayar bilimi arasındaki ilişkiyi tartışır.

BT.6.5.2. Programlama

BT.6.5.2.1. Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanır.

Scratch (scratch.eba.gov.tr), Kodadı2023, Code.org, Codecademy vb. programlama platformları kullanılabilir.

BT.6.5.2.5. Doğrusal mantık yapısını içeren programlar oluşturur.

BT.6.5.2.7. Karar yapısını içeren programlar oluşturur.

BT.6.5.2.9. Çoklu karar yapıları içeren programlar oluşturur.

BT.6.5.2.11. Döngü yapısını içeren programlar oluşturur.

BT.6.5.2.13. Bir algoritmayı uyarlamak için en uygun karar yapılarını seçer.

BT.6.5.2.14. Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.

BT.6.5.2.15. Tüm programlama yapılarını içeren özgün bir proje oluşturur.

2.10. İlgili araştırmalar

Bu bölümde programlamaya yönelik tutumla ilgili yapılmış olan yurtdışı ve yurtiçi araştırmalardan bazıları sunulmuştur.

Shasshani (1997), 202 yüksekokul öğrencisiyle yaptığı çalışmasında öğrencilerin bilgisayara yönelik tutumlarını cinsiyet, deneyim ve aile desteği açısından incelemiş ve kız ve erkek öğrencilerin tutumları arasında eğitim öncesinde var olan farkın eğitim sonucunda kalktığı gözlemlenmiştir.

Mahmoud, Dobosiewicz ve Swayne (2004) Java, JavaScript ve Html ile giriş seviye bir programlama kursu geliştirdikleri çalışmalarında, amaçları öğrencilerin tarayıcıda çalışan güçlü özelliklere sahip bir programlama dili kullanarak kendilerini daha üretken hissetmeli sağlamaktır. Çalışmalarının sonunda öğrencilerin Web tasarımı yoluyla programlama öğrenmelerinin sağlanmasının büyük oranda motive edici sonuçları olduğunu ifade etmişlerdir.

Kelleher ve arkadaşları (2007) çalışmalarında Alice'in hikaye desteği olan sürümü ile olmayan sürümünün kıyaslandığı çalışmalarında yaşları ortalama 12 olan 88 kız öğrenci ile çalışmışlardır. Bulgular, her iki Alice sürümünün de öğrencilerin programlama kavramlarının öğrenilmesinde, kullanım kolaylığında ve eğlencelilik konusunda da eşit derecede başarılı olduğunu, ancak hikaye desteği olan sürümünün motivasyon ve gelecekte kullanmaya devam etmeyi isteme konusunda daha başarılı olduğunu göstermektedir.

Hongwarittorn ve Krairit (2010), bilgi teknolojileri dersini alan üniversite öğrencileri ile yürüttüğü çalışmada, tutum ile sınav puanları arasında anlamlı bir korelasyonun olmadığı sonucuna ulaşmıştır (aktaran Özyurt ve Özyurt, 2015).

Başer (2013) çalışmasında öğrencilerin bilgisayar programlamaya karşı tutumlarını ölçen güvenli ve geçerli bir Türkçe araç geliştirmiştir. Bilgisayar Mühendisliği ve Bilgisayar ve Öğretim Teknolojileri Eğitimi Öğretmenliği öğrencilerinden Python programlama dili dersini almış toplam 220 öğrenciye uyguladığı ölçekte, öğrencilerin programlamaya karşı olumlu tutuma sahip olduklarını söylemenin zor olduğu sonucuna ulaşmıştır. Bulgular mühendislik öğrencilerinin öğretmenlik öğrencilerine, erkek öğrencilerin de kız öğrencilere göre tutumlarının daha yüksek olduğunu göstermiştir.

Gökalp ve Aydın (2013) çalışmasında meslek yüksekokulunun Bilgisayar Programcılığı ve Elektrik Programında okuyan 135 gönüllü öğrenciyle bilgisayar ve İnternet kullanımına yönelik tutumları ve bunları etkileyen faktörlerin analizini yapmıştır. Bulgular öğrencilerin İnternet ve bilgisayar kullanımına yönelik tutumlarının orta düzeyde olumlu olduğu ve bu tutumların bazı değişkenlerden etkilendiği belirlenmiştir.

Charters ve arkadaşları (2014) çalışmalarında 200 yetişkine programlama deneyimi kazandırmak için geliştirilmiş bir eğitsel oyunla, kişilerin bilgisayar programlamaya yönelik algılarını anlamaya ve iyileştirmeye yönelik çalışma yapmışlardır. Bulgular yetişkinlerin başlangıçta programlamayı sıkıcı, zor, öğrenilmesi imkansız şeklinde değerlendirdiklerini, uygulama sonucunda ise katılımcıların programlamaya yönelik tutumları olumlu yönde değiştiğini göstermiştir.

Özyurt ve Özyurt (2015) 'un çalışmasında meslek yüksekokulunda okuyan 325 bilgisayar programcılığı öğrencisiyle yürütülen çalışmada öğrencilerin programlamaya karşı tutumları, programlama özyeterlikleri ve bunlar arasındaki ilişki incelemiştir. Bulgular, öğrencilerin bilgisayar programlamaya karşı tutumunun olumlu, programlama öz-yeterliklerinin orta düzeyde olduğunu göstermekte, öğrencilerin tutumlarında ve programlama öz-yeterlik düzeylerinde cinsiyet, sınıf düzeyi ve öğrenim türü değişkenlerinde anlamlı fark olduğunu ve öğrenci tutumları ve öz-yeterlikleri arasında pozitif yönlü ve orta düzeyde bir ilişki olduğunu göstermiştir.

Good ve Howland (2015) “Herkes için programlama” bağlamında, uzun zamandır devam eden arařtırmalarının sonucunda gençler için oyun üretiminde kullanılacak doğal bir programlama dili tasarımını Neverwinter Nights 2 (NWN2) Electron araç setini kullanarak gerçekleřtirmişlerdir. Araç seti, uzmanlık becerisine sahip olmayan gençlerin bir oyun oluřturmaya hızlı ve kolay bir řekilde başlamalarını saęlamakta ve yaratılan oyunlar profesyonelce geliřtirilmiş 3D ticari oyunlara benzer řekilde görüldüğü için oldukça motive edici bir yapıya sahiptir (Good ve Howland 2015).

2015 sonbaharında Harvard Üniversitesi'nin yeni programlama kursu başlatılmıştır. Bu çalışmada temel programlama prensiplerini anlamada yardımcı olması için Scratch blokları kullanılırken temel de C programlama dili ele alınmıştır. Scratch, ilk 2 hafta boyunca kullanılmıştır (Malan, 2015).

Yiğit (2016) 42 üniversitesi öğrencisiyle yaptığı çalışmasında, görsel programlama ortamı olarak Blockly yazılımının programlamayı öğrenme ve programlamaya karşı tutuma etkisini incelemiştir. Bulgular deney gurubuna uygulanan Blockly yazılımının kontrol grubunda kullanılan Python yazılımına göre programlama başarısı ve programlamaya yönelik tutumu deęiřtirme konusunda daha etkili olduğunu göstermiştir.

Al-Tahat ve arkadaşları (2016) çalışmalarında Alice 3D programlama ortamının Arap Açık Üniversitesinin bilgisayar bölümünde öğrenim gören kız öğrencilerin tutum ve performansları üzerinde ki etkisini arařtırmışlardır. Yarı deneysel olarak 56 öğrenci içinde bulunan 21 kız öğrenciyle yürütülmüş olan çalışmanın bulguları, Alice yazılımının öğrencilerin performans ve tutumlarında olumlu deęiřime sebep olduğunu ve kız öğrencilerin performanslarının erkek öğrencilere göre daha yüksek olduğunu göstermiştir.

Sáez-López ve arkadaşları (2016) İspanya’da 5 farklı ilkokulda öğrenim göre 107 ilkokul öğrencisiyle Scratch yazılımı üzerinden görsel programlama dillerinin sınıf uygulamasında kullanımının deęerlendirmesini arařtırmışlardır. İki eğitim öğretim yılı boyunca devam eden uygulamalarda bilim dersleri ve sanat derslerine kodlama ve görsel programlama eklenmiştir. Bulgular, programlama kavramlarının öğrenilmesinde belirgin gelişmeyle beraber öğrencilerin geliřtirdikleri çalışmalarda motivasyon, eğlence, baęlılık ve heves gösterdiklerini göstermiştir.

Estonya da 35 saatlik olarak planlanmış olan kursun temel araçları Scratch, Visual Basic ve Python'dur. Konuları; uygulama üretmek için yöntem ve uygulamalar, algoritma ve modelleme prensipleri ve programlama temelleridir. Kursun amacı öğretmenlerin ve öğrencilerin motivasyonlarını yükseltmek ve öğrenenlerin ilgisi programlama alanında tutmaktır (Mironova ve diğerleri, 2017).

Alghamdi (2017) çalışmasında erken yaş programlama eğitiminde kullanılmak üzere öğrencilerin seviyesine uygun öğrenmesini sağlayan değerlendirme odaklı bir sistem geliştirmiştir. Geliştirilen sistemin etkililiğini değerlendirmek için iki ilkokulda iki deneysel çalışma yapmıştır. Bulgular öğrencilerin geliştirilen sistemin programlama öğrenmelerine yardım ettiğini ve eğlendiklerini göstermiştir.

Yolcu (2018), programlama eğitiminde robotik kullanımının öğrencilerin akademik başarısı, bilgi-işlemsel düşünme becerisi ve öğrenme transferi üzerindeki etkisini belirlemek ve kullanılan robotik setlere ilişkin öğrenci görüşlerini ortaya koymak için yaptığı çalışmasında ortaokul 6. sınıfta öğrenim gören 47 öğrenci ile 14 haftalık bir uygulama yapmıştır. Bulgular, deney grubunun akademik başarı puanları, öğrenme transferi puanları, bilgi-işlemsel düşünme becerisi puanlarının kontrol grubuna göre anlamlı düzeyde yüksek olduğunu göstermiştir. Ayrıca öğrencilerin tümünün robotik setlere ilişkin olumlu görüşlere sahip olduğu görülmüştür.

2.11. Alan yazın taramasının özeti

Bu bölümde programlama eğitimi ve önemi, programlamanın tanımı, dinamik programlama dilleri, görsel programlama dilleri ve sınırlılıkları, programlamaya yönelik tutumla ilgili çalışmalar incelenmiştir.

Ülkemizde ve dünyada programlama eğitimi çalışmaları ele alınmıştır. İncelenen araştırmalar sonucunda yüksek zihinsel ve bilişsel beceriler kazandırabilen bilgisayar programlamanın, yalnızca yazılım sektöründe çalışacaklar için değil, tüm öğrenciler için önemli olduğu dile getirilmiştir.

Görsel programlama dillerinin en yaygın olanlarından Scratch, Alice, Code.studio.org, App inventor ve Greenfoot yazılımlarının özellikleri ele alınmıştır. Kod ezberi gerektirmeyen,

söz dizimi hatalarına müsaade etmeyen, blok tabanlı programlama dillerinin öğrencilerin programlama kavramlarının öğretimi, animasyon ve benzeşim üretme konusunda etkililiği dile getirilmiştir.

Programlamaya yönelik tutumu etkileyen durumlar ülkemizde ve dünyada yürütülmüş olan çalışmalar üzerinden değerlendirilmiştir. Öğrencilerin özellikle programlama kavramlarını öğrenmekte zorlandıkları dile getirilmiştir. Öğrencilerin programlama eğitiminde zorlanmalarının en önemli sebebinin, geleneksel programlama dilleriyle verilen eğitimlerde öğrencilerin programlama kavramlarını, kodları ve söz dizimi ustalığını ayna zamanda öğrenmek zorunda kalmaları olduğu dile getirilmiştir.

Programlamaya yönelik tutumu olumlu yönde değiştirmek için üretilen görsel programlama dillerinin programlamaya yönelik tutumu çoğunlukla olumlu yönde değiştirdiği görülmektedir. Bunun yanı sıra öğrencilerin programlama ve oyun tasarımına devam etme konusunda istekli oldukları, motivasyonlarının arttığı, öğrencilerin eğlenmesini sağladığı da bildirilmiştir. Ancak görsel programlamala dillerinin, öğrencileri ileri seviye programlama dillerine hazırlayamadığı belirtilmektedir. Bu sorunu çözmek için ortaya atılan yollardan biri web tasarımı için kullanılan javascript gibi dinamik dillerdir. Bu dillerle yürütülen eğitimin öğrencileri büyük oranda motive ettiği bildirilmektedir.

JavaScript gibi dinamik dillerin son yıllarda yeniden popülerlik kazanmasıyla beraber JavaScript kütüphanesi olan JQuery web tasarımında popüler olarak kullanılmaya başlamış olan bir dildir.

Bu çalışma kapsamında araştırmacı tarafından geliştirilen JTBP yazılımı, görsel programlama dillerinde olan blok tabanlı programlama mantığıyla beraber JavaScript ve JQuery kodlarını beraber içermektedir. Bu yazılım literatürde dile getirilen ileri seviye programlama dillerine hazırlayamama sorununu, platform bağımlılığını ve blok tabanlı yazılımlarda ortaya konan sınırlılıkları ortadan kaldırmak ve öğrencinin arka planda çalışan kodları görebilmelerini sağlamak için tasarlanmıştır.

BÖLÜM III

YÖNTEM

Bu bölümde, araştırma deseni, katılımcı grubun yapısı, veri toplama araçları, deneysel işlemler ve verilerin analiz işlemleri açıklanmaktadır.

3.1. Araştırma Modeli

Bu araştırma, JQuery tabanlı blok programlama (JTBP) sistemini öğrencileri ileri seviye programlama dillerine hazırlamak amacıyla geliştirmek ve bu yazılımı 6. Sınıf öğrencilerinin programlamaya yönelik tutumlarını değiştirmede etkililiği ve öğrenci görüşleri üzerinden değerlendirmek üzere gerçekleştirilmiştir. Kocaeli-Başiskele Yuvacık Levent Kırcı Oya Başar Ortaokulu'nda öğrenim gören 26 öğrenciden görüşme formlarıyla nitel veriler toplanmıştır. Ürün ve araç tasarlama, geliştirme ve değerlendirme süreçlerine odaklanıldığı için bu çalışma Tip 1 Tasarım Geliştirme Araştırmasıdır (Büyüköztürk vd., 2016). Tasarım geliştirme araştırması analiz, tasarım, geliştirme ve uygulama süreçlerinin sürekli olarak tekrar ettiği, yapısında sistematikliği ve esnekliği bir arada barındıran, araştırmacılar ile uygulamada yer alan katılımcıların işbirliğini içeren bir araştırma türüdür (Wang ve Hannafin, 2005'den aktaran Özer, 2017). Tasarım geliştirme araştırması türü Tip 1 ve Tip 2 olarak ikiye ayrılmaktadır. Richey ve Klein (2014) tip 1 tasarım geliştirme araştırmasını ürün ve araç tasarlama, geliştirme ve değerlendirme süreçlerine odaklanan bir araştırma, tip 2 tasarım geliştirme araştırmasını bir modelin tasarlanarak geliştirilmesi, onaylanması ile alanında kullanılarak değerlendirmesi ve sürekli olarak geliştirilmeye devam edilmesi süreçlerine odaklanan bir araştırma olarak tanımlamaktadır (aktaran Özer, 2017).

Geliştirilen JTBP uygulamasının öğrencilerin programlama eğitiminde kullanılması toplam 7 hafta olarak planlanmıştır. Öğrenciler uygulama öncesi programlama ve matematik ilişkisi, algoritma ve programlama konusunda genel olarak bilgilendirilmiştir. Ancak programlama kavramları verilmemiştir. Öğrencilere programlama kavramları anlatılmadan

önce ön görüşme formu uygulanmıştır. Araştırmacı kendisi tarafından geliştirilen JTBP sistemiyle programlama kavramlarını 5 haftalık bir süreçte anlattıktan sonra son görüşme formuyla öğrencilerden yeniden veri toplamıştır. Ardından öğrencilerin öğrendiklerini uygulayarak kendi özgün çalışmalarını geliştirmeleri istenmiş ve öğrenci tasarımları tamamlandıktan sonra son görüşme formu tekrar uygulanmıştır. Toplanan veriler betimsel analiz yapılarak değerlendirilmiştir.

Araştırma süresince JTBP yazılımı geliştirilmiş, pilot uygulaması sırasında ortaya çıkan sorunlar belirlenmiş, yazılım yeniden elden geçirilmiş, uygulamaya geçildikten sonra ortaya çıkan sorunlar için yeni çözümler bulunmuştur. Bu süreç boyunca JTBP yazılımının programlamaya yönelik tutumu değiştirme konusunda etkililiği ve öğrencilerin JTBP yazılımına yönelik görüşleri araştırılmıştır.

Çalışmada yarı yapılandırılmış görüşme formlarıyla toplanan nitel veriler, programlamaya yönelik tutumun ve JQuery tabanlı blok programlama sistemine yönelik öğrenci görüşlerinin derinlemesine anlaşılması ve bağlamı değerlendirmek için kullanılmıştır.

3.2. JTBP Yazılımının Geliştirme Süreci

Öğretim tasarımı ve teknolojisi alanında çalışma yapacak araştırmalar için tasarım ve geliştirme araştırmaları önerilmektedir (Özyol, 2019). Bu çalışmanın Tip-1 tasarım ve geliştirme araştırması olması uygun görülmüştür. Çevrimiçi öğrenme ortamı tasarımı için öğretim tasarımı modellerinden ADDIE modeli temel alınmıştır. ADDIE Modeli, eğitimde Analiz, Tasarım, Geliştirme, Uygulama ve Değerlendirme süreçlerinden oluşan sistematik bir yaklaşım modelidir. Alanyazın incelendiğinde ADDIE modeli ile geliştirilen birçok internet uygulaması, mobil uygulama, oyun ve yazılımlar olduğu dikkat çekmiştir (Özyol, 2019).

JTBP yazılımının geliştirilme süreci aşağıda sunulmuştur. Yazılımın değerlendirme aşaması bulgular bölümünde ele alınmıştır.

3.2.1. Tasarım öncesi analiz

Alanyazın taramasında programlama eğitimi için kullanılan pek çok yöntem ve yazılıma rastlanmıştır. Bu bölümde kullanılan yöntemleri, araçları ve kısıtlılıkları özetleyip bu çalışmayı yapmamız konusunda bizi motive eden sebeplere odaklanılmıştır.

Programlama eğitiminin küresel yararlarını ortaya koyan çalışmalar arttıkça programlama eğitiminin müfredata eklenmesiyle ilgili adımlar atılmaya başlanmıştır (Sáez-López ve diğerleri, 2016). Bazı ülkeler müfredatlarına programlama eğitimini eklemeye başlamıştır. 2011 yılında Estonya (Mironova ve diğerleri, 2017), 2014 yılında Birleşik Krallık (Alghamdi, 2017), 2007 de Kuzey İrlanda (Gibson ve Bradley, 2017) ve son olarak 2012 yılında ülkemizde bu alanda adım atılmıştır.

Estonya da programlama eğitimi Scratch üzerinden yapılması ile ilgili çalışmalar yapıldığı bildirilmiştir (Mironova ve diğerleri, 2017). Ülkemizde 2017 yılına kadar kullanılacak yöntem konusunda öğretmenlere herhangi bir öneri sunulmamaktaydı. Ancak değişen müfredatla beraber Blockly, Code.org, Scratch, Kodadı2023, ve Codecademy programlama platformları müfredatta görülmektedir (MEB, 2017).

Giriş seviye programlama öğretimi için geliştirilmiş olan blok tabanlı, kod ezberlemeyi gerektirmeyen yazılımlara genel olarak görsel programlama dilleri denmektedir. Bu yazılımlar metin tabanlı programlama dilleri gibi sözdizimi ustalığı, kod ezberi gerektirmemektedir (de Kereki ve Manataki, 2016). Bu yazılımlardan Alice ve Scratch en popülerleri olarak öne çıkmaktadır (Silla, Przybysz ve Leal, 2016; de Kereki ve Manataki, 2016). Bunun haricinde Greenfoot, Processing (Silla, Przybysz ve Leal, 2016) ve AppInventor (de Kereki ve Manataki, 2016) kullanılmaktadır. Görsel programlama dilleri birçok sorunu çözerek ilkokulda kodlamaya izin verir. Bloklarla görsel programlama, yalnızca şekilleri nedeniyle mantıklı olan biçimlerde yerine getirilebilir, bu nedenle derleyiciden hata mesajları almak mümkün değildir. Bu, giriş seviye programlama dersi için mükemmel bir rahatlama ve öğrenciyi geleneksel metin tabanlı dillerin verdiği gönül yarasından kurtarmaktadır (Wilson ve Moffat, 2010'dan aktaran Sáez-López ve diğerleri, 2016). Bu programlama araçlarının temel özelliği görselleştirilmiş blok kod yapısı, ara yüzünün kolay olması, çevrimiçi paylaşım imkânlarının bulunmasıdır (Resnick ve diğerleri, 2009). Blok tabanlı uygulamaların çoğunlukla programlamaya yeni başlayan

öğrenciler açısından etkili olduğu sonucuna varılan pek çok çalışma bulunmaktadır (Kalelioğlu ve Gülbahar, 2014).

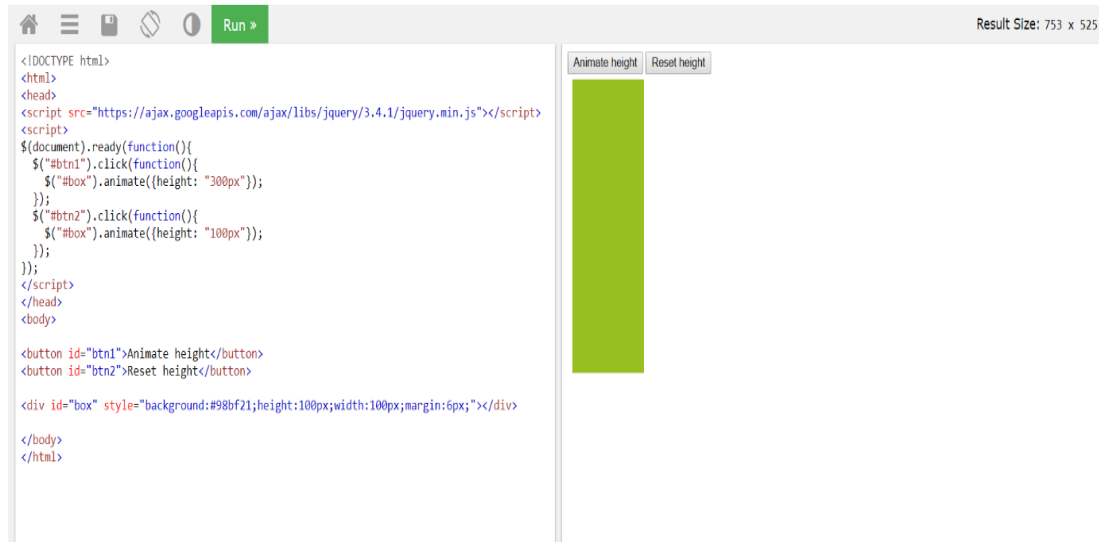
Görsel programlama dillerini kolay ve eğlenceli hale getiren özellikler kod ezberi gerektirmemesi, blok mantığı ve söz dizimi hatası gibi sorunların olmamasıyla beraber oyun, benzeşim ve animasyon tasarımı yoluyla öğrencilerin programlama kavramlarını öğrenmelerini sağlamasıdır. Alice eğitimi sonrasında öğrenciler C++ veya java'ya geçtiklerinde noktalama işaretleriyle uğraşmaktan dolayı kolaylıkla yılgınlığa düştüklerini ve bilgisayar bilimi için gerekli becerilere sahip olmadıklarını düşünmeye başladığını ifade etmişlerdir (Powers, Ecott ve Hirshfield, 2007). Du, Wimmer ve Rada (2016) çalışmalarında, code.org platformuyla yapılan bir eğitim sonrasında öğrencilerin tutumlarının olumlu yönde değiştiğini ancak programlama becerilerinde belirgin bir değişim gözlenmediğini bildirmişlerdir. Bunun sebebinin code.org üzerinden verilen eğitimden sonra öğrencilerin becerilerinde ki değişimi test etmek için uygulanan ölçme aracında bulunan soruların, geleneksel eğitimde kullanılan kodlar olduğu ve code.org üzerinden verilen eğitimin ölçme aracında ki sorulara öğrencileri hazırlayamadığı bildirilmiştir. Bununla beraber code.org sisteminin geleneksel araçların yerini almak için geliştirilmediği vurgulanmıştır (Du, Wimmer ve Rada, 2016). Bunun yanı sıra Scratch ve benzeri yazılımlar editör bağımlıdır ve ileri düzey özellikler eklemeyi desteklememektedir (Good ve Howland, 2015). Özyol (2019) çalışmasında kodlama geliştirme ortamı tasarlayacak ve geliştirecek olan geliştiricilerin ortamın amacına uygun olarak yapısal bir programlama dili etrafında uygulamayı geliştirmelerini ve arka planında çalışan programlama dili ile uygulamanın ara yüzünde sunulan araçların tutarlı olmasının tavsiye etmiştir. Ayrıca kullanılan blokların bir veya birden fazla programlama dilinde karşılıklarının sunulmasının erken yaş programlama dilleriyle ilgili zorluklar göz önüne alındığında faydalı olacağı bildirilmiştir (Özyol, 2019).

Araştırmacı yukarıda belirtilen çalışmalardan yola çıkarak görsel programlama dillerinde öğrencilere çekici gelen özellikler olan blok mantığı ve oyun tasarımı öğesini içeren ancak öğrencilere arka planda çalışan gerçek kodları da gösteren bir sistem tasarlamıştır. Bu geliştirilen sistemle araştırmacı, öğrencilere programlama kavramlarını oyun tasarımı yoluyla anlatırken öğrencilerin kodlarla tanışmasını sağlayarak öğrencileri ileri seviye programlama dillerine hazırlamak istemiştir.

3.2.2. Uygulama aracı tasarımı

Araştırmacı tarafından programlama eğitimi için tasarlanan JQuery Tabanlı Blok Programlama sistemi, öğrencilere Scratch görsel programlama dilinin oyun tabanlı programlama öğrenme özelliğini sunarken aynı zamanda öğrencileri kodlarla da tanıştırmak amacıyla tasarlanmıştır.

Bu sistem PHP, JQuery, Javascript, CSS ve Html kullanılarak tasarlanmıştır. Sistemin tasarımında sol tarafa yazılan kodun karşılığını çalıştır komutuyla sağ tarafta görebildiğimiz w3schools web sitesinden yararlanılmıştır. Bu web sitesinde kullanılan kodların örnek gösterimleri şekil 1 de sunulmuştur.



Şekil 1. w3scholls örnek ekranı

Geliştirilen JTBP yazılımında w3schools sitesinden farklı olarak kodları kod ekranına ekleyebilmek için fazladan bir de blok ekleme ekranı eklenmiştir. Bu haliyle geliştirilen yazılım Blockly ye benzemektedir. Ancak Blockly yazılımında kodlardaki değişim bloklara yansımamaktaydı. Uygulama ekranının sol bölümünde yer alan blok ekleme bölümünden öğrenciler blokları eklediklerinde bloklar algoritma adı verilmiş alana yerleşmektedir. Eklenen bloklar sürükleyerek sıraları ve yerleri algoritma sütünü içinde değiştirilebilmektedir. Blokların eklenmesiyle beraber ekranın sağında yer alan bölümde

onların JQuery kod karşılıkları çıkmaktadır. Bloklar üzerinde yapılan değişiklikler kodlara yansıdığı gibi kod bölümünde yapılan değişikliklerde bloklara yansımaktadır.

DÖNGÜ Ekle

Algoritma

Değişken Adı: puan = 0

oyun Sürekli yap

Puan:

insan ile hiyar1 dokununca

hiyar1 nesnesini gizle

Değişken değiştir: puan = puan+10

insan ile hiyar2 dokununca

hiyar2 nesnesini gizle

Değişken değiştir: puan = puan+10

insan ile hiyar3 dokununca

hiyar3 nesnesini gizle

Değişken değiştir: puan = puan+10

insan ile alev1 dokununca

insan nesnesini gizle

Değişken değiştir: puan = puan+10

insan ile alev2 dokununca

insan nesnesini gizle

Değişken değiştir: puan = puan+10

insan ile tup dokununca

alev2 nesnesini gizle

Değişken değiştir: puan = puan+10

insan ile alev3 dokununca

insan nesnesini gizle

Değişken değiştir: puan = puan+10

Eğer alev2 gizliyse

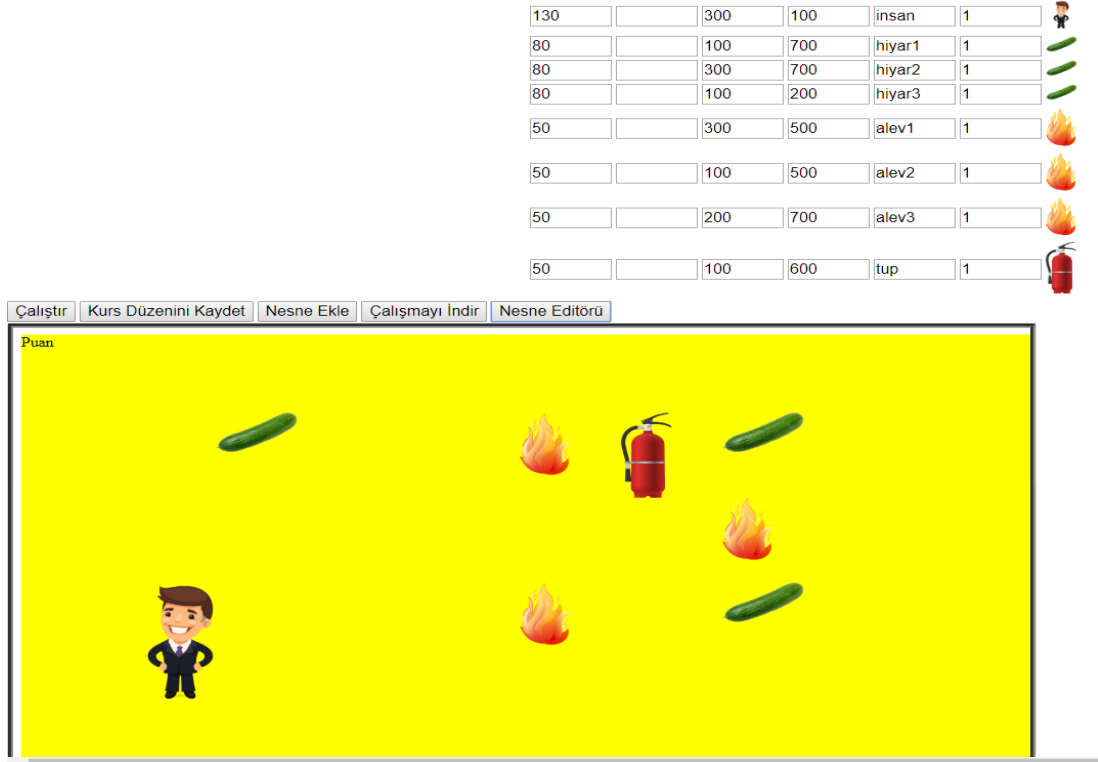
tup nesnesini gizle

```
var puan=0;
var oyun = setInterval(function(){
$("#puan").html("Puan:"+puan);
if (dokununca($("#insan"),$("#hiyar1"))){
$("#hiyar1").hide();
puan=puan+10;
}
if (dokununca($("#insan"),$("#hiyar2"))){
$("#hiyar2").hide();
puan=puan+10;
}
if (dokununca($("#insan"),$("#hiyar3"))){
$("#hiyar3").hide();
puan=puan+10;
}
if (dokununca($("#insan"),$("#alev1"))){
$("#insan").hide();
puan=puan+10;
}
if (dokununca($("#insan"),$("#alev2"))){
$("#insan").hide();
puan=puan+10;
}
if (dokununca($("#insan"),$("#tup"))){
$("#alev2").hide();
puan=puan+10;
}
if (dokununca($("#insan"),$("#alev3"))){
$("#insan").hide();
puan=puan+10;
}
if($("#alev2").is(':hidden')){
$("#tup").hide();
}
});

for (var i=0; i<1; i++){
$("#insan").animate({left: '+=200'});
$("#insan").animate({top: '+=100'});
$("#insan").animate({left: '+=300'});
$("#insan").animate({top: '-=100'});
$("#insan").animate({left: '-=100'});
}
```

Şekil 2. JTBP yazılımı arayüzü kod bölümü

Bu özellik sayesinde öğrencilere asıl işi yapan kodlar tanıtılırken aynı zamanda programlamadan daha fazlasını bekleyen, programlamayla ilgili öğrenciler kodları doğrudan kullanabilmektedir. Örneğin eklemek istediği blokları tek tek eklemek yerine onların karşılığı olan kodların bir kısmını ekledikten sonra kopyala-yapıştır yaparak gerektiği kadar çoğaltabilmektedir.



Şekil 3. JTBP yazılımı arayüzü ekran bölümü

Programlama işlemi bittikten sonra öğrenciler tasarımlarını paketleyip kendi bilgisayarlarına indirebilmektedirler.

Öğrencilere konu anlatımında JTBP yazılımı ile hazırlanmış olan senaryolar kullanılmıştır. Bu senaryolar 3 farklı ad altında toplanmışlardır. Bunlar aşağıda sunulmuştur.

1. Arılar
2. Balıklar
3. Yıldız Savaşları

Arılar senaryosunun altında 6 bölüm bulunmaktadır. Bu bölümler ile öğrencinin programı temel özellikleriyle beraber kullanmayı öğrenmesi ve döngü kavramını kullanmayı öğrenmesi amaçlanmıştır. Aynı ekran üzerinden verilen görevler aşağıda sunulmuştur.

1. Görev: Arıyı ilk çiçeğe ulaştırabilir misin?
2. Görev: Haydi şimdi de arıyı yolunun üzerinde ki ikinci çiçeğe ulaştıralım.
3. Görev: Şimdi de arıyı yolunun üstünde ki üçüncü çiçeğe ulaştıralım.

4. Görev: Arıyı üçüncü çiçeğe ulaştırmak için daha az blok kullanmak mümkündür. Bunu döngü komutu yardımıyla daha az blok kullanarak yapmayı deneyin. Döngü komutu içine konan blokları belirttiğiniz kadar tekrarlar.
5. Görev: Arıyı dördüncü ve son çiçeğe de ulaştıralım.
6. Görev: Arı yeterince çiçekten polen topladı. Onu kovanına ulaştıralım



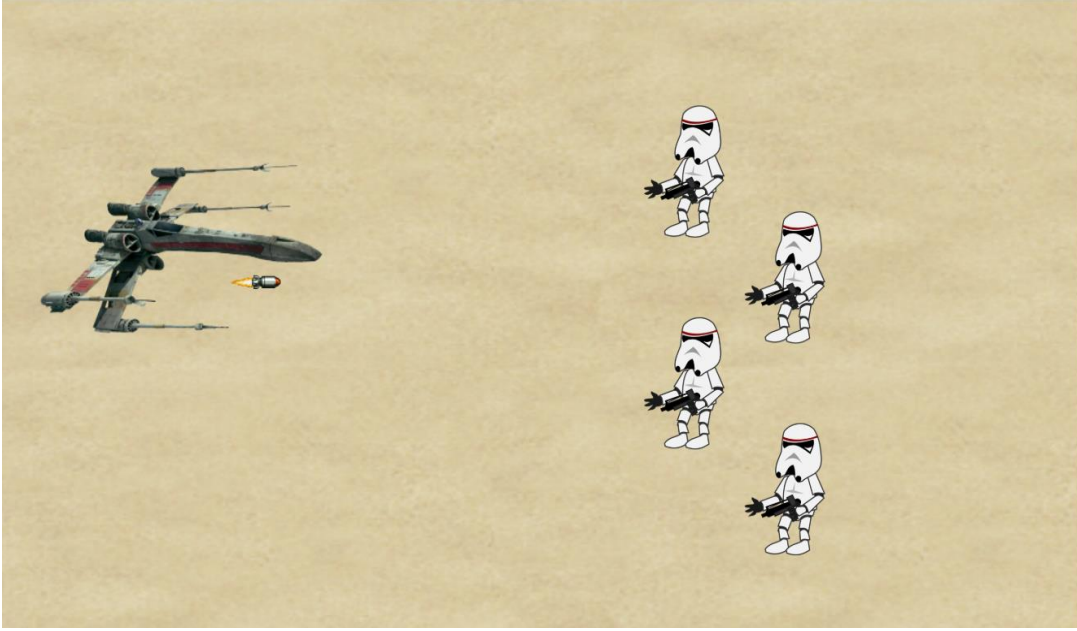
Şekil 4. Arılar senaryosu ekranı

Balıklar senaryosunun altında 4 bölüm bulunmaktadır. Bu bölümler ile öğrencinin programı temel özellikleriyle beraber kullanmayı öğrenmesi ve döngü kavramını kullanmayı öğrenmesi amaçlanmıştır.

Yıldız Savaşları senaryosunun altında 4 bölüm bulunmaktadır. Bu bölümler ile öğrencinin döngü ve şart kavramını kullanmayı öğrenmesi amaçlanmıştır. Farklı ekranlar üzerinden verilen görevler aşağıda sunulmuştur.

1. Görev: Ekrandaki tüm hurda metalleri toplamalısın. Dikkat et evlere dokunma.
2. Görev: Hurdaları toplarlarken askerlerden kaçınmayı unutma.
3. Görev: Haydi toplarladığımız hurdalarla gemiyi onaralım ve evimize dönelim. Ama dikkat et her defa da bir parça taşıyabilirsin. Bu sebeple daha az blok kullanmak için döngü kullan.

4. Görev: Şimdi de bu oyunda eksik olanı tamamlamalısın. Dokununca bloğunu Sürekli döngü bloğu içinde kullanarak bombanın düşmanlara çarptığı zaman düşmanların yok olmalarını sağlamalısın.



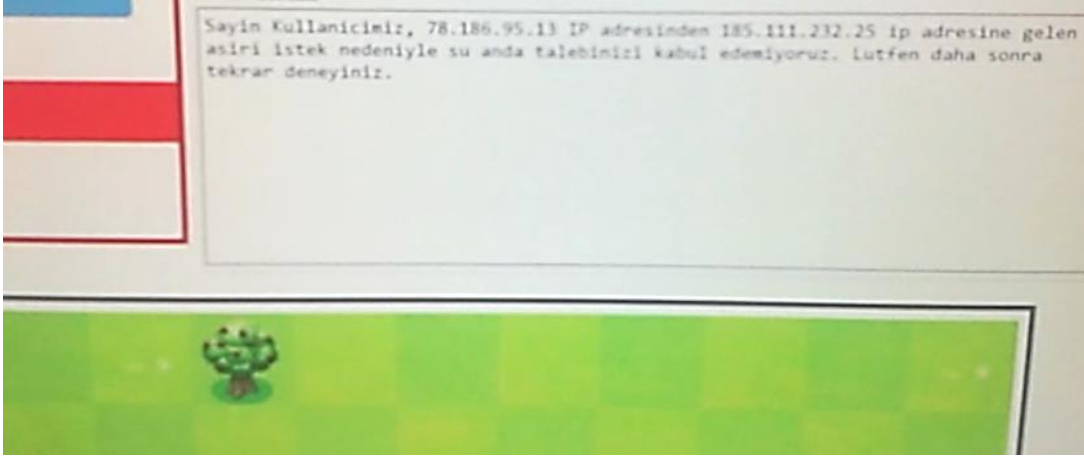
Şekil 5. Yıldız savaşları 4. senaryo ekranı

3.2.3. Pilot Uygulama

Geliştirilen uygulama tez amacıyla uygulanmaya başlanmadan önce devlet okullarından birinde çalışan bir bilgisayar öğretmeni tarafından ortaokul öğrencilerine kullanılarak pilot uygulaması yapılmıştır. Ortaya çıkan sorunlar ve eksiklikler tespit edilerek düzeltilmiştir. Uygulamada ortaya çıkan hosting firması kaynaklı sorunlardan dolayı hosting firması değiştirilmiştir. Çalışma esnasında birden fazla öğrenci sisteme girdiği zaman hosting saldırı var zannederek hata veriyordu. Bu sorundan kaynaklı hata ekranı şekil4'te sunulmuştur.

Ekran tasarımı için bazı iyileştirmeler yapılmıştır. Daha önce ihtiyaç duyulmamış olan bazı kod blokları gerekli olabileceği anlaşıldığından eklenmiştir. Uygulamaya geçildikten sonra öğrencilerin ekran tasarımı konusunda yaşadıkları sorunları azaltmak için araştırmacı nesne editörünü geliştirmiştir. Öğrenci JTBP sistemiyle tasarıma başladığında eklenen resimlerin, isimlerinin ve konumlarının tam listesini görememekte ve aynı resmi ekrana üst

üste birkaç defa ekleyip bunu fark edememekteydi. Bu durum nesne editörü ekranının tasarlanması ile beraber son bulmuştur.



Şekil 6. Pilot uygulamada karşılaşılan hosting hata ekranı

3.3. Katılımcılar

Katılımcılar Başiskele ilçesinde bir ortaokulda öğrenim gören 6. Sınıf ortaokul öğrencileri içinden seçilmiştir. Bu çalışmada amaçlı örneklem seçimi yöntemiyle örneklem seçimi yapılmıştır. Ortaokul 6. Sınıfta okuyan 3 sınıfın tamamına ön görüşme uygulanmıştır. İçlerinde en fazla geri dönüt alınabilen, çalışmaya katılım açısından en istekli olan, programlama konusunda ön öğrenmeleri en fazla olan sınıf çalışma için seçilmiştir. Araştırmanın örneklemini Kocaeli ili Başiskele ilçesi Yuvacık Levent Kırca Oya Başar Ortaokulunda öğrenim gören 16 erkek, 10 kız öğrenci olmak üzere toplam 26 altıncı sınıf öğrencisi oluşturmaktadır.

Tablo 2
Çalışma Örnekleme

	Kız	Erkek	Toplam
6.Sınıf	10	16	26

3.4. Veri Toplama Aracı

Araştırmada veriler katılımcılardan görüşme yöntemiyle toplanmıştır. Araştırmada derinlemesine bilgi toplamak için yarı yapılandırılmış görüşme formları kullanılmıştır. Alinyazında öğrencilerinin programlamaya yönelik tutumlarını ölçmek için kullanılmış olan ölçek maddeleri toplanıp incelendikten sonra amaca uygun cümleler temel alınarak ön görüşme ve son görüşme formları oluşturulmuştur. Sınıf seviyesine uygun olabilecek sorular ayıklandıktan sonra Sakarya Üniversitesi BOTE bölümü öğretim üyeleri ve görevlilerinin görüşleri alınmıştır. Ortaokul öğrencilerine uygun olmayacağı düşünülen maddeler uzman görüşüne başvurularak çıkartılmıştır.

6. sınıf öğrencilerinin programlamaya yönelik tutumlarını ve JTBP yazılımına yönelik görüşlerini incelemek için açık uçlu sorulardan oluşan yarı yapılandırılmış görüşme formu uygulanmıştır. Öğrencilere yöneltilecek sorular belirlenirken şu aşamalar takip edilmiştir:

1. Programlamaya yönelik tutumu ölçmek amacıyla kullanılmış makale ve tezler incelenip kullanılabilir sorular belirlenmiştir.

2. Sınıf seviyesine uygun olabilecek sorular ayıklandıktan sonra Sakarya Üniversitesi BOTE bölümü öğretim üyeleri ve görevlilerinin görüşleri alınmıştır.

3. Öğretim üyelerinin görüşleri doğrultusunda uygun sorular seçildikten sonra 8 soru öğrencilerin uygulama öncesi tutumlarını belirlemek amacıyla, 14 soru uygulama sonrası tutumlarını belirlemek amacıyla seçilmiştir.

Ön görüşme formunda öğrencilere yöneltilen sorular öğrencilerin programlamaya ve oyun tasarımına yönelik tutumlarını incelemek için, son görüşme formunda öğrencilere yöneltilen sorular ise öğrencilerin programlamaya yönelik tutumlarıyla beraber JTBP sistemine yönelik görüşlerini incelemek için hazırlanmıştır.

Ön Görüşme Formu

1. Daha önce programlama kursu aldınız mı?
2. Programlama öğrenmek ister misiniz? Neden?
3. Oyun tasarlamayı öğrenmek ister misiniz? Neden?
4. Sizce programlamayı herkes öğrenmeli midir? Neden?

5. Programlama size katkı sağlar mı? Açıklayınız.
6. Programlama sizce zor mudur?
7. Oyun tasarlamak zekanızı geliştirir mi?
8. Oyun tasarlamak yaratıcılığınızı geliştirir mi?

Son Görüşme Formu

1. Bu uygulamanın sizin için en güzel yanları nelerdir?
2. Bu uygulamanın sevmediğiniz yanları nelerdir?
3. Uygulamanın kullanımını sizce kolay mı?
4. Uygulama ile ilgili zorluk yaşadınız mı? Açıklayınız.
5. Bu uygulama sizi heyecanlandırdı mı? Hangi yönüyle heyecanlandırdı?
6. Bu uygulamayı eğlenceli buldunuz mu? Hangi yönlerini eğlenceli buldunuz?
7. Bu uygulamayı bir başka arkadaşınıza önerir misiniz?
8. Programlamayı sevdiniz mi? Neleri sevdiniz?
9. Programlama konusunda kendini geliştirmek ister misiniz? Neden?
10. Programlama konusunda bir şey öğrendiniz mi?
11. Programlama ile ilgili başka ne öğrenmek istersiniz?
12. Başka programlama dilleri de öğrenmek ister misiniz?
13. JQuery konusunda kendinizi geliştirmek ister misiniz?
14. Programlama ile ilgili ek görüşleriniz ve yorumlarınızı aşağıya ekleyiniz.

Yarı yapılandırılmış görüşme formu, yapılandırılmış görüşme formundan esnek olduğundan, görüşmenin akışına göre katılımcının yanıtları açmasını isteyerek derinlemesine veri toplanmasını ve ayrıntılara inmeyi sağlamaktadır. Eğer kişi görüşme esnasında belli soruların yanıtlarını başka soruların içerisinde yanıtlamış ise araştırmacı bu soruları sormayabilir. Yarı yapılandırılmış görüşme tekniği sahip olduğu esneklik nedeni ile eğitim bilim araştırmalarında daha uygun bir teknik görünümü vermektedir. Bu görüşme, nitel araştırma içerisinde görülebilir (Ekiz, 2003).

Yarı yapılandırılmış görüşme tekniğinin araştırmacıya sunduğu en önemli kolaylık görüşmenin önceden hazırlanmış görüşme protokolüne bağlı olarak sürdürülmesi nedeni ile daha sistematik ve karşılaşılabılır bilgi sunmasıdır (Yıldırım ve Şimşek, 2013). Bu

haliyle eğitim bilim çalışmalarına daha uygun bir araştırma biçimidir.

3.5. Verilerin Toplanması ve Çözümlemesi

Öğrencilerin çok azı daha önce programlama kursu almamışlardır. Öğrencilerin 4 tanesi ders dışı faaliyet olarak Kocaeli belediyesinin sunduğu bir hizmetten faydalanarak programlama konusunu c# dili üzerinden tanımıştır. Ancak bunlardan yalnızca bir tanesinin konu hakkında bilgiye sahip olduğu görülmüştür.

Öğrencilerin programlamaya yönelik ön tutumlarını belirlemek için yapılan görüşmeler, JTBP sistemi kullanılmaya başlamadan önceki iki haftalık süreçte gerçekleştirilmiştir. Görüşmeler gerçekleştirilmeden önce, katılımcılara verdikleri cevapların sadece bu araştırmada kullanılacağı ve görüşmenin amacı ile ilgili bilgiler verilmiştir.

Uygulama 7 hafta olarak planlanmıştır. JTBP sistemi ile konular 5 hafta anlatıldıktan sonra öğrencilerin tutumlarında değişikliği belirlemek amacıyla öğrencilere son görüşme formu uygulanmıştır. Son görüşme formunun uygulanmasından sonra öğrencilerin JTBP sistemi içinde kendi oyunlarını geliştirebilmeleri için oluşturulmuş bölüm yardımıyla kendi özgün oyunlarının geliştirilmesi sağlanmıştır. Öğrencilere verilen 2 haftalık süre bitiminde öğrencilere son görüşme formunda yöneltilen sorular tekrar uygulanarak öğrencilerin tutumlarında ki son değişim gözlenmiştir.

Nitel araştırmalarda veri analiz süreci üç temel aşama görülmektedir. Bunlar; betimleme, analiz ve yorumlamadır. Veri analizinin betimleme aşamasında, görüşülen bireylerin ne söyledikleri belirlenmeye çalışılır. Analiz aşamasında, verilerden elde edilen temalar ve veriler arasındaki ilişkiler kurulur. Son olarak süreçte, elde edilen bulguların araştırma bağlamı içerisinde yorumlanmasıyla tamamlanır (Yıldırım ve Şimşek, 2013).

Betimsel analizde, bireylerin görüşlerini çarpıcı bir şekilde yansıtmak için doğrudan alıntılara sık sık yer verilir. Bu tür analizde amaç, bulguların düzenlenmesi ve yorumlanarak okuyucuya sunmaktır (Yıldırım ve Şimşek, 2013).

Toplanan veriler kodlanmıştır. Oluşturulan kodlar, tutum kavramının alt başlıkları olan “Önem”, “İlgi”, “Motivasyon” başlıkları altında, olumsuz kodlar ise “Olumsuz tutum” başlığı altında toplanmıştır. Bunun haricinde “Öğrenme süreci” adında bir tema daha

oluřturulmuřtur. Bu temada 6ğrenme s6ireci sonunda elde edilenler ve s6ure7le beraber deęiřen tutumlar sunulmuřtur.

BÖLÜM IV

BULGULAR

Bu bölümde bulgular “Programlamaya yönelik tutum” ve “JTBP sistemine yönelik tutum” adında iki başlık altında sunulmuştur.

4.1. Programlamaya yönelik tutumla ilgili bulgular

Uygulanan görüşme formunda toplanan veriler “Programlamaya yönelik ilgi”, “Programlamayı faydalı ve önemli görme”, “Programlamaya yönelik motivasyon”, “Programlamaya yönelik olumsuz tutum” başlıkları altında sunulmuştur. Ayrıca programlama öğrenme süreciyle ilgili görüşleri ayrı bir başlık altında sunulmuştur.

4.1.1. Uygulama öncesi programlamaya yönelik tutum

4.1.1.1. Uygulama öncesi programlamaya yönelik ilgi

Tabloda öğrencilerin programlamaya yönelik ilgileriyle ilgili kodlar verilmiştir.

Tablo 3

Uygulama Öncesi Programlamaya Yönelik İlgi

Kod	f (n=26)
Bilgisayara ilgim var	6
Oyun tasarlamak isterim	6
Hoşuma gidiyor	2
Kolay	9

Öğrencilerin çoğu bilgisayara yönelik ilgilerine (f=6) bağlı olarak, programlamaya

yönelik ilgilerini ifade etmişlerdir. Öğrencilerden bazıları bilgisayara ilgileri olduğundan programlama öğrenmeye de isteklidirler. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

k02 “Evet öğrenmek isterim. Çünkü bilgisayar gibi şeyler derinine öğrenmek istiyorum”

k09 “Evet, çünkü ilgimi çekiyor”

k22 “Bilgisayarla ilgili bilgin yok öğrenmek için programlama öğrenmek istiyorum.”

k26 “Bilgisayarı daha iyi anlayabilmek için programlama öğrenmek isterim.”

Öğrencilerin bir kısmı da oyun tasarlamak (f=6) istediklerini ifade etmişlerdir. Oyun oynamanın eğlenceli olmasında ve oyun oynamayı sevdiklerinden dolayı oyun tasarlamayı öğrenmek istediklerini ifade etmişlerdir. Bir öğrenci de zeka geliştirici oyunlar tasarlamak istediğini ifade etmiştir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

k02 “Bilgisayarla aram olmasa da kendime oyun hazırlamak isterim”

k03 “Oyun tasarlamayı öğrenmek isterim oyun oynamayı severim”

k14 “Küçük oyunlar yaparak basit basit ilerleyerek daha ileriki zamanlarda zeka geliştirici oyunlar yapardım.”

k21 “Oyun oynamak eğlenceli olduğu için, oyun tasarlamayı öğrenmek isterim. Olsa iyi olur. Bence iyi olur.”

k23 “Oyun tasarlamayı öğrenmek isterdim.”

Öğrencilerden ikisi programlama öğrenmeyi farklı sebeplerle istediklerini belirtmişlerdir. Programlama öğrenmeyi istediğini belirten bu öğrencilerden bir tanesi sanal alemi sevdiğini ifade ederken diğeri eğlenceli olacağını düşündüğünü belirtmişlerdir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

k01 “Programlama öğrenmeyi isterim. Çünkü sanal alem hoşuma gidiyor.”

k15 “Programlama öğrenmeyi isterim. Hoşuma giderdi ve eğlenceli olurdu.”

Öğrenciler bir kısmı programlamanın kolay (f=9) olduğuna inanmaktadırlar. Bu görüşte olan öğrencilerden bir kaçı mantığını öğrenirsen programlama kolay diyerek düşüncelerini ifade etmişlerdir. Bunun yanında bir öğrenci algoritmanın kolay olduğunu ve birkaç yıldır bilgisayarla uğraştığını ifade etmiştir. Bir başka öğrenci ise istekli olan için

programlamanın kolay olduğunu ifade etmiştir. Öğrencilerin görüşlerinin bir kısmı aşağıdaki sunulmuştur.

k08 “Bence programlama zor değil. Kimine zor kimine değil. Öğrenmek isteyen kişiye zor gelmez bence zaten. Mesela başkası programlamayı sevmeyi için yazılım yapmak istemediği için çok ilerleyemez. Oyunu seven bir kişi çabaladığı için daha hızlı öğrenir. Hayır, isteyince her şey kolay”

k12 “Programlama da karışık kelimeler noktalama işaretleri var. Çok zor değildir. Programlamada mantığı bilersen zor değildir”

4.1.1.2. Uygulama öncesi programlamayı faydalı ve önemli görme

Öğrencilerin görüşleri incelendiğinde programlamayı faydalı ve önemli görme ile ilgili görüşleri aşağıdaki gibi özetlenmiştir.

Tablo 4
Uygulama Öncesi Programlamayı Faydalı Ve Önemli Görme

Kod	f (n=26)
Zekamızı geliştirir	31
Üretkenliğimizi geliştirir	24
Bana faydası olur	23
Gelecekte lazım olur	21
Bilişim çağındayız	7

Öğrencilerin çoğu oyun tasarlanmanın zekâyı geliştireceğini (f=31) ifade etmişlerdir. Oyun tasarlarken sürekli düşündüğümüzü, hayal gücümüzü ve beynimizi kullandığımızı bu yüzden zekâmızın gelişeceğini ifade etmişlerdir. Bunun yanı sıra bazı öğrenciler oyun tasarımını programlamayla yaptığımızdan, programlamanın da matematikle ilgili olduğundan dolayı zekâmızı geliştireceğini ifade etmişlerdir. Öğrenci görüşlerinden bazıları aşağıdaki sunulmuştur.

k02 “Hocam bu çağda herkes telefonunu eline almış sadece mesajlaşıyor. O yüzden kafalarını telefonlarından kaldırıp algoritma geliştirirsek zekamız gelişir. Oyun tasarlarken beynimizi kullanıyoruz bu da zekamızı geliştirir.”

k04 “Zekamızı geliştirir çünkü hayalimizi gerçekleştirir somutlaştırır. Kendi başımıza yeni bir dünya ortaya çıkartırız”

k06 “Bir defa yaparsak algoritmayı geliştirirsek daha iyi yapabiliriz ve zekamız artar.

k07 “Zekamızı geliştirir çünkü mantık gerekli”

k11 “Oyun tasarladıktan sonra bu bilgiyle başka şeyler de tasarlayabiliriz. Evet, oyun tasarlarken beynimizi kullanıyoruz bu da zekamızı geliştirir. Oyun tasarlarken mantığını anlıyoruz. Böylelikle başka bir oyun tasarlamak istediğimizde mantığını anladığımız için daha farklı ve güzel oyunlar tasarlayabiliriz.”

Öğrencilerin çoğu oyun tasarlamının üretkenliği arttıracığını (f=24) ifade etmişlerdir. Oyun tasarlama sürecinde sürekli düşündüğümüz için ve fikirlerimizi hayata geçirmek için uğraştığımız için üretkenliğimizin artacağını ifade etmişlerdir. Oyun geliştirirken zihnimizde hayalimizde yeni bir dünya ürettiğimizi ifade etmişlerdir. Bu süreçte hayal gücümüzün ve üretkenliğimizin de gelişeceğini ifade etmişlerdir. Öğrenci görüşlerinden bazıları aşağıda sunulmuştur.

k04 “Evet geliştirir çünkü kendi basımıza yeni bir dünya ortaya çıkartırız”

k08 “Mesela çizgi roman yazarken de hayal gücünü kullanıyorlar. Bunları hayalgücü olmayanlar yapamaz. Bu yüzden bence üretkenliği geliştirir.”

k11 “Oyun tasarlarken mantığını anlıyoruz. Böylelikle başka bir oyun tasarlamak istediğimizde mantığını anladığımız için daha farklı ve güzel oyunlar tasarlayabiliriz.”

k20 “Evet geliştirir ileride mimar olabiliriz park gibi şeyler tasarlayabiliriz”

Programlama öğrenmenin çeşitli şekillerde faydası olacağı (f=23), bu yüzden önemli olduğu dile getirilmiştir. Öğrenciler programlama öğrenmenin ve oyun geliştirmenin özellikle bilgisayar konusundaki bilgimizi artıracığını ifade etmişlerdir. Bize katkı sağlayacağını ve hayatımızı kolaylaştıracak bilgiler öğrenmemizi sağlayacağını ifade etmişlerdir. Öğrenciler öğrendikleri bilgilerin bilgisayarı daha iyi anlamalarını

sağlayabileceğini ve vakitlerini daha iyi değerlendirmelerini sağlayacağını ifade etmişlerdir. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k01 “Vaktimizi değerlendirmek için iyi bir yol olurdu. Hem boş vaktini değerlendiriyorsun hem de bir şey üretmiş oluyorsun. Oyun tasarlamayı öğrenmek isterim boş zamanımın çoğunu oyuna harcıyorum”

k05 “Programlama bence bize katkı sağlar. Çünkü matematik kullanılıyor.”

k07 “Programlama öğrenmeyi isterim çünkü bilgisayarı daha iyi öğrenmek için. Programlama katkı sağlar ben büyüdüğüm zaman daha iyi bilgi sahibi olmak için”

k14 “Programlama öğrenmeyi isterim çünkü programlama yaparak önemli bir şeyler öğrenebilirim”

Öğrencilerin bir kısmı programlamanın hayatlarında lazım olabileceğini (f=21) bu yüzden öğrenilmesi gerektiğini düşünmektedirler. Bazı öğrenciler programlamanın gelecekte lazım olacağına inanmaktadırlar. İş hayatında bilgisayarın günümüzde de her yerde olduğunu bu sebeple programlama öğrenmenin gelecekte daha rahat iş bulunmasını sağlayacağını ifade eden öğrencilerde bulunmaktadır. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k01 “Programlama kodları İngilizce olduğu için İngilizcemin gelişmesine de fayda sağlıyor. Yurtdışında da işe girebiliriz.”

k02 “Çoğu meslekte zaten bilgisayar olduğu için özellikle ben çok istiyorum”

k16 “Bence katkı sağlar. Yeni şeyler öğreniyoruz bunlar ileride katkı sağlar. Benim akrabam da anasınıfta öğretmeni. Bilişimle ilgili bir ödev verildi ve bilgisayarla arası olmadığı için yapamadı.”

Bazı öğrencilerde bilişim çağında olmamızdan (f=7) dolayı programlama öğrenmenin önemli olduğu ifade edilmiştir. Özellikle teknolojinin hızla gelişmesinden dolayı programlama öğrenmemiz gerektiğini ifade eden öğrencilerden bazıları okul çağına gelmemiş çocukların bile ellerinde akıllı telefonların olduğunu belirtmişlerdir. Bilişim araçları bizi bu denli çevrelediği bir çağda geri kalmamak için programlamayı öğrenmemiz gerektiğini ifade etmişlerdir. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k01 “Teknoloji gittikçe geliyor. Geride kalırsak ileri ki zamanlarda sorun yaşarız. Ben de bilgisayar mühendisi olmak istiyorum. Programlamayı bence herkes öğrenmeli. Teknoloji geliyor ve herkesin programlamayı bilmesi lazım.”

k02 “Programlamayı bence herkes öğrenmeli. Hayatımızın yarısında çoğu programlamayla ilgilidir. Çünkü teknoloji baya bir çağ atlıyor. Her geçen gün baya bir şey öğreniyoruz.”

k09 “Benim kardeşim youtube a girmeyi öğrendi. 1-2 yaşında kendine video açmayı öğrendi. Teknolojiye karşı kişiler istemeyebilir. Ama kültürün ilerlemesi lazım”

4.1.1.3. Uygulama öncesi programlamaya yönelik motivasyon

Öğrencilerin görüşleri incelendiğinde programlamayı yönelik motivasyonlarıyla ilgili görüşleri aşağıdaki gibi özetlenmiştir.

Tablo 5
Uygulama Öncesi Programlamaya Yönelik Motivasyon

Kod	f (n=26)
Dışsal motivasyon	12
İçsel motivasyon	11

Öğrencilerin bir kısmı içsel motivasyon (f=11) sebepleriyle programlama öğrenme isteği içinde olduklarını belirtmişlerdir. Bu öğrenciler çoğunlukla programlamanın başarılarını arttıracaklarını düşündükleri için programlama ve oyun geliştirmeye yönelik istekli olduklarını belirtmişlerdir. Bunun dışında kendi oyunları olmasını istedikleri için programlamaya yönelik isteklerini belirten öğrencilerde bulunmaktadır. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k01 “Vaktimizi değerlendirmek için iyi bir yol olurdu. Hem boş vaktini değerlendiriyorsun hem de bir şey üretmiş oluyorsun. Oyun tasarlamayı öğrenmek isterim boş zamanımın çoğunu oyuna harcıyorum”

k03 “Evet zekamızı geliştirir herkes bilgi sahibi olur daha iyi zeka oyunu yaparız”

k07 “Oyun tasarlamayı öğrenmek isterim çünkü daha iyi tasarlamak için”

k14 “Oyun tasarlamak zekâmızı geliştirir. Öğrenip bilgi sahibi oluruz ve ilerde başarılı oluruz”

Öğrencilerin bir kısmı dışsal motivasyon (f=12) sebepleriyle programlama öğrenme isteği içinde olduklarını belirtmişlerdir. Bu öğrenciler programlama ve oyun tasarımına yönelik isteklerini çoğunlukla başkalarıyla paylaşmak için şeklinde ifade etmişlerdir. Bu öğrenciler öğrendiği bilgileri başkalarıyla paylaşmak istediklerini belirtmişlerdir. Bu öğrencilerden bir tanesi youtube kanalı olduğunu ve gelecekte youtuber olmak istediğini, başkalarının sorduğu sorulara cevap verebilmek için programlama konusunda kendini geliştirmek istediğini belirtmiştir. Öğrencilerin bir kısmı küçük çocuklara faydalı oyunlar tasarlayabilmek için oyun tasarımını öğrenmek istediklerini ifade etmişlerdir. Bu öğrencilerden bir tanesi yerli oyun geliştirmek istediğini belirtirken bir diğeri insanları eğlendirmeyi sevdiği için oyun tasarımını öğrenmek istediğini belirtmiştir. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k02 “Küçük yaşta teknolojiyle tanışıyor çocuklar. Bende o çocuklar için eğlenceli zeka geliştirici oyunlar tasarlamak isterdim.”

k05 “Mesela hocam Türkiye kendini oyun alanında çok az geliştirdi. Sürekli yurtdışında tasarlanmış oyunları oynuyoruz. Kendi oyunlarımızı tasarlarız. Nasıl bir oyun oynamak istediğimi biliyorum.”

k09 “Biri internette veya blogta bir şey sorunca cevaplarız. Sosyal medyadan anlatırım hem ünlü olurum hem de insanlara faydam olur (youtube kanalı olan bir öğrenci). youtuber olmak istiyorum.”

4.1.1.4. Uygulama öncesi programlamaya yönelik olumsuz tutum

Öğrencilerin görüşleri incelendiğinde programlamayı yönelik olumsuz tutum görüşleri aşağıdaki gibi özetlenmiştir.

Tablo 6
Uygulama Öncesi Programlamaya Yönelik Olumsuz Tutum

Kod	f (n=26)
Zor	25
İlgim yok	16
Gereksiz	13
İsteyen öğrensin	13
Beceremem	4

Bazı öğrenciler programlamanın zor (f=25) olduğunu ifade etmişlerdir. Programlamayı zor olarak niteleyen öğrencilerin bir kısmı kodların İngilizce olmasının zor geldiğini ifade ederken, programlamanın karmaşık olduğunu ifade eden öğrenciler de vardır. Programlama zor diyen öğrencilerin bir kısmı mantığını oturtmanın zor olduğunu ifade etmişlerdir. Bilgisayara ilgisi olmadığı için programlamanın da zor ve karmaşık geldiğini ifade eden öğrencilerde bulunmaktadır. Öğrencilerden biri programlamanın pek çok süreci içeren zahmetli bir iş olduğunu ifade ederken bir diğeri yapmak istemeyene zor geleceğini ifade etmiştir. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k02 “evet programlama zor.”

k03 “Bence programlama zor konuyu anlamadıysak yapamayız. Mantığı oturtmak kurmak zor. Programlama orta zorluktadır”

k04 “programlama zor çünkü bilgisayarla aram iyi değil”

k06 “bilmiyorsak programlama zordur öğrenmediğimiz her şey zor gelebilir”

k09 “Biraz ama çalışana her şey kolay. İngilizce yoksa bir şey yapamazsın. İngilizce bilmezsek komutları anlayamayız. çok zor. Böyle basit bir tane yaparım. Böyle yıllarca sürecek veya aylarca sürecek bir oyun tasarlamak istemem.”

Bazı öğrenciler programlamaya ilgileri (f=16) olmadığını ifade etmişlerdir. Bunu ifade ederken öğrenciler çoğunlukla bilgisayara olan ilgisizliklerinden dolayı programlamaya da ilgi duymadıklarını ifade etmişlerdir. Bazı öğrenciler bilgisayarı anlamadıklarından dolayı programlamanın da karmaşık, sıkıcı ve saçma olacağını ifade etmişlerdir. Bir öğrenci oyun tasarlamayı insanları oyun bağımlısı yapmak istemediği için öğrenmek istemediğini ifade

etmiştir. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k02 “Bilgisayarla aram iyi değil”

k05 “ben bilgisayarın kendisini sevmiyorum. Ben teknolojinin kendisini sevmiyorum. Bana bunların hepsi sıkıcı geliyor.”

k10 “Anlamıyorum ve bilgisayarla aram yok. Oyun tasarlamayı öğrenmek istemem ilginç gelmiyor. Bilgisayarla ile alakam yok. Bana sıkıcı geliyor.”

Bazı öğrenciler programlamaya gereksiz gördüklerini (f=13) ifade etmişlerdir. Bu öğrencilerin bir kısmı programlama öğrenmenin bize katkısı olmayacağını ifade ederken, bir kısmı da sadece kullanmak gerektiği zaman lazım olacağını ifade etmişlerdir. Öğrencilerin bir kısmı da herhangi bir işe yaramayacağını bilgisayarı bile kullanmayacakken programlamanın ne yararı olmayacağını düşündüklerini ifade etmişlerdir. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k01 “Bilmem. Çok fazla katkısı olmaz”

k09 “Programlamanın katkısı olmaz. Yani sağlamaz çünkü sadece kullanırken lazım”

k17 “Programlamayı öğrenmek istemem. Çünkü gerekli görmüyorum.”

k19 “Bence sağlamaz çünkü benim bir işime yaramıyor”

Bazı öğrenciler programlamayı sadece isteyenlerin öğrenmesi (f=13) gerektiğini ifade etmişlerdir. Bu öğrenciler programlamayı herkesin öğrenmesi gerekmediğini ifade etmiştir. Bazı insanların öğrenmeye ihtiyacı olmayacağını ifade eden öğrenciler olduğu gibi ilgisi olmayanların sıkıcı veya saçma bulanların öğrenmesinin gereği olmadığını ifade eden öğrencilerde vardır. Bazı öğrenciler ise istekli olmayanların öğrenmesine gerek olmadığını ifade etmiştir. Öğrenci görüşlerinin bir kısmı aşağıda sunulmuştur.

k04 “İsteyenler öğrenmeli”

k05 “Bence herkes değil isteyen öğrenmelidir. Bana eğlenceli gelen diğerine sıkıcı gelebilir.”

k09 “Doğayla iç içe yaşamak isteyenler öğrenmese de olur. İsteyen öğrenmeli bence. Patron adam olsa çok fena zengin olsa tutar bir iki kişi onlara yaptırır işi. Kendisinin öğrenmesine gerek kalmaz. Bu yüzden herkes öğrenmemeli.”

k24 “Bence herkes değil isteyen öğrenmelidir. Bana eğlenceli gelen diğerine sıkıcı gelebilir. Öğrenmesi gerekmez çünkü bilişimle arası olmaya bilir.”

Bazı öğrenciler programlama ve oyun geliştirme konusunda kendilerine güvenleri olmadıklarını (f=4) ifade etmişlerdir. Bu öğrenciler yapacakları oyunun bir işe yaramayacağını ifade ederken diğerleri de beceremeyeceklerini ifade etmişlerdir. Öğrenci görüşleri aşağıda sunulmuştur.

k04 “Oyun tasarlamayı öğrenmek istemem bu konuda beceriksizim”

k09 “Benim yapacağım oyundan bir şey olmaz ve kötü olur”

k23 “Ben yapamam, beceremem de o yüzden de istemiyorum. Hem benim yaptığım kötü olur”

4.1.2. Uygulama sonrası programlamaya yönelik tutum

4.1.2.1. Uygulama sonrası programlamaya yönelik ilgi

Öğrencilerin programlamaya ilgileri ilgili görüşleri tabloda yer almaktadır.

Tablo 7
Uygulama Sonrası Programlamaya Yönelik İlgi

Kod	f (n=26)
Hoşuma gidiyor	15
Eğlenceli	5
Heyecanlı	2
Öğrenmek isterim	12
Öğrenince kolay	1

Öğrencilerin programlamaya yönelik görüşleri incelendiğinde programlamaya yönelik ilgileri eğlenceli (f=5), heyecanlı (f=2) ve hoşuma gidiyor (f=15) şeklinde özetlenebilir. Öğrencilerin bu görüşlerinin ayrıntılı incelemesi aşağıdadır.

Programlamanın hořlarına gittiđini ifade eden ğrenciler bu grřlerini řu řekilde ifade etmiřlerdir. ğrenciler programlamayı keyifli geldiđi ve oyun tasarladıkları iin sevdikleri ve hořlarına gittiđini ifade etmiřlerdir. Bunun yanı sıra iki ğrenci tasarım srecinde heyecanlandıklarını ifade etmiřlerdir. ğrencilerin oyunu sevdikleri iin oyun tasarlama srecini ve programlamayı da sevdikleri dřnlmektedir. ğrencilerin grřleri ařađıdaki sunulmuřtur.

K01 “Keyifli geliyor.”

K04 “ünkü emek verilmiř bir řey.”

K06 “Eđlenceli geldiđi iin programlamayı sevdim. Oyun tasarlamak hořuma gitti.”

K12 “Oyun tasarlamayı sevdim.”

K14 “Algoritma ve kodlamayı sevdim.”

K16 “Programlama ok gzel tasarlamak hořuma gidiyor.”

Programlamayı eđlenceli ve heyecanlı bulan ğrenciler bu grřlerini řu řekilde ifade etmiřlerdir. ğrenciler programlamayı oyun geliřtirebildikleri iin eđlenceli bulduklarını ifade etmiřlerdir. Bunun yanı sıra kodları deđiřtirip yenisini yapmak ve algoritma geliřtirmenin heyecanlı olduđunu belirten ğrenciler de vardır. ğrencilerin grřleri ařađıdaki sunulmuřtur.

K01 “Evet eđlenceli programlama olması. Heyecanlandırđı kodları deđiřtirip yenisini yapmak.”

K12 “Programlama eđlenceliydi. Algoritma tasarlarırken heyecanlandım.”

K15 “Evet algoritma tasarlamak eđlenceli.”

Bazı ğrencilerin grřleri incelendiđinde, ğrencilerin bir kısmı programlamayla bađlantılı grdkleri alanları ğrenmeye istekli oldukları (f=12) grlmřtr. Bu ğrenciler oyun tasarımını, canlandırma yapmayı, basit yazılımlar ğrenmeyi ve programlamayla ilgili her řeyi ğrenmeyi isteklidirler. ğrencilerin grřleri ařađıdaki sunulmuřtur.

K04 “ünkü ben derste daha iyi olurum. Kendimi geliřtirirken yaratıcılıđımı da geliřtiririm.”

K05 “Basit yazılımlar ğrenmek isterdim ama basit!”

K07 “Eğer okulda varsa gitmek isterim.”

Bir öğrenci programlamayı artık ezberlediği için kolay olduğunu ifade etmiştir. Uygulama öncesi öğrenci görüşleri ile kıyaslandığında yalnızca bir öğrenci programlamayı kolay gördüğünü ifade etmiştir. Öğrenci görüşü aşağıda sunulmuştur.

K09 “biraz kolay geldi çünkü arttık ezberledim.”

4.1.2.2. Uygulama sonrası programlamaya yönelik motivasyon

Öğrencilerin programlamaya yönelik motivasyonlarıyla ilgili görüşleri tabloda yer almaktadır.

Tablo 8
Uygulama Sonrası Programlamaya Yönelik Motivasyon

Kod	f (n=26)
İçsel motivasyon	9
Dışsal motivasyon	6

Programlama öğrenimi konusunda istekli olan öğrencilerin içsel sebeplerle veya dışsal sebeplerle bu isteklerini ifade etmişlerdir. Öğrencilerin yanıtları aşağıda ayrıntılı olarak ele alınmıştır.

Öğrencilerden dışsal sebeplerle (f=6) programlama konusunda kendini geliştirmeye istekli olanlar bu isteklerini başkalarına fayda sağlama ve gelecekte lazım olacağı düşüncesiyle açıklamışlardır. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K02 “çünkü sıkıcı ve zorlu olmasına rağmen bir şeylerle uğraşmak ve mesleğimi elime aldığımda lazım olacak.”

K07 “oyun tasarlamayı çocuklara öğretip onların da tasarlamasını sağlamak.”

Öğrencilerden içsel sebeplerle (f=9) programlama konusunda kendini geliştirmeye istekli olanlar bu isteklerini başarılı olma isteği, programlamanın eğlenceli olması, daha gelişmiş

çalışmalar yapma isteği ve programlamayı faydalı görme düşüncesiyle açıklamışlardır. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K01 “Daha profesyonel işler falan üretmek isterim. Programlama konusunda kendimi geliştirmek isterim. Sanal alem hoşuma gidiyor.”

K06 “Öğretici ve eğlenceli olduğu için programlama konusunda kendimi geliştirmek isterim.”

4.1.2.3. Uygulama sonrası programlamayı önemli ve faydalı görme

Öğrencilerin programlamayı önemli ve faydalı görme görmeleri ile ilgili görüşleri tabloda yer almaktadır.

Tablo 9
Uygulama Sonrası Programlamayı Önemli ve Faydalı Görme

Kod	f (n=26)
Gelecekte lazım olur	5
Bana faydası olur	7

Öğrencilerin bir kısmı programlamanın faydası (f=7) olacağı görüşündedir. Bazı öğrenciler programlamanın akli, hayal gücünü ve üretkenliğini geliştireceğini düşünmektedirler. Bunun haricinde bazı öğrenciler programlamanın gelecekleri için gerekli (f=5) olduğu görüşündedir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K02 “çünkü sıkıcı ve zorlu olmasına rağmen bir şeylerle uğraşmak ve mesleğimi elime aldığımda lazım olacak.”

K05 “Hayal gücümüz gelişir.”

K09 “Aklımızı geliştiriyor.”

K19 “Tasarım yapma gücümüz gelişir.”

K24 “çünkü programlamayla ilgili önüme bir şey çıkarsa yapmak için.”

4.1.2.4. Uygulama sonrası programlamaya yönelik olumsuz tutum

Öğrencilerin programlamaya yönelik olumsuz görüşleri tabloda yer almaktadır.

Tablo 10
Uygulama Sonrası Programlamaya Yönelik Olumsuz Tutum

Kod	f (n=26)
Zor	18
Anlamadım	8
Sevmedim	5
Sıkıcı	1

Programlama öğrenimine yönelik olumsuz tutum içinde olanlar bu düşüncelerini zor (f=18), anlamadım (f=8), sevmedim (f=5) ve sıkıcı (f=1) olarak ifade etmişlerdir. Öğrencilerin bir kısmı da programlama konusunda ek bir şey veya farklı bir dil öğrenmek istemedikleri ifade etmişlerdir. Öğrencilerden bir tanesi İngilizce kelimelerin çokluğundan şikayet etmiştir. Öğrencilerin görüşlerinin bir kısmı aşağıdaki sunulmuştur.

K02 “İngilizceye çok ihtiyaç duyuyoruz ve sıkıcı”

K06 “Halen hangi komutları kullanmam gerektiği konusunda takılıyorum”

K10 “Algoritma oluştururken halen zorlanıyorum”

K13 “Programlamada hiçbir şeyi sevmedim çünkü anlamadım. Evet, zorluk yaşadım. Oyun tasarlama kısmında. Ben bu konuyu anlamadım ama kendi oyunumuzu tasarlamak iyi bir şey. Hayır, kolay değil çünkü anlamadım.”

K24 “Bence hayır kolay değil. Zorluk yaşadım çok İngilizce kelime var.”

4.1.2.5. Uygulama sonrası programlama öğrenme süreci

Programlama öğrenme sürecinde öğrencilerin programlamaya yönelik görüşlerin değişimi bu bölümde sunulmuştur.

Tablo 11
Uygulama Sonrası Programlama Öğrenme Süreci

Kod	f (n=26)
Öğrenmek hoşuma gidiyor	10
Programlama öğrendim	11
Oyun tasarımını öğrendim	5
Diğer	4

Programlama öğrenim sürecince öğrenciler, programlamaya yönelik ilgilerinin öğrendikçe (f=10) arttıklarını ifade etmişlerdir. Öğrenciler algoritma, komutlar, tasarım ve programlamayı öğrendiklerini ifade etmişlerdir. Öğrencilerin çok azı programlama öğrenirken ofis yazılımlarını da kullanmayı öğrendikleri ifade etmişlerdir. Öğrencilerin görüşlerinin bir kısmı aşağıdaki sunulmuştur.

K06 “Öğrenmemiz ve eğlenmemiz en güzel yanları.”

K07 “Yeni bir şey öğrendiğim için heyecanlandım.”

K15 “Oyun ve ofis programı öğreniyoruz.”

K22 “Programlamayı öğrendim. Programlamanın nasıl yapıldığını öğrendim.”

K24 “Bilgisayarla ilgili fazladan bir şey öğrendiğin heyecanlandım.”

K25 “Sinema film gibi şeyleri yapmayı öğrendim.”

4.1.3. Tasarım sonrası programlamaya yönelik tutum

4.1.3.1. Tasarım sonrası programlamaya yönelik ilgi

Öğrencilerin özgün tasarımlarını çalışan oyunlar haline getirdikten sonra programlamaya yönelik ilgileri bu bölümde sunulmuştur.

Tablo 12
Tasarım Sonrası Programlamaya Yönelik İlgi

Kod	f (n=26)
Hoşuma gidiyor	29
Eğlenceli	5
Kolay	5
Bilişim konusunda kendimi geliştirmek istiyorum	14
Oyun tasarımı yapmak isterim	11

Öğrencilerin programlamaya yönelik ilgileri eğlenceli (f=5), hoşuma gidiyor (f=29), kolay (f=5), bilişim konusunda kendimi geliştirmek istiyorum (f=14) ve oyun tasarımı yapmak isterim (f=11) şeklinde özetlenebilir. Öğrenci görüşleri aşağıda ayrıntılı olarak ele alınmıştır.

Öğrenciler programlamayı eğlenceli bulduklarını ifade etmişlerdir. Öğrenciler programlamayı algoritma düzenleme eğlenceli (f=2), programlama eğlenceli (f=1), oyun geliştirme eğlenceli (f=1) ve eğlenceli (f=1) diyerek değerlendirmişlerdir. Öğrenciler programlamayı bir oyun geliştirme süreci olarak tanıdıklarından eğlenceli bulmuşlardır. Bir öğrenci programlamanın kolay olduğu görüşündedir. Öğrenci görüşleri aşağıda sunulmuştur.

K06 “Oyun geliştirmek Eğlenceli olduğu için JQuery konusunda kendimi geliştirmek isterim. Hem eğlenceli.”

K07 “Algoritmaları düzenlemek eğlenceli geliyor.”

K10 “Programlamayı eğlenceli buluyorum.”

K09 “evet. Büyüsem böyle bilgisayar konusunda ilerlesem geliştirmek isterim.

Programlama bana kolay geliyor.”

Öğrenciler oyun tasarım sürecine bağlı olarak programlamanın hoşlarına gittiğini grup çalışmasını sevdim (f=1), programlamayı seviyorum (f=5), bilgisayarı seviyorum (f=3), hoşuma gidiyor (f=8), oyun tasarımını sevdim (f=10) ve komutlarla çalışmayı sevdim (f=2) şeklinde ifade etmişlerdir. Öğrencilerin görüşleri aşağıda sunulmuştur.

K05 “Hazırlamayı ve çalıştırmayı sevdim. Programlamayı ve oyun tasarlamayı sevdim.”

K07 “Algoritma oluşturmayı ve nesne eklemeyi. oyun geliştirmeyi sevdim. Arkadaşımla beraber bunun üzerinde kafa yorarken vakit geçirmeyi. Programlama konusunda kendimi geliştirmek istiyorum. Nedeni yok istiyorum hoşuma gidiyor.”

K09 “Programlamayı ve bilgisayarı sevdiğim için programlama konusunda kendimi geliştirmek isterim. Küçüklüğümde beri uğraşıyorum bilgisayarla uğraşıyorum.”

K15 “Programlamayı sevdim. Yeni bir şeyler öğrenmeyi sevdim. Oyun tasarlamayı sevdim. Hayalgücümü gerçekleştirmeyi sevdim.”

K16 “Algoritma oluşturmayı ve nesne eklemeyi. oyun geliştirmeyi sevdim.”

K21 “Hayal gücümü gerçekleştirmeyi sevdim.”

K22 “Programlamayı sevdim. Kendimize oyun tasarlayabilmemizi ve hayal gücümüzü katmamızı sevdim.”

Bazı öğrenciler bilişim konusunda kendilerini geliştirmek istediklerini (f=14) belirtmişlerdir. Bu görüşte olanlar isteklerini profesyonel anlamda yazılım üretmek isterim, kolaysa başka programlama dilleri de öğrenmek isterim ve bilişim alanında kendimi geliştirmek isterim ve son olarak bir öğrenci bilişim öğretmeni olmak isterim şeklinde ifade etmişlerdir. Öğrenci görüşleri aşağıda sunulmuştur.

K07 “bir tanesini öğrendik ya farklı türlerini de öğrenip becerimi artırmak isterim. hocam kendimizi geliştiriyoruz.”

K09 “Bütün komutları ezberleyim editörsüz yazmayı isterdim.”

K09 “ekran tasarlamayı görüntüyü iyileştirme kaydedici gibi şeyleri yapmayı isterdim.”

K13 “böyle kolaysa öğrenmek isterim.”

K14 “Kendimi geliřtirmek isterim. Hedefim bt öğretmeni olmak. Sizin gibi web site kurup öğrencilerime güzel şeyler öğretirim. insanlara yardımcı olmak istiyorum.”

K21 “Biliřim alanında kendimi geliřtirmeyi çok isterim.”

Bazı öğrenciler geliřmiş ve daha gerçekçi oyun tasarlamak istediklerini (f=11) ifade etmektedirler. Bu öğrenciler isteklerini geliřmiş oyun tasarımlarını (f=10), milli oyunumuz olsun (f=1) şeklinde ifade etmişlerdir. Öğrenciler burada öğrendiklerini kullanıp üzerine yeni bilgiler öğrenerek kendilerini geliřtirip, daha gerçekçi ve faydalı oyunlar tasarlamak istediklerini belirtmişlerdir. Öğrenci görüşleri ařağıda sunulmuřtur.

K01 “Daha geliřmiş oyunlar tasarlasak daha güzel olur. řimdi yeni çıkan oyunlar gibi oyunlar tasarlasak ne olur. Hem tanınmış oluruz. Daha iyi oyunlar tasarlamak için.”

K15 “kendimi geliřtirip kendi oyunumu kendim yapmak istiyorum.”

K20 “Ülkemizde yerli oyun çok fazla yok ondan dolayı oyun geliřtirmeyi isterdim.”

K24 “Daha iyi oyunlar tasarlamak için. Daha geliřmiş oyunlar tasarlasak daha güzel olur. Daha iyi oyunlar tasarlamak için faydası olur.”

K25 “evet. Daha gerçekçi şeyleri yapmak isterim.”

Öğrencilerden bazıları programlamanın öğrenince kolaylařtığını ifade ederken çok azı programlamanın kolay olduğunu ifade etmektedirler. Öğrenci görüşleri ařağıda sunulmuřtur.

K04 “Sürekli döngüyü anladıktan sonra ana konuyu anladıktan sonra gerisi kolaylařıyor.”

K09 “Programlama bana kolay geliyor.”

K13 “Öğrendikten sonra kodlamalar da kolay geliyor güzel oluyor.”

4.1.3.2. Tasarım sonrası programlamaya yönelik motivasyon

Öğrencilerin özgün tasarımlarını çalışan oyunlar haline getirdikten sonra programlamaya yönelik motivasyonları bu bölümde sunulmuřtur.

Tablo 13
Tasarım Sonrası Programlamaya Yönelik Motivasyon

Kod	f (n=26)
Dışsal motivasyon	10
İçsel motivasyon	14

Öğrencilerin programlamaya yönelik motivasyonlarını dışsal motivasyon (f=10) ve içsel motivasyon (f=14) olarak özetleyebiliriz. Öğrenci görüşleri ayrıntılı olarak aşağıda işlenmiştir.

Öğrencilerden programlamaya yönelik motivasyonu dışsal etkilere bağlı olanlar görüşlerini, başkalarına faydalı olmak isterim (f=1), üstünlük sağlama isterim(f=1), hava atmak isterim (f=1), tanınmak isterim (f=4) ve başkalarıyla paylaşma isterim (f=3) şeklinde özetleyebiliriz.

K01 “Daha gelişmiş oyunlar tasarlasak daha güzel olur. Şimdi yeni çıkan oyunlar gibi oyunlar tasarlasak ne olur. Hem tanınmış oluruz.”

K09 “Çok fena heyecanlandırdı. Bazı kişiler yapamamıştı. Ben de kendim yaptığım için heyecanlandım. Kendimi üstün gördüm.”

K13 “Daha gelişmiş oyunlar tasarlasak daha güzel olur. Şimdi yeni çıkan oyunlar gibi oyunlar tasarlasak ne olur. Hem tanınmış oluruz.”

K14 “Kendimi geliştirmek isterim. Hedefim bilişim teknolojileri öğretmeni olmak. Sizin gibi web site kurup öğrencilerime güzel şeyler öğretirim. İnsanlara yardımcı olmak istiyorum. Öğrendiklerimi öğrencilerime ve diğer insanlara göstermek ve önermek isterim. Böylece site de büyür.”

K21 “Daha gelişmiş oyunlar tasarlasak daha güzel olur. Şimdi yeni çıkan oyunlar gibi oyunlar tasarlasak ne olur. Hem tanınmış oluruz.”

Öğrencilerden programlamaya yönelik motivasyonu içsel sebeplerle ilişkili olanların bu görüşleri, faydalı (f=2), oyun geliştirmek isterim (f=3), bilgisayarı seviyorum (f=3), bu konuda iyiyim (f=1), hoşuma gidiyor (f=3) ve kendini geliştirme isterim (f=2) şeklinde özetlenebilir. Öğrenci görüşleri aşağıda sunulmuştur.

K06 “. Oyun geliřtirmek Eđlenceli olduđu iin kendimi geliřtirmek isterim.”

K07 “Programlama konusunda kendimi geliřtirmek istiyorum. Nedeni yok istiyorum hořuma gidiyor. Geleceđime katkısı olacađını dűřünüyorum.”

K09 “Büyüsem böyle bilgisayar konusunda ilerlesem geliřtirmek isterim. Programlamayı ve bilgisayarı sevdiđim iin programlama konusunda kendimi geliřtirmek isterim. Küüklüđümden beri uğrařıyorum bilgisayarla uğrařıyorum. Programlama bana kolay geliyor.”

K20 “Ülkemizde yerli oyun ok fazla yok ondan dolayı oyun geliřtirmeyi isterdim. Bilgisayarla iie olayı seviyorum. Bilgisayarla i ie olmayı sevdiđim iin JQuery konusunda kendimi geliřtirmek isterim.”

4.1.3.3. Tasarım sonrası programlama önemli ve faydalı görme

Öđrencilerin özgün tasarımlarını yaptıktan sonra programlamayı önemli ve faydalı görme ile ilgili görüşleri bu bölümde sunulmuřtur.

Tablo 14
Tasarım Sonrası Programlama Önemli ve Faydalı Görme

Kod	f (n=26)
Bana faydası olur	25
Geleceđime faydası olur	16

Öđrencilerin programlamayı önemli ve faydalı gördüklerini ifade etmişlerdir. Öđrencilerin bu görüşleri, bana faydası olur (f=25) ve geleceđime faydası olur (f=16) şeklinde özetlenebilir. Öđrenci görüşleri ařađıda ayrıntılı olarak ele alınmıştır.

Programlamanın faydalı olduđunu dűřünen öđrenciler bu görüşlerini, bana faydası olur (f=7), faydalı (f=9), kendimi geliřtiririm (f=3), kodlama öğrenmiş olurum (f=1), zeka geliřtirir (f=1) ve üretkenliđimi geliřtirir (f=4) şeklinde ifade etmişlerdir. Bu görüşü benimseyen öđrenciler, bilgisayarlar hayatın her yerinde olduđundan programlama öğrenmenin de faydalı olduđunu dűřünmektedirler. Bir şekilde programlama öğrenmenin

kendilerine fayda sağlayacağını düşünmektedirler. Öğrenci görüşlerinden bazıları aşağıda sunulmuştur.

k01 “Kodlamayı öğreniyoruz daha ne olsun. zeka geliştirici yanları var”

k02 “Bilgisayar çağındayız. Çoğu kişi vaktini boşa harcıyor. Onun yerine bununla uğraşmalarını tavsiye ederdim. Oyunların arka planını görüyoruz. Nasıl yapıldığını öğreniyoruz. Daha faydalı şeyler yapmak istiyorum. Kendime faydası olacağını düşündüğüm için programlama konusunda kendimi geliştirmek isterim.”

k03 “Yaratıcılığımızı geliştirir. Daha faydalı şeyler yapmak istiyorum. Kendime faydası olacağını düşündüğüm için programlama konusunda kendimi geliştirmek isterim.”

Bazı öğrenciler programlama öğrenmenin geleceklerine fayda sağlayacağını (f=16) düşünmektedirler. Gelecekte bilişim alanında bir kariyer düşünmeseler bile bilgisayar programlamanın bir yerde lazım olabileceğini veya başka şekilde faydası olacağını düşünmektedirler.

k11 “Üniversiteye gidince lazım olur. Kendime faydası olacağını düşündüğüm için. Gelecekte bir şey tasarlamak istediğimiz zaman yararı olur.”

k16 “Gelecekte lazım olacağını düşünüyorum. Mesele teyzem öğretmenlik okuyor. Ama bilgisayar bilmediği için bir çok konuda sorun yaşıyor. Bizim de başımıza bu gelmemesi için öğrenmemiz gerekiyor”

k20 “Programlama konusunda kendimi geliştirmek isterim. ilerde bana katkı sağlayabilir. Bu işi meslek edinmesem de başka türlü katkı sağlayabilir. İnsanın zihnini geliştirmesi açısından güzeldi. Sayılarla iç içe.”

k21 “Kendime ilerde katkı sağlarım. İşimize çok yarar. Geleceğimize katkısı olacağını düşünüyorum daha iyi oyunlar tasarlamak için.”

4.1.3.4. Tasarım sonrası programlama yönelik olumsuz tutum

Öğrencilerin özgün tasarımlarını çalışan oyunlar haline getirdikten sonra programlamaya yönelik olumsuz tutumları bu bölümde sunulmuştur.

Tablo 15
Tasarım Sonrası Programlama Yönelik Olumsuz Tutum

Kod	f (n=26)
Zor	17
Sevmedim	3
Anlamadım	1

Öğrencilerin programlamaya yönelik olumsuz görüşleri anlamadım (f=1), sevmedim (f=3) ve zor (f=17) şeklinde özetlenebilir. Öğrencilerin görüşlerinin ayrıntılı incelemesi aşağıda yapılmıştır.

Programlamayı sevmedin diyen öğrenciler bunu algoritmayı sevmedim (f=2) ve bilgisayarı sevmiyorum (f=1) diyerek ifade etmişlerdir. Programlamayı zor olarak niteleyen öğrenciler programlama karışık (f=6), zor (f=5), komutlar zor (f=4), öğrenmekte zorlandım (f=2) diyerek ifade etmişlerdir. Öğrenci görüşleri aşağıda sunulmuştur.

K08 “En başta kelime işlemci de planı yaptığımız da komutları tam kavrayamamıştım. Sizin şablondan bakıp bakıp yaptım.”

K10 “Algoritma oluştururken halen zorlanıyorum. Bu kadarı yeterli. Çok karmaşık olduğu ve zorlandığım için istemem.”

K14 “Kullanımı döngü olarak biraz zor ama gene de kolay. Sürekli döngü kavramı zor. Kodları algılamakta zorlandım.”

K13, K21 “çok karmaşık geldi. Onu yapayım derken bunu bozdum. Hep karıştı. Başlarda çok karıştı. Ezber gerektiriyor.”

4.1.3.5. Tasarım sonrası programlama öğrenme süreci

Öğrencilerin özgün tasarımlarını çalışan oyunlar haline getirdikten sonra programlamaya yönelik görüşlerinin öğrenme sürecindeki değişimi bu bölümde sunulmuştur.

Tablo 16
Tasarım Sonrası Programlama Öğrenme Süreci Öğrenci Görüşleri

Kod	f (n=26)
Öğrenince ilgim arttı	5
Öğrenince yapabildim	7
Öğrenmeyi seviyorum	4

Öğrencilerin programlamayı öğrenme süreçlerinde oluşan değişiklikler bu bölümde işlenmiştir. Öğrenciler öğrenme süreci içinde programlamaya yönelik ilgilerinin değişimi, öğrenince eğlenceli (f=2), öğrenince sevdim (f=2) ve öğrenince heyecanlandım (f=1) şeklinde özetlenebilir. Bunun yanı sıra bazı öğrenciler, öğrenince yapabildim (f=7) diyerek öğrenme süreci sonunda başarılı olabildiklerini ifade ederken, bazı öğrenciler de öğrenmeyi seviyorum (f=4) şeklinde öğrenme sürecini sevdiklerini ifade etmişlerdir. Öğrenci görüşleri aşağıda sunulmuştur.

K01 “Bir şeyler yaptığımızı görüyoruz. Öğrendikten sonra kodlamalar da kolay geliyor ve eğlenceli hale geliyor.”

K07 “İlk yaptığımızda biraz zor geliyordu sevmemiştim. İlerledikçe yaptıkça çok iyi gelmeye başladı.”

K08 “Program tasarlarken farklı farklı şeyler öğreniyorum. Farklı farklı şeyler öğrenmek benim hoşuma gidiyor.”

K22 “Başta hiçbir şey anlamamıştım. Şimdi herşeyi anladım ve yapabilirim.”

Öğrencilerin kendi görüşleri üzerinden öğrenme durumları bu bölümde ele alınmıştır. Öğrenciler bilişimi öğrendim (f=1), kodlama öğrendim (f=3), hepsini öğrendim (f=4), oyun tasarımı öğrendim (f=15), programlama öğrendim (f=4) şeklinde bu eğitimin sonunda ne öğrendiklerini ifade etmişlerdir. Öğrenci görüşleri aşağıda sunulmuştur.

K01,02,03 “Algoritma, basit hareket komutlarını öğrendim. Ekran tasarlamayı öğrendim.”

K06 “Basit hareket komutları öğrendim. Algoritmayı öğrendim. oyun tasarlamayı öğrendim.”

K15 “Programlama konusunda ilk defa sizden ingilizce komutları öğrendim. Türkçe komutları biliyordum.”

K25 “Eskiden oyunları oynarken keşke bende tasarlayabilsem derdim. Şimdi tasarlayabiliyorum.”

4.1.4. Programlamaya yönelik tutum görüşlerinin karşılaştırması

Öğrencilerden toplanan görüşler üzerinden öğrencilerin programlamaya yönelik tutumları incelenmiştir. Bu görüşler üzerinden programlamaya yönelik tutumda değişimi görebilmek için benzer kategoriler, görüşlerin toplanma sırasına göre tema bazında sunulmuştur.

4.1.4.1. Programlamaya ilgi duyma

Öğrencilerin programlamaya yönelik ilgileri üç bölüm halinde incelenmiştir. Birinci bölümde hoşlanma, eğlenceli bulma gibi ifadeler incelenmiştir. Diğer bölümlerde istekli olma durumu incelenmiştir.

Tablo 17

Programlamaya İlgi Duyma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm I

Birinci Görüşme	f	İkinci Görüşme	f	Üçüncü Görüşme	f
Hoşuma gidiyor	2	Hoşuma gidiyor	15	Hoşuma gidiyor	35
Sevmiyorum	2	Sevmedim	5	Sevmedim	3

Programlama ve oyun tasarımı hoşuna gidenlerin sayısının belirgin şekilde giderek arttığı görülmektedir. Öğrenciler özellikle öğrendikçe ve başardıkça hoşlarına gitme durumlarının arttığı gözlenmektedir.

Tablo 18

Programlamaya İlgili Duyuma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm2

Birinci Görüşme	f	İkinci Görüşme	f	Üçüncü Görüşme	f
		Heyecanlı	2	Heyecanlı	1
		Eğlenceli	5	Eğlenceli	7
Sıkıcı	1	Sıkıcı	1		

Çocuklar oyun tasarımı sürecine girdiklerinde öğrencilerin programlamayı eğlenceli, heyecanlı buldukları ve programlamanın hoşlarına giden bir süreç olduğunu söylediklerini görmekteyiz. Fakat sıkıcı bulan kişi sayısı oldukça azdır. Bu durumun oluşmasında, programlama öğrenme sürecinin merkezinde oyun geliştirmenin bulunduğu söylenebilir.

Tablo 19

Programlamaya İlgili Duyuma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm3

Birinci Görüşme	f	İkinci Görüşme	f	Üçüncü Görüşme	f
Oyun tasarlamak	6	Oyun tasarlamak	3	Oyun tasarlamak	11
isterim		isterim		isterim	

Öğrencilerin bilgisayara ilgilerine bağlı olarak oyun tasarlama (f=6) ve programlama öğrenme (f=6) konusunda istekli olma durumlarının ikinci görüşmede değişimi, tabloya bakınca sabit gibi görünse de, oyun tasarlamayı ve programlamayı öğrendiğini düşünenlerin sayısı bu değeri geçmektedir.

Tablo 20

Programlamaya İlgili Duyuma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm4

Birinci Görüşme	f	İkinci Görüşme	f	Üçüncü Görüşme	f
Bilgisayara ilgim	6	Bilgisayara ilgim	9	Bilgisayara ilgim	14
var		var		var	
İlgim yok	13				

Son olarak ikinci görüşmede (f=9) bilgisayarla ilgili bilgi öğrenmek isteyen öğrenci sayısı, üçüncü görüşmede bilişim alanında kendini geliştirmek isteyenlerin (f=14) sayısından az olduğu görülmektedir. Öğrenciler çalışma öncesi programlamaya ilgileri olmadığını (f=13) düşünceleri sonraki görüşmelerde tamamen kaybolmuştur.

Öğrencilerin görüşleri incelendiğinde ilk görüşmede oyun tasarımına yönelik var olan ilginin artarak devam ettiği görülmüştür. Bunun yanı sıra bilgisayara ve programlamaya yönelik var olan ilgi de artmıştır. Ancak bilgisayara ve programlamaya ilgisi olmayanların sayısı azalarak sıfıra inmiştir.

Öğrencilerin programlamaya yönelik ilgiyle ilgili görüşmelerinin tamamı incelendiğinde, JTBP yazılımıyla oyun tasarımının programlamaya yönelik ilginin değişmesinde belirgin şekilde etkili olduğu söylenebilir.

Tablo 21

Programlamaya İlgi Duyma İle İlgili Öğrenci Görüşlerinin Karşılaştırılması Bölüm 5

Birinci Görüşme	f	İkinci Görüşme	f	Üçüncü Görüşme	f
Kolay	9	Kolay	1	Kolay	5
Zor	25	Zor	18	Zor	17
Beceremem	4				

Öğrencilerden toplanan veriler karşılaştırıldığında programlamayı kolay olarak niteleyen öğrenci görüşleri tasarım öncesi azalıp, tasarım sonrası tekrar arttığı gözlenmektedir. Bunun yanı sıra öğrencilerin programlamayı zor bulma durumlarında ciddi bir düşüş görülmüştür. Buna karşın programlama kolay görülmemektedir. Öğrencilerin çalışma öncesi programlamayı beremeyecekleri (f=4) düşüncesi sonraki görüşmelerde tamamen kaybolmuştur.

Programlamayı kolay görmeyle ilgili görüşler incelendiğinde, JTBP yazılımıyla oyun tasarımının öğrencilerin programlamayı kolay görmeyle ilgili görüşlerini değiştirmede etkili olduğu söylenebilir. Öğrenciler programlamayı tamamen kolay olarak değerlendirmeseler de, zor olarak değerlendirme durumlarının belirgin şekilde azaldığı, ayrıca beremeyeceklerini düşünenlerin ise tamamen kaybolduğunu görmekteyiz.

4.1.4.2. Programlamayı önemli ve faydalı görme

Öğrencilerin programlamayı önemli ve faydalı görmeleriyle ilgili görüşleri bu bölümde karşılaştırılmıştır.

Tablo 22

Programlama Önemli ve Faydalı Görme İle İlgili Öğrenci Görüşlerinin Karşılaştırılması

Birinci Görüşme	f	İkinci Görüşme	f	Üçüncü Görüşme	f
Gelecekte lazım olur	21	Gelecekte lazım olur	5	Gelecekte lazım olur	16
Bana faydası olur	23	Bana faydası olur	7	Bana faydası olur	25
Gereksiz	13				
Sadece isteyenler öğrensin	13				

Öğrencilerin görüşleri incelendiğinde programlamayı önemli ve faydalı görme ile ilgili öğrenci görüşlerinin ikinci görüşme de azaldığı görülmektedir. Buna karşın son görüşme de bana faydası olur, gelecekte lazım olur görüşlerinin yeniden arttığı görülmektedir. Öğrenciler programlama eğitiminin geleceklerinin bir parçası olduğunu bilmektedir. Ancak ikinci görüşmede önemli ve faydalı görme açısından azalma olması, öğrencilerin öğrendiklerini henüz uygulamaya koyamamaları ve öğrendikleri bilgiyle ne yapacakları konusunda tereddütte kalmalarından kaynaklanabilir.

Öğrenciler çalışma öncesi programlamaya gereksiz (f=13) bulduklarını ve programlamanın sadece isteyenler (f=13) tarafından öğrenilmesi gerektiği düşünceleri sonraki görüşmelerde tamamen kaybolmuştur.

Öğrenci görüşleri incelendiğinde JTBP yazılımıyla oyun tasarımının, öğrencilerin programlamayı önemli ve faydalı görmelerinde belirgin şekilde değişikliğe sebep olduğu söylenemeyebilir. Ancak programlamayı gereksiz görenlerin diğer görüşmelerde

görülmemesi, öğrencilerin programlamayı gereksiz görmemelerini sağladığı söylenebilir. Ayrıca programlamayı faydalı bulanların sayıları son görüşmede (f=25) da ilk görüşmeden bile fazla olduğu görülmüştür. Bu durumda programlamanın faydalı görülmesinde JTBP yazılımının az da olsa etkisi olduğu söylenebilir.

4.1.4.3. Programlamaya yönelik motivasyon

Öğrencilerin programlamaya yönelik motivasyonlarıyla ilgili görüşleri bu bölümde karşılaştırılmıştır.

Tablo 23

Programlamaya Yönelik Motivasyon Görüşlerinin Karşılaştırılması

Birinci Görüşme	f	İkinci Görüşme	f	Üçüncü Görüşme	f
Dışsal motivasyon	12	Dışsal Motivasyon	6	Dışsal motivasyon	10
İçsel motivasyon	11	İçsel Motivasyon	9	İçsel Motivasyon	14

Öğrencilerin programlamaya yönelik motivasyonlarında ikinci görüşme de bir düşüş yaşandığı görülmektedir. Bu durumun tasarım öncesi, öğrencilerin öğrendikleri bilgilerle ne yapacaklarını tam anlamamalarından dolayı olduğu düşünülmektedir. Bu durumun son görüşme de değiştiği öğrencilerin özellikle içsel motivasyonlarında olumlu yönde bir değişim olduğu söylenebilir.

Öğrenci görüşleri incelendiğinde programlamaya yönelik motivasyonun değişiminde yazılımla oyun tasarımının etkili olduğu söylenemeyebilir. Ancak içsel motivasyonun artmasında JTBP yazılımıyla oyun tasarımının az da olsa etkili olduğu söylenebilir. Bunda öğrencilerin programlama ve oyun tasarımının kendilerine faydası olacağı düşüncesinin etkili olduğu düşünülebilir.

4.2. JTBP sistemine yönelik tutumla ilgili bulgular

Uygulama sonrasında katılımcıların programlamaya yönelik tutumları ve geliştirilen sisteme yönelik tutumlarını ölçmek için 14 soruluk ikinci bir ölçek uygulanmıştır. Bu ölçeğin sorularının bir kısmı JTBP ye yönelik tutumla ilgiliyken diğer kısmı programlamaya yönelik tutumla ilgilidir. Ancak öğrencilerin sorulara verdikleri yanıtların bazıları doğrudan JTBP sistemine mi yoksa programlamaya yönelik mi olduğu tam ayırlamadığından doğrudan programlamaya yönelik olanları programlama ilgili kodlar doğrudan JTBP ile ilgili olan JTBP kodları olarak kabul edilmiştir.

4.2.1. Uygulama sonrası JTBP ye yönelik tutum

Ders anlatımı JTBP yazılımı üzerinden düzenlenmiş olan senaryolarla yapılmıştır. Öğrencilerin tutumlarının genellikle olumlu yönde olduğu görülmüştür. Öğrenciler senaryoları ve ders anlatımlarını bitirdikten sonra kendi özgün oyunlarını geliştirmeye başlamışlardır. Öğrencilerin cevapları incelendiğinde geliştirilen JTBP sistemine yönelik tutumları aşağıdaki gibidir.

4.2.1.1 Uygulama sonrası JTBP'ye yönelik olumlu tutum

Öğrencilerin cevapları incelenmiş ve aşağıdaki tabloda olduğu şekilde özetlenmiştir.

Tablo 24
Uygulama Sonrası JTBP'ye Yönelik Olumlu Tutum

Kod	f (n=26)
Eğlenceli	21
Kolay	16
Hoşuma gidiyor	14
JTBP'yi öneririm	16
Heyecanlı	5
Orta kolaylıkta	5

Öğrencilerin JTBP sistemine yönelik olumlu tutumları beş başlık altında toplandığı görülmüştür. Öğrencilerin sisteme yönelik görüşleri eğlenceli (f=21), heyecanlı (f=5), kolay (f=16), hoşuma gidiyor (f=14) ve orta kolaylıkta (f=5) şeklinde özetlenebilir. Bunun yanı sıra öğrencilerin çoğu bu uygulamayı başkalarına da önereceklerini (f=16) ifade etmişlerdir. Öğrencilerin JTBP sistemine yönelik olumlu görüşleri aşağıda ayrıntılı olarak ele alınmıştır.

Uygulamayı eğlenceli bulan öğrenciler bu görüşlerini eğlenceli (f=9), tasarım eğlenceli (f=8), oynatması eğlenceli (f=4) şeklinde ifade etmişlerdir. Bu düşüncelerini bazı öğrenciler oyunu tasarlamak süreç olarak eğlenceli diyerek ifade ederken diğerleri nesne eklemek eğlenceli veya çalıştırmak eğlenceli olarak ifade etmişlerdir. Bir öğrenci tasarım sürecinin çok eğlenceli ama aynı zamanda yorucu olduğunu ifade etmiştir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K02 “Çok eğlenceli, hem eğleniyoruz hem yoruluyoruz.”

K04 “Eğlenceli çünkü mantıklı.”

K06 “Oyunu oynatınca eğlenceli.”

K09 “Çok eğlenceli. Eh oynaması eğlenceli olabilir. Oynaması zevkli olabilir.”

K21,K22 “Bu çalışmayı arkadaşlarıma öneririm. Çok eğlenceli.”

Öğrenciler JTBP yazılımının hoşlarına gittiğini ve JTBP yazılımını güzel bulduklarını ifade etmişlerdir. Öğrencilerin bu görüşleri çalıştırmak hoşuma gidiyor (f=2), oyun tasarımı hoşuma gidiyor (f=6), hoşuma gidiyor (f=6) şeklinde özetlenebilir. Öğrenciler çoğunlukla bu görüşlerini oyunu çalıştırmak hoşuma gidiyor, tasarlamak hoşuma gidiyor şeklinde ifade etmişlerdir. Bazı öğrencilerde bu yazılımın sevmedikleri bir yanı olmadığını ifade etmişlerdir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K06,12,14,20,26 “Sevmediğim yanı yok.”

K13 “kendi oyunumuzu tasarlamak güzel.”

K18 “oyun tasarlamak güzel çünkü kendi oyunumu kendim yapıyorum. Oyun karakterlerini hareket ettirmek güzel.”

K21 “Evet heyecanlandım. Çünkü kendim tasarlamak hoşuma gidiyor.”

JTBP sistemini kolay bulduğunu belirten öğrenciler kolay (f=14), öğrendikçe kolaylaşıyor (f=1), nesne editörü kolaylaştırdı (f=1), orta zorlukta (f=5) diyerek görüşlerini belirtmişlerdir. Bir öğrenci uygulamayı kolay ama zahmetli bulduğunu ifade etmiştir. Uygulamayı kolay bulan öğrenciler genellikle zorluk yaşamadıklarını ifade etmişlerdir. Bir öğrenci başta zorlandığını ancak anlayınca kolaylaştığını ifade etmiştir. Öğrencilerin çok azı uygulamayı ne zor ne kolay bulduklarını ifade etmişlerdir. Uygulama zorluk olarak değerlendirdiklerinde orta zorluk veya kolaylıkta bulduklarını ifade etmişlerdir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K01,02,05,14,15,19,21 “zorluk yaşamadım.”

K04,14,10 “Kolay.”

K06 “son eklenen nesne editörü fotoları düzenlemeye daha kolay hale getirdi.”

Katılımcı-k10 “Başta azıcık zorlandım çünkü anlayamamıştım. Şimdi anladım kolaylaştı.”

Katılımcı-k06,12,21 “Orta zorlukta.”

Öğrencilerin bir kısmı uygulamayı heyecanlı bulduklarını ifade etmişlerdir. Öğrenciler kendilerine ait oyun tasarlama düşüncesinin onları heyecanlandırdıklarını ifade etmişlerdir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K05 “Biraz oyun çıkaracağım için heyecanlandım.”

K15,k19 “Oyunu tasarlarken heyecanlandım.”

K18 “Kendime ait bir programım olmasını isterim.”

4.2.1.2 Uygulama sonrası JTBP’ye yönelik olumsuz tutum

Öğrencilerin JTBP ye yönelik olumsuz görüşleri tabloda yer almaktadır.

Tablo 25
Uygulama Sonrası JTBP'ye Yönelik Olumsuz Tutum

Kod	f (n=26)
Zor	16
Sıkıcı	13
Sevmedim	5
Zahmetli	2
Uygulama sorunları var	1
Endişeliyim	1

Öğrencilerin JTBP sistemine yönelik görüşleri genellikle olumlu olsalar da cevapları incelendiğinde uygulamaya yönelik olumsuz görüşleri de bulunmaktadır. Uygulamaya yönelik olumsuz öğrenci görüşleri, oldukça zor (f=16), henüz öğrenemediğim için sıkıcı (f=13), sevmedim (f=5), zahmetli (f=2), endişeliyim (f=1) ve uygulama hataları var (f=1) şeklinde özetlenebilir. Öğrenci görüşleri aşağıda ayrıntılı olarak ele alınmıştır.

JTBP sistemini zor (f=16) bulan öğrenciler resimleri sisteme yüklemekte, konumlandırmakta, eklediği resimleri kodlarla ilişkilendirmede sorun yaşadıklarını belirtmişlerdir. Öğrenciler genellikle sistemi karmaşık, zor ve desteksiz kullanılamayacak şekilde tanımlamaktadır. Bir öğrenci konu belirlemede zorluk yaşadığını belirtmiştir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K01 “bazen kafamı karıştırıyor.”

K04 “kolay değil. Zorluk yaşadım. Kolayca nesne eklemeyi öğrenmek isterdim (zorlanıyor).”

K06 “çok az sorun yaşadım konu belirlemede ve fotoğrafları sisteme yükleyip ayarlamakta oldu.”

K13 “evet zorluk yaşadım. Oyun tasarlama kısmında. Ben bu konuyu anlamadım ama kendi oyunumuzu tasarlamak iyi bir şey. Hayır, kolay değil çünkü anlamadım.”

K22 “Bilgisayara ilgim olmadığından zorlanıyorum. Öğretmenimiz örnek vermeseydi yapamazdım. Çok zor.”

JTBP sistemini sıkıcı (f=13) bulan öğrenciler bu görüşlerini şu şekilde ifade etmişlerdir. Bir öğrenci sistemi rahatsızlık veren derecede sıkıcı diyerek tanımlamıştır. Sistemi sıkıcı bulan diğer öğrenciler görüşlerini genellikle eğlenceli değil, sıkıcı, heyecanlanmadım şeklinde ifade etmişlerdir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K13 “Kimseye önermem. Çünkü sıkıcı.”

K14,16,20,22,25 “hayır heyecanlanmadım.”

K16 “hayır heyecanlanmadım.”

JTBP sistemini sevmeyen (f=5) öğrenciler bu görüşlerini şu şekilde ifade etmişlerdir. Öğrencilerden biri yabancı kelimler içermesini sevmediğini ifade ederken bir öğrenci sevmediği için sistemin onu heyecanlandırmadığını ifade etmiştir. Bunun yanı sıra uygulamayı kullanmayı başaramadığı için uygulamayı sevmediğini ifade eden öğrenci da mevcuttur. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K02 “İngilizceye çok ihtiyaç duyuyoruz.”

K13 “Diğer hiçbir şeyini sevmedim.”

K26 “Evet. Yapamayacağım, öğretmen sevmeyecek diye korktum.”

JTBP sistemini zahmetli bulan öğrenciler bu görüşlerini şu şekilde ifade etmişlerdir. İki öğrenci kolay ama zahmetli diyerek görüşlerini ifade etmiştir. Öğrencilerin görüşleri aşağıdaki sunulmuştur.

K09 “kolay ama zahmetli bir şey.”

K25 “zor değil ama çok uğraşmak gerek.”

JTBP sisteminde uygulama hatası olduğunu söyleyen öğrenci bu görüşlerini şu şekilde ifade etmişlerdir.

K14 “bence TC no olmalı herkes başkalarının yaptıklarını silebilir.”

4.2.2. Tasarım sonrası JTBP'ye yönelik tutum

4.2.2.1. Tasarım sonrası JTBP'ye olumlu tutum

Öğrencilerin kendi özgün oyun tasarımları sonucu JTBP sistemine yönelik olumlu tutumlarını gösteren görüşleri bu bölümde sunulmuştur.

Tablo 26
Tasarım Sonrası JTBP'ye Olumlu Tutum

Kod	f (n=26)
Eğlenceli	29
Heyecanlı	17
Hoşuma gidiyor	19
Kolay	10
Orta kolaylıkta	2
JTBP'yi öneririm	20

Geliştirilen JTBP sistemine yönelik öğrenci görüşleri değerlendirildiğinde, öğrencilerin bu sisteme yönelik olumlu görüşleri, eğlenceli (f=29), heyecanlı (f=17), hoşuma gidiyor (f=19), kolay (f=10) ve orta kolaylıkta (f=2) şeklinde özetlenebilir. Bunun yanı sıra öğrencilerin çoğu bu uygulamayı başkalarına da önereceklerini (f=20) ifade etmişlerdir. Bu öğrencilerin kimi uygulamayı eğlenceli bulduğu için kimi de faydalı bulduğu için uygulamayı önereceğini ifade etmişlerdir. Öğrencilerin olumlu tutumlarının ayrıntılı değerlendirmesi aşağıda sunulmuştur.

Geliştirilen uygulamayı eğlenceli bulan öğrenciler bu görüşlerini, eğlenceli (f=8), oyun tasarımı eğlenceli (f=17), başarınca eğlenceli (f=2), oyun gibi olması (f=1) şeklinde ifade etmişlerdir. Öğrenciler oyun tasarımı sürecinde özellikle oyun geliştirmeyi ve oyun ortamının tasarımını eğlenceli bulmuşlardır. Oyun nesnelerinin resim olarak sisteme yüklenmesi süreci öğrenciler yönergeleri takip etmediğinde zor olsa da oyun çalışır hale gelince öğrenciler bu süreci eğlenceli bulmaya başlamışlardır. Öğrenci görüşleri aşağıda sunulmuştur.

K05 “çok eğlenceliydi (yüksek sesle gülererek). Tıkıyoruz balık hareket ediyor falan ya onlar çok hoşuma gitti. Oyun tasarım kısmı eğlenceliydi.”

K13 “hem eğleniyorum hem de bir şeyler geliştiriyorum. Bu sistemi başkalarına öneririm. Başta zorlanırdı ama oyun tasarlama tarafına geçince eğlenir.”

K21 “Oyun yapıyoruz ya hani oyunun çalıştığını görünce eğlenceli oluyor, mutlu oluyoruz. JTBP’yi öneririm. Başta zorlanırdı ama oyun tasarlama tarafına geçince eğlenir. Hem eğleniyorum hem de bir şeyler geliştiriyorum.”

K22 “Eğlenceliydi. Biraz zor ama güzeldi. Eğlenceli denebilir. Hocam çalışmam da bitti. Eğlenceliydi. Konumuzu kendimiz seçiyoruz ve karakterlerimizi istediğimiz gibi hareket ettiriyoruz. Güzeldi hocam.”

Öğrencilerin bir kısmı bu çalışmayı kullanarak oyun tasarım sürecini heyecanlı bulmuşlardır. Öğrenciler bu görüşlerini başarınca heyecanlandım (f=2), çalışınca heyecanlandım (f=7), heyecanlı (f=4), tasarım süreci heyecanlı (f=2), oyun yapacak olmak heyecanlı (f=2) şeklinde ifade etmişlerdir. Öğrenci görüşleri aşağıda sunulmuştur.

K09 “Çok fena heyecanlandırdı. Bazı kişiler yapamamıştı. Ben de kendim yaptığım için heyecanlandım.”

K10 “Çalıştır deyince hemen çalışması. Ana karakterin hepsini yemesi. İsteddiğimiz karakterleri eklemek. ekranı tasarlamak, çalıştır diyince sistemin çalışması. Acaba çalışacak mı çalışmayacak mı diye beklerken heyecanlandım.”

K11 “Acaba çalışacak mı çalışmayacak mı diye beklerken heyecanlandım. ekranı tasarlamak, çalıştır diyince sistemin çalışması.”

K13,21 “oyun yapacağınız demeniz heyecanlandırdı.”

K14 “Heyecanlandıran tarafları oldu. Mesela başarı elde edince heyecanlandım. Hocamın bana daha çok puanları vermesi ve daha iyi bir meslek edinme açısından heyecanlandırdı.”

Geliştirilen uygulama hoşuna giden öğrenciler bu konuda görüşlerini, hoşuma gidiyor (f=7), oyun tasarımı hoşuma gidiyor (f=5), başarınca mutlu oluyoruz (f=3), çalışması hoş (f=2), uygulama hoş (f=2) şeklinde ifade etmişlerdir. Özellikle oyun tasarlama süreci

öğrenciler tarafından sevilen bir süreç olduğu görülmüştür. Öğrenci görüşleri aşağıda sunulmuştur.

K08 “JTBP yi başkalarına da öneririm. Bu uygulama benim hoşuma gidiyor. Onun da hoşuna gideceğini düşündüğümde önerirdim. komutları eklemeyi falan sevdim.”

K13 “Oyun yapıyoruz ya onun çalıştığını görünce mutlu oluyoruz. Bir şeler yapabildiğimizi görüyoruz.”

K20 “JTBP nin sevmediğim yanı yok. Güzel bir uygulama. Karakterler hareket edince çok güzeldi. Karakterleri ve nesnelere ekleyebiliyoruz. Bu çok güzel. Kodları ekliyoruz. güzel oluyor. Bloklarla kod eklemek güzel oluyor.”

K25 “JTBP nin sevmediğim yanı yok. Hayalindeki şeyleri yapabiliyorsun. Oyun tasarlayabiliyorsun.”

K22 “Hiç baştan yapmaya kendim uğraşamam ama çok güzel bir uygulama. Arkadaşlarımın da yapmak isteyeceğini düşündüğüm için öneririm. Kendi oyunumuzu tasarlamak gerçekten çok güzeldi.”

Öğrencilerin bir kısmı sistemi kolay (f=10) olarak nitelemektedir. Bunun yanı sıra uygulamanın orta (f=2) zorlukta olduğunu belirten öğrencilerde bulunmaktadır. Öğrencilerin görüşleri aşağıda sunulmuştur.

K07 “Hiç zorlanmadım.”

K08 “Nesne Ekleme editörü işimizi çok kolaylaştırdı. Diğerlerine göre çok kolay hemen yapabiliyoruz. Kullanışlı buluyorum. Uygulama bence kolay.”

K14 “Kodlar hemen geliyor. Mesela yon hareketlerini kolayca yapabiliyoruz.”

4.2.2.2. Tasarım sonrası JTBP’ye yönelik olumsuz tutum

Öğrencilerin özgün tasarımlarını yaptıktan sonra JTBP sistemine yönelik olumsuz görüşleri bu bölümde sunulmuştur.

Tablo 27
Tasarım Sonrası JTBP'ye Yönelik Olumsuz Tutum

Kod	f (n=26)
Zor	17
Sıkıcı	5
Uygulama hataları	2
Sevmedim	2

Öğrencilerin JTBP sistemine yönelik olumsuz tutumları sevmedim (f=2), sıkıcı (f=6), uygulama hataları var (f=2) ve zor (f=17) olarak özetlenebilir. Bu görüşler aşağıda ayrıntılı olarak ele alınmıştır.

Uygulamayı zor bulan öğrenciler bu görüşlerini, ekran tasarımı zor (f=8), blokları sürükleme zor (f=4), karmaşık (f=2), komutlar zor (f=2) ve başaramadım (f=1) şeklinde ifade etmişlerdir. Öğrenciler özellikle ekran tasarımı yaparken zorlandıklarını ifade etmişlerdir. Öğrenciler nesnelere eklerken ve ekledikten sonra yapılacak işlerde, verilen yönergeleri takip etmediklerinden zorlanmışlardır. Bir öğrenci uygulamayı karmaşık olsa da kolay olarak değerlendirirken diğerleri karmaşık olduğu için zorlandıklarını ifade etmiştir. Öğrencilerin uygulama içinde nesne ekleme işlemi bittikten sonra algoritma tasarım kısmında nesnelere verdiği isimleri ayırt etmekte ve nesnelere konumlandırmakta çoğunlukla zorlandıkları görülmüştür. Bazı öğrenciler blokları sürüklerken zorlandıklarını ifade etmiştir. Sistem “google chrome” tarayıcısıyla uyumlu çalışacak şekilde tasarlanmıştır. Bazı öğrenciler uygulamayı “internet explorer” ile çalıştırınca sürüklemeye ilgili kodlarda sorunlarla karşılaşmıştır. Son olarak internet hızında ki yavaşlıktan dolayı zaman zaman donmalar yaşandığı görülmüştür. Öğrencilerin görüşleri aşağıda sunulmuştur.

K03 “Benim için çok zordu. Nesnelere yerini ayarlama hangi komutlarda hangi resmin olduğunu belirlemek zor oldu. resim eklediğimizde üst üste geliyordu. çok zorladım.”

K05 “Zorluklar vardı halada var. Mesela kediye tıklıyorum. Kedi değil de balığın hareket etmesi. Balık aşağı gitmiyor. Çok sinirim bozuldu.”

K16 “Nesne editöründen önce çok zorlandım. Resim eklemekte zorlanmadım. Komutlarda zorlandım.”

K20 “Başta kodların nasıl yerleşeceğini bilmiyordum. Nesnelerin konumunu ayarlarken nesne editörü olmadan zorluk yaşadım.”

K22 “Biraz zorlandım. Yardımsız biraz zor geliyor. Hareketlerini seçtiğimiz kısmı yardımsız zor geldi. Nesnelerin konumlarını ayarlarken sorun yaşadım.”

Uygulamanın kullanılması esnasında öğrencilerin karşılaştığı hatalar bulunmuştur. Öğrenciler bu görüşlerini uygulama hataları var (f=2) şeklinde ifade etmişlerdir. Öğrencilerden bazılarının çalışmaları sisteme öğrenci numarasıyla girilebildiğinden başka öğrenciler tarafından bozulduğu görülmüştür. Bu sorunun çözülmesi için öğrencilerin şifre olarak vatandaşlık numaralarını kullanması şeklinde bir önlem alınmıştır. Bunun yanı sıra internet hızına bağlı olarak veya öğrencilerin kullandıkları dosyaların büyüklüğüne bağlı olarak sorunlar ortaya çıkmıştır. Bu sorunlarda öğrencilerin gözünde uygulama hatası olarak görülmüştür. Öğrencilerin görüşleri aşağıda sunulmuştur.

K08 “bazı yerlerde tutukluk yapıyor. Onların giderilmesi daha hoş olur.”

K08 “Yanlış şeyler yapabiliyor. Bazen hatalar çıkıyor. Onlar benim sinirimi bozuyor. Uygulama hataları oluyor.”

Öğrencilerden çok azı sistemi kullanımını beğenmediğini (f=2) veya sıkıcı (f=5) bulunduğunu ifade etmişlerdir. Öğrenciler sistemi kullanamadığından sevmediklerini ve heyecanlı bulmadıklarını ifade etmişlerdir. Öğrencilerin görüşleri aşağıda sunulmuştur.

K05 “yaparken kazara sildiğim için çok sinirlerim bozuldu.”

K16 “yapamadığım için pek sevmedim.”

K05 “Heyecanlandırmadı.”

K08,14,15 “Beni heyecanlandırmıyor.”

K12,20 “Heyecanlı yanı yok.”

4.2.2.3. Tasarım sonrası JTBP ile ilgili öğrenci önerileri

Öğrencilerin özgün tasarımlarını yaptıktan sonra JTBP sistemine yönelik önerileri bu bölümde sunulmuştur.

Tablo 28
Tasarım Sonrası JTBP İle İlgili Öğrenci Önerileri

Kod	f (n=26)
Ayrıntılı komutlar	2
Birden fazla senaryo	1
Komutlar basit olsa	1
Şifre koruması	6

Öğrenciler sistemle ilgili bazı önerilerde bulunmuşlardır. Bazı öğrenciler ayrıntılı komutlar (f=2) olması gerektiğini ifade ederken, bir öğrenci birden fazla senaryo oluşturabilmeliyiz (f=1) demiştir. Bunun yanı sıra bir öğrenci de komutlar daha basit olsa (f=1) şeklinde görüşünü belirtmişlerdir. Bazı öğrenciler güvenlik açığı var (f=6) bunu düzeltmeliyiz demişlerdir. Öğrencilerin görüşleri aşağıda sunulmuştur.

K09 “ayrıntılı komutlar eklerdim. 3d komutlar da eklerdim. Çapraz gitmeyi eklerdim. Tuşlarla hareket ettirmeyi de isterdim.”

K14 “Mesela iki tane senaryo olan resim olur. Birinde savaş birinde mutlu sonlar bitecek senaryolar oluştururdum.”

K11 “komutlar daha basit olsa daha iyi olur.”

K09 “bazıları bizim hesabımıza girip çalışmalarımızı kurcaladı. Sistemin güven eksikliği var. Güvenlik açığı var.”

4.2.3. JTBP ye yönelik tutum görüşlerinin karşılaştırılması

Öğrencilerin JTBP sistemine yönelik görüşleri “olumlu” ve “olumsuz tutum” olmak üzere iki başlık altında incelenmiştir. Öğrencilerin geliştirilen JTBP sistemiyle ilgili ilgi görüşleri bu başlık altında kıyaslanmıştır.

Tablo 29

Öğrencilerin JTBP Sistemine Yönelik Tutumlarının Karşılaştırılması

Tasarım Öncesi Görüşler	f	Tasarım Sonrası Görüşler	f
Hoşuma gidiyor	14	Hoşuma gidiyor	19
Sevmedim	5	Sevmedim	2
Kolay	16	Kolay	10
Orta kolaylıkta	5	Orta kolaylıkta	2
Zor	16	Zor	17
Heyecanlı	5	Heyecanlı	17
Sıkıcı	13	Sıkıcı	5
Uygulama hataları	1	Uygulama hataları	2
JTBPyi öneririm	16	JTBPyi öneririm	20

Geliştirilen JTBP sistemine yönelik öğrenci görüşleri incelendiğinde tasarım öncesi öğrenci görüşleriyle, tasarım sonrası öğrenci görüşleri genel olarak birbirine çok yakın görülmektedir. Ayrıntılı incelendiğinde hoşuna gitme, eğlenceli ve heyecanlı bulma görüşlerinde artma gözlemlenirken, uygulamayı kolay görmeyle ilgili görüşlerde düşme görülmektedir. Bu durum öğrencilerin pek çok işlemi bir arada yapmak zorunda kalmalarından dolayı olduğu düşünülmektedir. Bu süreçte özellikle nesnelerin konumlandırılması nesne editörü öğrencilere sonradan sunulduğu için çok zorlayıcı olmuştur. Ancak nesne editöründen sonra da bazı öğrenciler rakamların ve kavramların ne ifade ettiği konusunda karmaşaya düşmüşlerdir. Bu durumun öğrencilerin JTBP sistemini kolay görme durumlarını düşürmüş olacağı düşünülmektedir. Buna rağmen öğrencilerin çoğu uygulamayı başkalarına önereceklerini ifade etmişlerdir.

JTBP uygulamasına yönelik olumsuz görüřlerin tasarım öncesi ve sonrası deęiřimi incelendięinde öęrencilerin uygulamayı sıkıcı bulma durumlarında ciddi bir düřüř görölürken zor bulma ile ilgili görüřleri deęiřmemiřtir. Öęrenci görüřleri genel olarak deęerlendirilirse öęrencilerin JTBP uygulamasına yönelik görüřleri öęrenci tasarımları sonrası olumlu yönde deęiřmiřtir.

BÖLÜM V

SONUÇ, TARTIŞMA VE ÖNERİLER

Bu araştırma, öğrencilerin programlamaya yönelik tutumlarını geliştirirken aynı zamanda onları ileri seviye programlama dillerine hazırlamak için geliştirilmiş olan, JQuery tabanlı blok programlama (JTBP) sistemini ve öğrencilerin programlamaya yönelik tutumlarını öğrenci görüşleri üzerinden değerlendirmek üzere gerçekleştirilmiştir. Öğrencilerin programlamaya yönelik tutumları üç aşamada alınmıştır. Bunlar uygulama öncesi, uygulama sonrası ve tasarım sonrasıdır. Ancak JTBP sistemine yönelik tutumları iki aşamada alınmıştır. Bunlar uygulama sonrası ve tasarım sonrasıdır.

5.1. Sonuç ve Tartışma

5.1.1. Programlama yönelik tutum

Bu çalışmada JTBP yazılımıyla oyun geliştirme süreci, öğrencilerin programlamaya yönelik eğlenceli bulma, heyecan verici bulma ve hoşuna gitme görüşlerinde olumlu değişime sebep olmuştur. Bunun yanı sıra programlamaya ve oyun tasarımına ilgilerini de arttırmıştır. Bu yönüyle çalışmanın bulguları aşağıda belirtilen çalışmaların çıktılarıyla uyumludur. Sáez-López ve arkadaşları (2016) çalışmalarında, Scratch yazılımının öğrencilerin programlama kavramlarını öğrenmelerinin yanı sıra çalışmalara eğlence, bağlılık ve coşku kattığını ifade etmişlerdir. Kelleher, Pausch ve Kiesler (2007) çalışmalarında Alice ile programlamanın kolay ve eğlenceli bulunduğunu ifade etmişlerdir.

Fesakis ve Serafeim (2009) endişeleri yüksek olan katılımcılarının olduğu bir grupta, Scratch yazılımı ile yaptığı çalışmalarının sonucunda, katılımcıların programlamaya devam etme konusunda istekli olduğunu bildirmiştir. Bu yönüyle çalışmanın çıktıları Fesakis ve Serafeim (2009) tarafından yapılan çalışmanın çıktılarıyla uyumludur.

Sáez-López ve arkadaşları (2016) çalışmalarında uygulanan yöntemleri öğrencilerin algıları açısından faydalı bulunduğunu ifade etmişlerdir. Bu çalışmada öğrenci görüşleri

üzerinden programlamanın gelecekte gerekli olacak bir beceri olduğu, ayrıca faydaları olduğu ifade edilmiştir. Bu yönüyle çalışma çıktıları uyumludur.

Feldgen ve Clúa (2004) çalışmalarında programlama öğreniminin öğrencinin kendi başına yapmak zorunda olduğu bir süreç olduğundan motivasyonun önemli olduğunu vurgulamaktadır. Motivasyonun öğrencilerin başarı ve derse devamını olumlu etkilediği belirtilmektedir. Kelleher, Pausch ve Kiesler (2007) çalışmalarında Alice yazılımı ile tasarım içeren çalışmalarında öğrencilerin daha fazla motive olduklarını bildirmişlerdir. Sáez-López ve arkadaşları (2006) çalışmalarında Scratch yazılımının öğrencilerin programlama kavramlarını öğrenmelerinin yanı sıra motivasyonlarını da arttırdığını ifade etmiştir. Mahmoud ve arkadaşları (2014) programlama giriş kursunda JS ve Html yapıları kullandıkları çalışmalarında, web tasarım çerçevesinin öğrenciler üzerinde oldukça motive edici etkiye sahip olduğunu bildirmişlerdir. Özyol (2019), 6-10 yaş arası saymayı öğrenmiş bireyler için yazılım geliştirdiği çalışmasında eğitimin başlangıç aşamalarında çocukların motivasyonlarının yüksek olmasına rağmen, ilerleyen haftalarda motivasyonlarının giderek düştüğünü ve bunu öğretilen araçların günlük hayatta kullanımlarına dair olan bağlamsal kopuklukların bu sonuca olabileceğini belirtmiştir. Bu çalışmada öğrencilerin dışsal motivasyonu, programlama eğitimi sonrası düşmüş, ancak öğrencilerin JTBP sistemiyle kendi özgün tasarımlarından sonra tekrar yükselmiştir. Öğrencilerin içsel motivasyonu konu anlatımı sonrası çok az düşmüş olsa da öğrenciler kendi oyunlarını tasarladıktan sonra belirgin şekilde artmıştır. Öğrencilerin motivasyonlarının oyun tasarımı sonrası artması yönüyle, çalışmanın çıktıları yukarıda belirtilen çalışmalarla uyumludur. Öğrencilerin görüşleri, oyun tasarımı işin içine girdiğinde, programlamaya yönelik motivasyonlarının arttığını, oyun tasarım sürecine girilene kadar geçen süreçte motivasyonun, oyun temelli bir ortamda bile azaldığını göstermektedir.

Literatür tarandığında programlamanın zor olarak nitelendiğini çoklukla görmekteyiz (Başer, 2013; Mahmoud ve diğerleri, 2004; Feldgen ve Clúa, 2004; Fesakis ve Serafeim, 2009). Bu çalışmada öğrencilerin programlamayı zor olarak görmelerinde belirgin bir düşüş gözlenmesine rağmen, öğrencilerin programlamayı kolay olarak görmelerinde konu anlatımı sonrası büyük bir düşüş ($f=1$), öğrenci tasarımları sonrası küçük bir artış ($f=5$) gözlenmiştir. Al-Tahat ve arkadaşları (2016) Alice ile verdikleri kursta, öğrencilerin programlamayı zor olarak görme durumlarında belirgin bir düşüş görürken, kolay olarak

görme durumlarında çok düşük bir artış gözlemlenmiştir. Bu çalışmanın çıktıları söz konusu çalışmanın çıktılarıyla benzerlik göstermektedir.

5.1.2. JTBP sistemine yönelik tutum

Oyun tasarımı sonrası kullanılan yöntem, yazılım veya geliştirilen sisteme yönelik görüşlerin olumlu yönde değişimi gözlemlendiği çalışmalar bulunmaktadır (Kelleher, Pausch ve Kiesler, 2007; Fesakis ve Serafeim, 2009; Al-Tahat ve diğerleri, 2016). Bu çalışma sonunda oyun tasarımıyla programlama öğretimi sonrası, olumlu görüşlerin artarken olumsuz görüşlerde azalma görülmüştür. Öğrencilerin JTBP uygulaması konusunda görüşleri genelde olumlu olsa da uygulamayı hala zor olarak değerlendirmektedirler.

Çalışmanın sonuçları, oyun tasarımıyla programlama eğitiminin öğrencilerde heyecana sebep olduğunu, öğrencilerin programlama kavramlarını öğrenmelerini kolaylaştırdığını, öğrencilere programlama sürecini sevdirdiğini, öğrencileri programlama ve oyun tasarımı konusunda isteklendirdiğini göstermektedir.

Oyun temelli tasarıma imkân veren JTBP sistemine yönelik öğrenci görüşlerinin, oyun tasarımı ile beraber olumlu yönde değiştiği görülmüştür.

5.2. Öneriler

Bu araştırma, öğrencilerin programlamaya yönelik tutumlarını geliştirirken aynı zamanda onları ileri seviye programlama dillerine hazırlamak için geliştirilmiş olan, JQuery tabanlı blok programlama (JTBP) sistemini ve öğrencilerin programlamaya yönelik tutumlarını öğrenci görüşleri üzerinden değerlendirmek üzere gerçekleştirilmiştir. Öğrenci tutumları, derinlemesine bir analiz sağlanabilmesi için nitel veri analizi kullanılarak incelenmiştir. Bunun sebebi araştırmacının öğrencilerin tutumlarını daha iyi anlayarak daha sonraki çalışmalarda programlama eğitimlerindeki kaliteyi ve etkililiği artırmak istemesidir.

Bulgular incelendiğinde JTBP yazılımıyla oyun geliştirme süreci, öğrencilerin programlamayı eğlenceli bulmalarını, heyecan verici bulmalarını ve sevmelerini sağlarken programlamaya ve oyun tasarımına ilgilerini de arttırmıştır. Öğrencilerin motivasyonunun

öğrendikleriyle ne yapacaklarını anlayana kadar düştüğünü ancak tasarımları yaptıktan sonra yükseldiği gözlenmiştir. Öğrencilerin programlamayı zor olarak görmelerinde belirgin bir düşüş gözlenmesine rağmen, öğrencilerin programlamayı kolay olarak görmelerinde yükselme görülmemiştir.

Çalışma sonunda oyun tasarımıyla programlama öğretimi sonrası, olumlu görüşlerin artarken olumsuz görüşlerde azalma görülmüştür. Öğrencilerin JTBP uygulaması konusunda görüşleri genelde olumlu olsa da uygulamayı hala zor olarak değerlendirmektedirler.

Bu sonuçların incelenmesinden sonra bu alanda araştırma yapmak isteyen araştırmacılar için şu konular önerilmektedir:

1. Öğrencilerin programlamayı kolay görmelerini sağlayacak bir tasarım yapılabilir veya bu tasarım üzerinde değişiklikler yapılarak görüşlerinde olumlu değişimlerin sağlanması üzerine bir başka tasarım çalışması yapılabilir.
2. Öğrencilerin programlamaya yönelik motivasyonun düşmesini engelleyecek değişimlerin sağlanması üzerine bir başka tasarım çalışması yapılabilir.
3. Öğrencilerin bu kursu sonrası programlama becerilerinde artma olup olmadığı araştırılabilir.
4. Öğrencilerin bu kurs sonrası diğer derslerinde ki başarılarında değişim incelenebilir.
5. Öğrencilerin bu kurs sonrası problem çözme becerilerinde iyileşme olup olmadığını içeren bir çalışma yapılabilir.
6. Ortaokul öğrencilerinin programlamaya yönelik tutumunu ölçmek için Türkçe ölçek çalışması yapılabilir.
7. JQuery dilinin Scratch ve benzeri görsel programlama dilleri ile kıyaslamasını içeren deneysel bir çalışma yapılabilir.
8. Bu çalışmada öğrencilerden sadece görüşme formlarıyla veri toplanmıştır. Bunun yanı sıra gözlem formları ve benzeri tekniklerle veriler zenginleştirilebilir.
9. Bu çalışma 6. Sınıf öğrencileriyle yapılmıştır. Bu çalışma diğer sınıf seviyelerinde ki öğrencilerle de yapılabilir.

KAYNAKLAR

- Akpınar, Y., ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13 (1). Erişim adresi: <http://ilkogretim-online.org.tr/index.php/io/article/view/2099>
- Alghamdi, M. Y. (2017). *Supporting the learning of computer programming in an early years education* (Doktora tezi, Liverpool John Moores University). Erişim adresi: <http://researchonline.ljmu.ac.uk/id/eprint/6390/>
- Al-Tahat, K., Taha, N., Hasan, B., ve Shawar, B. A. (2016, July). The impact of a 3d visual tool on female students attitude and performance in computer programming. In *2016 SAI Computing Conference (SAI)* (pp. 864-867). IEEE. doi: 10.1109/SAI.2016.7556080
- Andreasen, E., ve Møller, A. (2014, October). Determinacy in static analysis for jQuery. In *ACM SIGPLAN Notices* (Vol. 49, No. 10, pp. 17-31). ACM. doi: 10.1145/2714064.2660214
- Anttonen, M., Salminen, A., Mikkonen, T., ve Taivalsaari, A. (2011, March). Transforming the web into a real application platform: new technologies, emerging trends and missing pieces. In *Proceedings of the 2011 ACM Symposium on Applied Computing* (pp. 800-807). ACM. doi: 10.1145/1982185.1982357
- Başer, M. (2013). Bilgisayar programlamaya karşı tutum ölçeği geliştirme çalışması. *The Journal of Academic Social Science Studies*, 6(6), 199-215. doi: 10.9761/JASSS1702
- Beighley, L. (2010). *jQuery for Dummies*. Hoboken, NJ, USA : John Wiley & Sons Inc.
- Bers, M. U., Flannery, L., Kazakoff, E. R., ve Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers ve Education*, 72, 145-157. doi: 10.1016/j.compedu.2013.10.020
- Bibeault, B., ve Katz, Y. (2010). *jQuery in Action*. Stamford, CT, USA: Manning Publications.
- Brennan, K., ve Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting*

of the American Educational Research Association, Vancouver, Canada (Vol. 1, p. 25). Erişim adresi: <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>

Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö. E., Karadeniz, Ş. ve Demirel, F.(2016). *Nitel Araştırmalar. Bilimsel Araştırma Yöntemleri*. Ankara: Pegem A. Yayıncılık.

Charters, P., Lee, M. J., Ko, A. J., ve Loksa, D. (2014, March). Challenging stereotypes and changing attitudes: the effect of a brief programming encounter on adults' attitudes toward programming. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 653-658). ACM. doi:10.1145/2538862.2538938

Chu, A., Huber, J., Mastel-Smith, B., ve Cesario, S. (2009). “Partnering with seniors for better health”: computer use and internet health information retrieval among older adults in a low socioeconomic community. *Journal of the Medical Library Association: JMLA*, 97(1), 12. doi: 10.3163/1536-5050.97.1.003

Cooper, S., Dann, W., ve Pausch, R. (2003). Using animated 3d graphics to prepare novices for CS1. *Computer Science Education*, 13(1), 3-30. doi: 10.1076/csed.13.1.3.13540

Coşar, M. (2013). *Problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkileri* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 349113).

Çay, T., ve İşcan, F. (2002). Harita mühendisliğinde kullanılan programlama dilleri ve yazılımları. *Akademik Bilişim*, 2. Erişim adresi: <https://ab.org.tr/ab02/tammetin/34.doc>

de Kereki, I. F., ve Manataki, A. (2016). “Code yourself” and “a programar”: A bilingual mooc for teaching computer science to teenagers. In *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE. doi: 10.1109/FIE.2016.7757569

Demirer, V. ve Sak, N. (2016). Programming education and new approaches around the world and in Turkey. *Eğitimde Kuram ve Uygulama*, 12 (3), 521-546. Erişim adresi: <https://dergipark.org.tr/eku/issue/26697/280853>

- Du, J., Wimmer, H., ve Rada, R. (2016). "Hour of code": Can it change students' attitudes toward programming?. *Journal of Information Technology Education: Innovations in Practice*, 15, 53. doi: 10.28945/3421
- Durak, G., (2009), *Algoritma konusunda geliştirilen "programlama mantığı öğretici-P.M.Ö" yazılımının öğrenci başarısına etkisi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 245520).
- Ekiz, D. (2003). *Eğitimde araştırma yöntem ve metotlarına giriş*. Ankara: Anı Yayıncılık.
- Erekmekçi, M., ve Fidan, Ş. (2012). Oyunun tasarım platformları: Oyunun eğitim ve kültüre etkisi. *Batman Üniversitesi Yaşam Bilimleri Dergisi*, 1(1), 851-861. Erişim adresi: <https://dergipark.org.tr/download/article-file/313609>
- Ericson, B., ve McKlin, T. (2012, February). Effective and sustainable computing summer camps. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*(pp. 289-294). ACM. doi: 10.1145/2157136.2157223
- Feldgen, M., ve Clúa, O. (2004, October). Games as a motivation for freshman students learn programming. In *34th Annual Frontiers in Education, 2004. FIE 2004.* (pp. S1H-11). IEEE. doi: 10.1109/FIE.2004.1408712
- Fesakis, G., ve Serafeim, K. (2009, July). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. In *ACM SIGCSE Bulletin* (Vol. 41, No. 3, pp. 258-262). ACM. doi: 10.1145/1595496.1562957
- Fessakis, G., Gouli, E., ve Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers ve Education*, 63, 87-97. doi: 10.1016/j.compedu.2012.11.016
- Fisher, A., ve Margolis, J. (2002). Unlocking the clubhouse: The Carnegie Mellon experience. *ACM SIGCSE Bulletin*, 34(2), 79-83. doi:10.1145/543812.543836
- Fluck, A. E., Webb, M., Cox, M. J., Angeli, C., Malyn-Smith, J., Voogt, J., ve Zagami, J. (2016). Arguing for computer science in the school curriculum. *Educational Technology ve Society*, 19(3), 38-46. Erişim adresi: <https://hdl.handle.net/11245/1.547419>

- Gajraj, R., Singh, L., ve Malcolm, W. (2010). A computer based programming pedagogy: Stepwise instructed implementation of explained example code. Erişim adresi: https://www.academia.edu/1476129/A_COMPUTER-BASED_PROGRAMMING_Pedagogy_stepwise_instructed_implementation_of_explained_example_code
- Gibson, S., ve Bradley, P. (2017). A study of Northern Ireland Key Stage 2 pupils' perceptions of using the BBC Micro: bit in STEM education. *The STeP Journal*, 4(1), 15-41. Erişim adresi: <http://ojs.cumbria.ac.uk/index.php/step/article/view/374>
- Good, J., ve Howland, K. (2015, October). Natural language and programming: Designing effective environments for novices. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 225-233). IEEE. doi: 10.1109/VLHCC.2015.7357221
- Gökalp, S., ve Aydın, T. (2013). Teknik program öğrencilerinin bilgisayar ve internet kullanımına yönelik tutumlarının analizi. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 2(2), 127-134. Erişim adresi: <https://dergipark.org.tr/bitlisfen/issue/3701/48982>
- Gökdaş, İ. (2008). Bilgisayara Yönelik Tutumlar. D. Deryakulu (ed.), *Bilişim Teknolojileri Öğretiminde Sosyo-Psikolojik Değişkenler* (ss.125- 150). Ankara: Maya Akademi.
- Grover, S., ve Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43. doi: 10.3102/0013189X12463051
- Heersink, D., ve Moskal, B. M. (2010, March). Measuring high school students' attitudes toward computing. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 446-450). ACM. doi: 10.1145/1734263.1734413
- Hong, J., Fadjo, C., Chang, C. H., Geist, E., ve Black, J. (2010). Urban culture and constructing video games. In *EdMedia+ Innovate Learning* (pp. 1334-1337). Erişim adresi: <https://www.learntechlib.org/p/34808/>
- Hussain, A. J., Fergus, P., Al-Jumeily, D., Pich, A., ve Hind, J. (2015). Teaching primary school children the concept of computer programming. In *2015 International*

Conference on Developments of E-Systems Engineering (DeSE)(pp. 180-184). IEEE.

Erişim adresi: <https://ieeexplore.ieee.org/abstract/document/7563635>

Jay, G. M., ve Willis, S. L. (1992). Influence of direct computer experience on older adults' attitudes toward computers. *Journal of Gerontology*, 47(4), P250-P257. doi: 10.1093/geronj/47.4.P250

Kafai, Y. B., Peppler, K. A., ve Chiu, G. M. (2007). High tech programmers in low-income communities: Creating a computer culture in a community technology center. In *Communities and technologies 2007* (pp. 545-563). Erişim adresi: https://link.springer.com/chapter/10.1007/978-1-84628-905-7_27

Kalelioğlu, F., ve 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. Erişim adresi: <http://www.academia.edu/download/33652723/INFE232.pdf>

Karabak, D., ve Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 21(2-3), 163-169. Erişim adresi: <http://jret.org/FileUpload/ks281142/File/21.karabak.pdf>

Kzakoff, E. R., ve Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, 50(4), 553-573. doi: doi: 10.2190/EC.50.4.f

Kelleher, C., ve Pausch, R. (2003). Lowering the barriers to programming: a survey of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83-137. doi: 10.1145/1089733.1089734

Kelleher, C., Pausch, R., Pausch, R., ve Kiesler, S. (2007, April). Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464). ACM. doi: 10.1145/1240624.1240844

Kert, S. B., ve Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. In *The First International Congress of Educational Research, Çanakkale*,

Turkey.

Erişim

adresi:

http://www.academia.edu/download/34971233/Kert_Ugras_Scratch.pdf

- Lau, W. W. F. ve Yuen, A. H. K. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology*, 40 (4), 696-712. doi: 10.1111/j.1467-8535.2008.00847.x
- Lerner, B. S., Elbert, L., Li, J., ve Krishnamurthi, S. (2013). Combining form and function: Static types for JQuery programs. Erişim adresi: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.416.9552>
- Lindley, C. (2009). *JQuery Cookbook: Solutions ve examples for JQuery developers*. CA, USA: O'Reilly Media, Inc.
- Mahmoud, Q. H., Dobosiewicz, W., ve Swayne, D. (2004, March). Redesigning introductory computer programming with HTML, JavaScript, and Java. In *ACM SIGCSE Bulletin* (Vol. 36, No. 1, pp. 120-124). ACM. doi: 10.1145/1028174.971344
- Malan, D. J. (2015). CS50. Web: <https://cs.harvard.edu/malan/publications/CMU.pdf> adresinden erişildi.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., ve Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16. Erişim adresi: <https://www.learntechlib.org/p/50521/>
- Mikkonen, T., ve Taivalsaari, A. (2007). Using JavaScript as a real programming language. Erişim adresi: <https://dl.acm.org/citation.cfm?id=1698202>
- MEB. (2017). *Bilişim teknolojileri ve yazılım dersi öğretim programı (ortaokul 5. ve 6. sınıflar)*. Ankara: Milli Eğitim Bakanlığı. Web: <http://ttkb.meb.gov.tr/> adresinden erişildi.
- Mironova, O., Amitan, I., ve Vilipõld, J. (2017, April). Programming basics for beginners: Experience of the institute of informatics at Tallinn University of Technology. In *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 735-739). IEEE. doi: 10.1109/EDUCON.2017.7942929

- Molina-García, J. C., Rodríguez-Elías, O. M., Glasserman-Morales, L. D., ve Rodríguez-Pérez, J. M. (2016, April). Designing a strategy of programming learning for kids through the use of the "micompu. mx" federal program: A pilot study. In *2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)* (pp. 104-109). IEEE. doi: 10.1109/CONISOFT.2016.24
- Monroy-Hernández, A., ve Resnick, M. (2007). Empowering kids to create and share programmable media. *Education (ISTE)*. Erişim adresi: <https://arxiv.org/pdf/1507.01282.pdf>
- Nikiforos, S., Kontomaris, C., ve Chorianopoulos, K. (2013). MIT scratch: A powerful tool for improving teaching of programming. *Conference on Informatics in Education*, 1-5. Erişim adresi: http://cnseclab.di.ionio.gr/cie/images/documents13/CIE2013_proceedings/data/cie2013_020.pdf
- Nixon, R. (2014). *Learning PHP, MySQL ve JavaScript: With jQuery, CSS ve HTML5*. CA, USA: O'Reilly Media, Inc.
- Ocariza Jr, F. S., Pattabiraman, K., ve Zorn, B. (2011, November). JavaScript errors in the wild: An empirical study. In *2011 IEEE 22nd International Symposium on Software Reliability Engineering* (pp. 100-109). IEEE. doi: 10.1109/ISSRE.2011.28
- Özer, E. A. (2017). *Sürdürülebilir bir bilişim teknolojileri öğretim programı geliştirme modelinin (SÖPGEM) hazırlanması ve etkililiğinin incelenmesi* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 470998).
- Özyol, B. (2019). *Bilgi-işlemsel düşünme becerisinin kazandırılmasına yönelik bir ortam tasarımı ve geliştirilmesi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 544479).
- Özyurt, H., ve Özyurt, Ö. (2015). A study for determining computer programming students' attitudes towards programming and their programming self-efficacy. Erişim adresi: <https://dergipark.org.tr/download/article-file/63464>

- Powers, K. (2007). Through the looking glass: Teaching CS0 with alice. In *SIGCSE Bulletin*. Erişim adresi: http://pages.cs.wisc.edu/~strommen/projects/thesis/alice_paper.pdf
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... ve Kafai, Y. B. (2009). Scratch: Programming for all. *Commun. Acm*, 52(11), 60-67. doi: 10.1145/1592761.1592779
- Rodger, S. H. (2002, June). Using hands-on visualizations to teach computer science from beginning courses to advanced courses. In *Second Program Visualization Workshop* (pp. 103-112). Erişim adresi: <http://cs.duke.edu/csed/rodger/papers/pviswk02.pdf>
- Rodger, S. H., Hayes, J., Lezin, G., Qin, H., Nelson, D., ve Tucker, R. (2009). Engaging middle school teachers and students with alice in a diverse set of subjects. doi:10.1145/1539024.1508967
- Sáez-López, J. M., Román-González, M., ve Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers ve Education*, 97, 129-141. doi: 10.1016/j.compedu.2016.03.003
- Sarısakal, N., ve Uysal, M. (2001). Web teknolojilerindeki hızlı gelişmelerin ve web programlama araçlarının incelenmesi. *Istanbul University-Journal of Electrical ve Electronics Engineering*, 1 (1), 6-16. Erişim adresi: <https://dergipark.org.tr/download/article-file/98943>
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., ve Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351-380. doi:10.1007/s10639-012-9240-x
- Shashaani, L. (1997). Gender differences in computer attitudes and use among college students. *Journal of Educational Computing Research*, 16 (1), 37-51. doi: 10.2190/Y8U7-AMMA-WQUT-R512

- Silla, C. N., Przybysz, A. L., ve Leal, W. V. (2016, October). Music education meets computer science and engineering education. In *Frontiers in Education Conference (FIE), 2016 IEEE* (pp. 1-7). IEEE. doi: 10.1109/FIE.2016.7757413
- Smith, G., ve Sullivan, A. (2012, February). The five year evolution of a game programming course. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 87-92). ACM. doi:10.1145/2157136.2157168
- Taivalsaari, A. (2009). Mashware: The future of web applications. Eriřim adresi: <https://dl.acm.org/citation.cfm?id=1698213>
- Taivalsaari, A., Mikkonen, T., Anttonen, M., ve Salminen, A. (2011, January). The death of binary software: End user software moves to the web. In *2011 Ninth international conference on creating, connecting and collaborating through computing* (pp. 17-23). IEEE. doi: 10.1109/C5.2011.9
- Taivalsaari, A., Mikkonen, T., Ingalls, D., ve Palacz, K. (2008). Web browser as an application platform: The lively kernel experience. doi: 10.1109/SEAA.2008.17
- Tutorialspoint. (2015). JQuery web application library. Web: www.tutorialspoint.com/jquery/ adresinden eriřilmiřtir.
- Utting, I., Cooper, S., K lling, M., Maloney, J., ve Resnick, M. (2010). Alice, greenfoot, and scratch-a discussion. *ACM Transactions on Computing Education (TOCE)*, 10 (4), 17. doi:10.1145/1868358.1868364
- Yıldırım, A., ve Őimřek, H. (2013). *Sosyal bilimlerde nitel arařtırma y ntemleri*. Ankara: Seękin Yayıncılık.
- Yięit, M. F. (2016). *G rsel programlama ortamı ile  ğretimin  ğrencilerin bilgisayar programlamayı  ğrenmesine ve programlamaya karřı tutumlarına etkisinin incelenmesi* (Y ksek Lisans Tezi). Y K Tez Merkezi veri tabanından eriřildi (Tez No: 442990).
- Yolcu, V. (2018). *Programlama eęitiminde robotik kullanımının akademik bařarı, bilgi-iřlemsel d ř nme becerisi ve  ğrenme transferine etkisi* (Y ksek Lisans Tezi). Y K Tez Merkezi veri tabanından eriřildi (Tez No: 509835).

Yükseltürk, E., ve Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4 (1), 50-65. Erişim adresi: <https://dergipark.org.tr/amauefd/issue/1732/21264>

EKLER

ÖZGEÇMİŞ VE İLETİŞİM BİLGİSİ

İbrahim BAŞTUĞ, 1984 yılında Ankara’da doğdu. İlk ve orta eğitimini Ankara’da tamamladıktan sonra, 2002 yılında Çanakkale Onsekiz Mart Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Öğretmenliği bölümünde lisans eğitimine başladı. 2006 yılında bu bölümden mezun olduktan hemen sonra MEB’e bağlı eğitim kurumlarında bilgisayar öğretmeni olarak çalışmaya başladı. Halen görevine Kocaeli Başiskele Yuvacık Levent Kırca Oya Başar Ortaokulunda bilişim teknolojileri ve yazılım dersi öğretmeni olarak devam etmektedir. Evli ve bir çocuk babasıdır.

Yabancı Dil: İngilizce – İleri seviye.

E-postası:ibrahim_bastug@hotmail.com

GÖREVLER:

Görev Unvanı	Görev Yeri	Yıl
Öğretmen	Milli Eğitim Bakanlığı	2006-devam ediyor.

ESERLER:

1. Effect Of Javascript On Middle Schoolers' Attitude Toward Computer Course - Journal Of Pedagogical Research
2. Modeling The Psychological Factors Affecting Computer Programming Self-Efficacy - Anatolian Journal of Educational Leadership and Instruction
3. Predicting Cyberbullying Tendencies Of Adolescents With Problematic Internet Use - The Journal of Academic Social Science Studies
4. 5. ve 6. Sınıf Öğrencilerinin Programlama Eğitiminde Edmodo Kullanımına Yönelik Görüşlerinin Değerlendirilmesi - ITTES 2016 Elazığ
5. Sanal Ortamlarda Otantik Öğrenme - V. Sakarya’da Eğitim Araştırmaları Kongresi



T.C.
KOCAELİ VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : 99332089-605.01-E.3378727

15/02/2019

Konu: Araştırma İzni
(İbrahim BAŞTUĞ)

SAKARYA ÜNİVERSİTESİ REKTÖRLÜĞÜNE
(Eğitim Bilimleri Enstitüsü Müdürlüğü)

İlgi: 31/01/2019 tarihli ve 1341 sayılı yazımız.

Üniversiteniz Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı Yüksek Lisans öğrencisi İbrahim BAŞTUĞ'un "Jquery Tabanlı Blok Programlama Öğretiminin Programlamaya Yönelik Tutama Etkisinin Değerlendirilmesi" konulu araştırma çalışmasını İlimiz Başiskele ilçesi ortaokullarında uygulama talebinin uygun görüldüğüne ilişkin, 13/02/2019 tarih ve 3188600 sayılı Valilik Onayı ekte gönderilmiştir. Gereğini rica ederim.

Osman EKŞİ
Vali a.
Vali Yardımcısı

Ek: Valilik Onayı

Güvenli Elektronik İmza
Aslı ile Aynıdır.

15.02/2019


İbrahim TURAN
V.H.K.İ.

Körfez Mah. Ankara Karayolu Cad. No:129 Valilik Binası B Blok Kat:3
Elektronik Ağ: www.kocaelimem.meb.gov.tr
e-posta: stratejigelistirme41@meb.gov.tr

Bilgi için: İbrahim TURAN - V.H.K.İ.
Tel: (0262) 300 58 71
Faks: (0262) 321 15 54

Bu evrak güvenli elektronik imza ile imzalanmıştır. <https://evraksorgu.meb.gov.tr> adresinden 9cce-08a3-3ea7-9d5f-954d kodu ile teyit edilebilir.