



PROGRAMLAMA ÖĞRETİMİNDE KULLANILABİLECEK YAZILIMLARA İLİŞKİN ÖĞRETMEN GÖRÜŞLERİ*

Adem UZUN¹, Salih BALTALI²

Makale Bilgisi	Özet
DOI: 10.19171/uefad.561833	<p>Bu çalışmanın amacı BT öğretmenlerinin programlama öğretiminde kullanılabilecek bazı yazılımların kullanılabilirlik ve tasarım yönleriyle ilgili görüşlerini incelemektir. Çalışmada Scratch 1.4, Microsoft Small Basic 1.2, Microsoft Kodu Game Lab 1.4.64 ve Robomind 6.01 yazılımları incelenmiştir. Çalışmaya Bursa ilinde ortaokulda görev yapan 92 Bilişim Teknolojileri (BT) öğretmeni katılmıştır. Bu yazılımların arayüz tasarımı ve kullanılabilirlik incelemesinde Nielsen'in (2015) sezgisel (heuristic) rehberi esas alınmıştır. Bu rehberin orijinali, araştırma amaçları doğrultusunda düzenlenerek elektronik bir form halinde öğretmenlere iletilmiştir. Araştırma nicel araştırma desenlerinden tekil tarama modeli temel alınarak gerçekleştirilmiştir. Elde edilen verilerin frekans, ortalama ve yüzdeleri hesaplanıp katılımcıların araştırma konusuna göre fikri yönü incelenmiştir. Araştırmada elde edilen veriler ışığında, programlama öğretiminde yazılım olarak görsel araç kullanımının, öğrenciler için pozitif etkiye sahip olacağı yönünde görüşlere ulaşılmıştır. BT öğretmenleri bu araçların genel itibarıyla ortaokul kademelerine daha uygun olduğu yönünde görüş belirtmişlerdir. Bunun yanı sıra Small Basic 1.2 yazılımının ortaokulda kullanılabileceği gibi içerdiği kod tabanlı yapı nedeniyle profesyonel dillere daha yakın durduğundan lise kademesinde de yararlanılabileceği ortaya çıkmıştır. Diğer taraftan, katılımcılara göre yazılım hazırlanırken kullanılabilirlik ve tasarım yönlerinin her ikisine de dikkat edilmesi gerektiği sonucuna ulaşılmıştır. Görselleştirme araçlarının genel itibarıyla kullanılabilirlik yönlerinin ve tasarımlarının da iyi olduğu, bunlar arasından Robomind 6.01 yazılımının en yüksek düzeyde kullanılabilir olduğu ve tasarım bakımından en az soruna sahip olduğu sonucuna ulaşılmıştır.</p>
<i>Makale Geçmişi:</i>	
Başvuru 08.05.2019	
Kabul 22.10.2019	
<i>Anahtar Kelimeler:</i>	
Programlama öğretimi, görselleştirme yazılımı, kullanılabilirlik.	

TEACHERS' OPINIONS ABOUT SOFTWARE THAT CAN BE USED IN TEACHING PROGRAMMING

Article Info	Abstract
DOI: 10.19171/uefad.561833	<p>The purpose of this study is to examine the opinions of IT teachers about the usability and design aspects of some software that can be used in programming teaching. In this study, Scratch 1.4, Microsoft Small Basic 1.2, Microsoft Code Game Lab 1.4.64 and Robomind 6.01 software were examined. 92 IT teachers working in secondary schools in Bursa participated in the study. The interface design and usability analysis of the software was based on Nielsen's (2015) heuristic guide. The original version of this guideline was arranged in line with the research objectives and sent to teachers in an electronic form. The research was carried out based the survey model. The frequency, average and percentages of the data were calculated, and the opinions of the participants regarding the research subject were examined.</p>
<i>Article History:</i>	
Received 08.05.2019	
Accepted 22.10.2019	
<i>Keywords:</i>	
Teaching programming, visualization software,	

* Bu çalışma Salih Baltalı tarafından Uludağ Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı'nda Doç. Dr. Adem Uzun danışmanlığında yürütülen yüksek lisans tezinden üretilmiştir. Ayrıca çalışmada sunulan verilerin bir kısmı "The 4th International Symposium on New Issues in Teacher Education" (ISNITE 2016) sözlü bildiri olarak sunulmuştur.

¹ Doç. Dr., Uludağ Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, auzun@uludag.edu.tr, OrcID: 0000-0001-6935-346X

² Bilişim Teknolojileri Uzmanı, suvari58@gmail.com, OrcID: 0000-0002-7625-4926

usability.

Findings showed that using visual tools as software in programming instruction could have a positive effect on students' learning. IT teachers stated that these tools were generally more suitable for secondary schools. Participants also stated that Small Basic 1.2 software could be used at the high school level because it is closer to professional languages due to its code-based structure. On the other hand, participants remarked that both the usability and design aspects should be considered when developing software. These four visualization tools generally have good usability aspects and designs, among which the Robomind 6.01 software is the most readily available and has the least problem in terms of design.

GİRİŞ

1980'lerde sanayi toplumundan bilgi toplumuna geçiş yapıldığında temel bilgisayar bilgi ve becerilerinin de artık okuma yazma becerisinden sonra elde edilmesi gereken bir yeterlik olduğu tartışılmaya başlanmıştır (Drucker, 2000). Çoğunlukla bilginin değerli olduğu ve sadece ona erişmekle yetinilen ve bunun yeterli bulunduğu 1980-1990'lı yıllarda programlamanın önemi yavaş yavaş kavranmaya başlanıp 2000'li yıllarda bilgiye erişimden bilgi ve içerik üretimine geçiş yapılmıştır. Artık farklı ihtiyaçları karşılayabilecek yazılımlar programlama sayesinde üretilip insanlığın hizmetine sunulmuştur. Bilgisayarların eğitim, sağlık, ulaşım, haberleşme gibi birçok sektörde varlık göstermesi ve yayılması programlamanın cazibesini hiçbir zaman yitirmemesini sağlamıştır.

Programlamanın önemini bilen ve gelecekte kendilerine neler kazandırabileceğini öngören ülkelerde, eğitim politikaları içerisine dâhil edilip, uzun yıllardan bu yana çeşitli öğretim kademelerinde programlama eğitimi verilmiş, programlama öğretiminin önemli olduğu birçok kaynakta vurgulanmış ve konuyla ilgili çalışmalar yapılmıştır. Buna göre, Tucker ve diğerleri (2003) araştırmalarında Avrupa, Rusya, Asya, Güney Afrika, Yeni Zelanda ve Avustralya'da bilgisayar bilimleri dersinin K-12 öğretim programına eklendiğini belirtmektedirler. İsrail ve Kanada'da ise programlama eğitimi ile ilgili dersler liselerde verilmektedir.

Dünyada programlama dersleri, mesleki yeterlilik için gerekli olmanın yanında bilgisayar okuryazarlığının önemli bir parçası olmaları nedenleriyle yaygınlaşmaktadırlar. Zira bu tür derslerin öğrencilerin analitik düşünme ve problem çözme becerilerinin gelişmesi açısından faydalı olduğu bilinmektedir (Sleeman, Putnam, Baxter & Kuspa, 1984). Bu nedenlerden ötürü dünyada birçok ülkede programlamaya giriş dersleri ilk ve ortaöğretim öğretim programlarına eklenmiştir (Tucker et al., 2003).

Programlama öğretiminin ilköğretim düzeyinde verilmeye başlanmasıyla çocuklara programlama öğretiminin nasıl verileceği sorunu ortaya çıkmıştır. Eğitim kurumlarında programlama öğretimi genelde teorik bir yöntemle verilmektedir. Teorik yöntemin öğretimde hem sıkıcı hem de çok etkili olmadığı bilinmektedir (Arabacıoğlu, Bülbül ve Filiz, 2007). Programlama öğretimi kavramlarından olan koşul, döngü, şartlı döngü gibi temel programlama kavramlarının teorik olarak aktarılmasının, öğretimi zorlaştırabileceği düşünülmektedir. Soyut olarak geçen bu kavramlar zaten

zihinsel süreçleri bakımından somut evreden soyut evreye geçtikleri ortaokul yaşlarında öğrenci zihninde tam olarak oturamamakta; amaç, işlev ve sonuçları tam anlamıyla anlaşılabilir değildir. Bu problemi aşmak ve programlama kavramlarının anlaşılabilirliği için uzun yıllardan beri kullanılan öğretim yazılımları ve görselleştirme araçları bulunmaktadır. Bazı ülkeler öğretim programlarının içerisinde bu yazılımları kullanarak programlama öğretimi yapılmasını yönünde uygulamalar geliştirmişlerdir. Ancak bahsi geçen yazılım ve görselleştirme araçlarının belirli yaş ve sınıflara uygulanmasında öğrencilerin mevcut zihinsel algı durumları göz önüne alınırsa yazılımların kullanılabilirlik ve tasarımları açısından sorunlar oluşturacağı açıktır. Tasarım bakımından zayıf, o yaş veya sınıfa hitap etmeyen, kullanılabilirliği düşük olan bir programlama öğretimi aracının öğrenciye vereceği bir şey olamayacağı gibi karmaşıklığa ve yanlış öğrenmeye sebep olma ihtimali de vardır. Bu yüzden öğretimde kullanılacak yazılımların o dersi verecek öğretmen tarafından doğru seçilmesi veya öğretmen görüşlerinin alınması gerekmektedir.

Kelleher, Pausch ve Kiesler (2007), programlama eğitiminde görselliğin önemini vurgulamak amacı ile gerçekleştirdikleri araştırma sonucunda, görsel özelliklerin öğrenci başarı ve motivasyonunda önemli etkisi olduğunu belirlemişlerdir. Konu ile ilgili yapılan diğer araştırmalar, yazılımlarda grafik ve animasyon kullanımının öğrencilerin derslere olan ilgisini arttırmada daha etkili olduklarını göstermektedir (Bishop-Clark, Courte & Howard, 2007; Lin & Zhang, 2003). Tüm bu nedenlerden ötürü, görsellik ve canlandırma içeren yardımcı araçlarla yapılacak uygulamaların uygun yöntemlerle derslere dâhil edilmesi durumunda, ilköğretim öğrencilerinin programlamaya ilgi duymalarını ve sevmelerini sağlayabileceği söylenebilir. Araştırmalar, yazılımda kullanılan görselliğin ve etkileşimin öğrenci başarısı üzerindeki etkisini vurgulamaktadır (Arabacıoğlu ve ark., 2007; Gültekin, 2006). Lahtinen, Ahoniemi ve Salo (2007), araştırma sonuçlarına göre problemlerin karmaşıklığı arttıkça görselleştirme araçlarına daha çok ihtiyaç duyulmaktadır. Yapılan araştırmalar genellikle üniversite düzeyinde gerçekleştirilmiş olup, ilköğretim ve ortaokul seviyesine uygun görselleştirme aracının belirlenmesinde yetersiz kalmaktadır. Bu nedenle ilköğretim ve ortaokul seviyesine programlamanın temelleri öğretilirken, hangi görselleştirme araçlarını kullanmanın ve bunların kullanılabilirliği ve tasarımı bakımından hangilerinin daha uygun olacağını araştırılmasında yarar vardır.

Bu araştırmanın amacı, BT öğretmenlerinin programlama öğretiminde kullanılabilecek Scratch 1.4; Small Basic 1.2; Microsoft Kodu Game Lab 1.4.64 ve Robomind 6.0.1 yazılımlarını, Nielsen'in (2015) sezgisel kontrol aracına göre kullanılabilirlik ve tasarım açısından görüşlerini analiz etmektir. Bu yazılımlar literatürdeki çalışmalar (Akçay, 2003; Calder, 2010; Prawalpatagool, 2010; Kaucic ve Asic, 2010; Fesakis ve Serafeim, 2009) ve okullarda görev yapan bilişim teknolojileri (BT) öğretmenlerinin derslerinde bu yazılımları kullanma durumları göz önünde bulundurularak seçilmiştir. Araştırma kapsamında aşağıdaki sorulara cevaplar aranacaktır:

1. BT öğretmenleri programlama öğretiminde kullanılan görselleştirme araçlarının kullanımını genel olarak nasıl değerlendirmektedir?

- BT Öğretmenleri görselleştirme araçlarını faydalı bulmakta mıdır?
- BT Öğretmenleri bu araçlardan hangilerinin hangi öğretim kademesinde kullanılması gerektiğini düşünmektedir.
- BT Öğretmenleri bu araçlarda kullanılabilirlik ve tasarım kavramlarından hangisine daha çok önem vermektedir?
- BT Öğretmenlerinin bu araçların kullanılabilirliğine ilişkin görüşleri nelerdir?
- BT Öğretmenlerinin bu araçların tasarımına ilişkin görüşleri nelerdir?

2. BT öğretmenlerine göre kullanılan görselleştirme araçlarının kontrol aracı başlıklarına göre öne çıkan sorunları nelerdir?

3. BT öğretmenlerine göre kullanılan görselleştirme araçlarının sahip olduğu sorunlar nelerdir?

4. BT öğretmenlerine göre kullanılan görselleştirme araçlarının kullanılabilirlik ve tasarım açılarından sorunları nelerdir?

YÖNTEM

Araştırmanın Modeli

Araştırma nicel araştırma desenlerinden tekil tarama modeli temel alınarak gerçekleştirilmiştir. Değişkenlerin tek tek, tür ya da miktar olarak oluşumlarının belirlenmesi amacı ile yapılan araştırma modellerine tekil tarama modelleri denir. Bu tür bir yaklaşımda, ilgilenilen olay, madde, birey, grup, kurum, konu vb. birim ve duruma ait değişkenler, ayrı ayrı tanıtılmaya çalışılır (Karasar, 2013). Tekil tarama modellenli araştırmalarda, daha çok, betimsel istatistik teknikleri kullanılır. Araştırmacı, ortalama (medyan), tepe değer (mod), standart sapma, değişkenlik (varyans), dizi genişliği (ranj), frekans dağılımı, normal dağılım, oran, yüzde vb. analizleri kullanır (Karasar, 2013). Genellikle geniş popülasyonlar için uygun olup, araştırmacının uyguladığı anketler ve görüşme planları yaygın veri toplama tekniklerinden ikisidir. Her ikisi de soruların standardizasyonunu gerektirir.

Katılımcılar

Araştırmaya, Bursa ilinde 2014-2015 eğitim-öğretim yılında tamamı ortaokulda Bilişim Teknolojileri (BT) öğretmeni olarak görev yapan 92 öğretmen katılmıştır. Katılımcılar Bursa'nın 16 farklı ilçesinden gönüllülük esasına göre seçilmiştir.

Araştırmaya katılan BT öğretmenlerinin cinsiyete göre dağılımı ise Tablo 1'de gösterilmiştir.

Tablo 1.
Katılımcıların Cinsiyet Bilgileri

Cinsiyet	Kişi Sayısı
Kadın	35
Erkek	57
Toplam	92

Katılımcıların Milli Eğitim Bakanlığı (MEB)'ndaki kıdem süreleri Tablo 2'de gösterilmiştir.

Tablo 2.
Katılımcıların Mesleki Kıdem Bilgileri

Mesleki Kıdem	1-3 yıl	4-6 yıl	7-9 yıl	10-12 yıl	13-15 yıl	16-18 yıl
Katılımcı Sayısı	11	26	34	19	1	1

Katılımcıların sahip oldukları programlama bilgisine ilişkin öz-yeterlik algıları ise Tablo 3'te gösterilmiştir. Buna göre öğretmenlerden sahip oldukları programlama bilgilerini 1 ile 10 arasında bir derece vermeleri istenmiştir.

Tablo 3.
Katılımcıların Programlama Bilgisi Öz-Yeterlik Algıları

Düzye	1	2	3	4	5	6	7	8	9	10
Katılımcı Sayısı	-	1	4	6	18	10	28	18	7	-

Tablo 3 incelendiğinde, katılımcıların çoğunun programlama bilgisi öz-yeterlik algısının yüksek olduğu görülmektedir.

Veri Toplama Aracı

Araştırmada elde edilen veriler, Nielsen'in (2015) Sezgisel Kontrol Aracı kullanılarak toplanmıştır. Bu araç yazılımların kullanıcı arayüzlerinin tasarımında dikkat edilmesi gereken hususların çok boyutlu olarak değerlendirilmesi için kullanılan oldukça detaylı bir araçtır. Fakat bu araştırma kapsamında, konu alanı uzmanı eşliğinde araştırma konusuna göre yazılım kullanılabilirliği ve tasarımını inceleyen sorular haricindeki sorular kontrol aracından çıkarılmıştır. Bu şekilde araç araştırmanın amacına uygun olacak şekilde adapte edilmiştir. Bu araçta BT öğretmenlerinin demografik bilgilerinin alınması dışında programlama öğretiminde kullanılan görselleştirme yazılımlarının kullanılabilirlik ve tasarım kavramları bakımından incelenmesini sağlayan 12 ayrı başlık bulunmaktadır. Bu başlıklar şu şekildedir:

1. Sistem Durumunun Görünürlüğü;
2. Sistem ve Gerçek Dünyanın Eşleşmesi;
3. Kullanıcı Kontrol ve Özgürlüğü;
4. Tutarlılık ve Standartlar;
5. Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma;
6. Hataları Önleme;
7. Hatırlama Yerine Tanıma;
8. Esneklik ve Verimlilik;
9. Estetik ve Sade Tasarım;
10. Yardım ve Dokümantasyon;
11. Yetenekler;
12. Kullanıcı ile Seviyeli Bir İletişim.

Her başlığın altında o başlıkla ilgili bahsedilen kavramın olup olmadığını tespit amacıyla çeşitli sorular bulunmaktadır. Soruların Evet, Hayır ve Uygulaması Yok şeklinde üç farklı seçeneği vardır. Eğer soruda bahsedilen özellik görselleştirme aracında bulunuyorsa Evet, istenileni karşılamıyorsa Hayır, aranan özelliğin yazılımda bir karşılığı bulunmuyorsa yani yoksa Uygulaması Yok cevabı verilecektir.

Oluşturulan yeni sezgisel kontrol aracı BT öğretmenlerine Google Forms arayüzünde oluşturulan bir form haline öğretmenlerin e-posta hesaplarına gönderilmiştir.

Veri Toplama Süreci

Araştırmaya katılan BT öğretmenleri sezgisel kontrol aracını 14 Nisan 2015 ve 30 Mayıs 2015 tarihleri arasında doldurmuşlardır. Araştırmaya katılmayı kabul eden BT öğretmenlerine bir e-posta gönderilmiştir. Bu e-postada araştırmacı ve araştırma konusu ile ilgili açıklamalar, tasarım ve kullanılabilirlik değerlendirilmesi yapılacak yazılım ile ilgili bilgilerin olduğu bir web linki ve değerlendirmenin yapılabilmesi için gereken anket formunun linki yer almaktadır. Her BT öğretmeni kendisine e-posta yoluyla gelen programlama öğretiminde kullanılan görselleştirme aracını kullanarak onun kullanılabilirlik ve tasarım açısından değerlendirmesini yapmıştır.

Verilerin Analizi

Araştırma için Google Forms platformunda toplanan veriler Excel formatında dışarı aktarılmıştır. Girilen verilerin geçerlilikleri konu alanı uzmanıyla kontrol edilmiştir. Buna göre sezgisel kontrol aracında yer alan ana başlıkların altındaki sorulara verilen Evet, Hayır ve Uygulaması Yok yanıtlarının ayrı ayrı frekans, yüzde ve ortalamaları Excel ve SPSS paket programları kullanılarak hesaplanmıştır. Elde edilen bulgular tablolar halinde gösterilerek yorumlanmıştır.

BULGULAR

Araştırmada elde edilen bulgular, araştırma sorularında yer alan sıralamaya göre başlıklar halinde gruplandırılarak sunulmuştur.

BT Öğretmenlerinin Programlama Öğretiminde Görsel Araç Kullanımına Genel Bakışları

BT öğretmenlerinin kendilerine sorulan “Programlama öğretiminde görsel araçların kullanılmasını nasıl değerlendiriyorsunuz?” sorusuna verdikleri yanıtlar Tablo 4’te sunulmuştur.

Tablo 4.

BT Öğretmenlerinin Programlama Öğretiminde Görsel Araç Kullanımına Genel Bakışları

Soru – 6	Scratch	Small Basic	Kodu Game	Robomind	Toplam
Çok faydasının dokunacağı kanaatindeyim.	20	19	19	20	78
Kararsızım.	0	1	3	1	5
Yararlı olacağını düşünmüyorum	1	4	2	2	9

Tablo 4’e göre BT öğretmenlerinin çoğunun (%84,7) programlama öğretiminde görsel bir yazılım kullanmaya olumlu baktığı görülmektedir.

Görsel yazılımların hangi öğretim kademelerinde kullanılması gerektiğine dair görüşler Tablo 5’te verilmiştir.

Tablo 5.

Öğretim Kademelerine Göre Görselleştirme Araçlarının Dağılımı

Soru – 7	Scratch	Small Basic	Kodu Game	Robomind	Toplam
İlkokul 3-4	0	0	1	0	1
İlkokul tüm sınıflar	0	1	1	1	3
İlkokul Toplam	1	1	2	1	4 (%4,3)
Ortaokul 5-6	9	1	4	9	23
Ortaokul 7-8	1	8	4	2	15
Ortaokul tüm sınıflar	7	7	8	7	29
Ortaokul Toplam	17	16	16	18	67 (%72,8)
Lise 9-10	0	1	1	2	4
Lise 11-12	0	1	0	0	1
Lise tüm sınıflar	1	3	0	0	4
Lise Toplam	1	5	1	2	9 (%9,7)
İlkokul ve Ortaokul	2	0	3	1	6
Ortaokul ve Lise	1	2	2	1	6
İki Kademe Toplam	3	2	5	2	12 (%13,0)
Genel Toplam	21	24	24	23	92

Tablo 5'e göre BT öğretmenleri kullanılan bütün yazılımların yüksek oranda (%72,8) ortaokul kademesine hitap edebileceği yönünde görüş bildirilmiştir.

Görselleştirme araçlarında kullanılabilirlik ve tasarım kavramlarından hangilerine daha çok dikkat edilmesi gerektiğine dair görüşler Tablo 6'da verilmiştir.

Tablo 6.
BT Öğretmenlerinin Görselleştirme Araçlarında Kullanılabilirlik ve Tasarıma Dikkat Etmeye İlişkin Görüşleri

Soru – 8	Scratch	Small Basic	Kodu Game	Robomind	Toplam
Kullanılabilirlik	1	4	1	2	8
Tasarım	1	1	2	0	4
Her ikisi de	19	19	21	21	80

Tablo 6'ya göre BT öğretmenlerinin kullanılan bütün yazılımlarda her iki kavramın birlikte göz önüne alınması gerektiği yönünde görüş bildirdiklerini görülmektedir.

Tablo 7'de BT öğretmenlerinin kullandıkları görselleştirme araçlarının kullanılabilirliğini değerlendirdiği görüşler yer almaktadır.

Tablo 7.
BT Öğretmenlerinin Görselleştirme Araçlarının Kullanılabilirliklerine İlişkin Görüşleri

Soru – 9	Scratch	Small Basic	Kodu Game	Robomind	Toplam
1 (Çok kötü)	0	0	0	0	0
2	0	0	1	0	1
3	3	10	4	2	19
4	16	12	8	20	56
5 (Çok iyi)	2	2	11	1	16

Tablo 7'deki bilgilere göre BT öğretmenleri kullanılabilirlik bakımından %60 oranında 4 (İyi) derecesinde yoğunlaşmışlardır. 19 kişi bütün yazılımlara 3 derecesini verirken 56 kişi ise 4 derecesini uygun görmüştür. Ayrıca Kodu Game hariç diğer yazılımlar 3 ve 4 derecelerinin toplamında birbirlerine yakın değerler almıştır.

BT öğretmenleri kullandıkları görselleştirme araçlarının tasarımlarına ilişkin değerlendirmeleri Tablo 8'de verilmiştir.

Tablo 8.
BT Öğretmenlerinin Görselleştirme Araçlarının Tasarımlarına İlişkin Görüşleri

Soru – 10	Scratch	Small Basic	Kodu Game	Robomind	Toplam
1 (Çok kötü)	0	0	0	0	0
2	0	0	2	1	3
3	3	10	5	7	25
4	16	11	8	12	47
5 (Çok iyi)	2	3	9	3	17

Tablo 8’de BT öğretmenlerinin %51’inin (47 kişi) görselleştirme araçlarının tasarımlarına 4 derecesini, %27,1’i (25 kişi) ise 3 derecesini verdiği görülmektedir. 3 ve 4 derecelerinin toplamında yine Kodu Game hariç diğerler yazılımlar birbirlerine yakın değerler almıştır.

BT Öğretmenlerinin Kontrol Aracı Başlıklarına Göre Öne Çıkan Sorunlara İlişkin Görüşleri

Sistem Durumunun Görünürlüğü

Sezgisel kontrol aracına göre, sistem (görselleştirme aracı) kullanıcıyı sürekli olarak durumunun nasıl gittiği konusunda gerekli zamanlarda gerekli geribildirim vererek bilgilendirmelidir. BT öğretmenlerinin görüşlerine göre Tablo 9’da görünürlük bakımından aşağıdaki verilere ulaşılmıştır.

Tablo 9.
BT Öğretmenlerinin Sistem Durumu Görünürlüğü Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	66	84	97	98	86,25
Hayır	17	15	12	9	13,25
Uygulaması Yok	22	21	11	8	15,5

Tablo 9’a göre Scratch yazılımı önemli bir farkla Uygulaması Yok ortalamasının (15,5) üstünde bir frekans almıştır. Scratch yazılımı bu başlığın 2. ve 3. sorularına sırasıyla 10 ve 9 Uygulaması Yok cevabı almıştır. İlgili sorular şu şekildedir:

Her menüdeki bilgilendirmeler, hatırlatmalar ve hata mesajları sistem içerisinde aynı yerde görünmekte midir?

Eğer hata mesajları açılan bir alan içerisinde gösteriliyor ise, kullanıcı, hatalı olan alanı görebilmekte midir?

Sistem ve Gerçek Dünyanın Eşleşmesi

Sezgisel kontrol aracına göre, sistem içerisinde kullanılan kelimeler, kavramlar, cümleler kullanıcıya tanıdık olmalıdır ve teknik bir dil kullanılmasından kaçınılmalıdır. Bilgi mantıksal sırasında ve doğallığında gösterilerek gerçek dünyada olan şekliyle sunulmalıdır. Bu maddeye ilişkin BT öğretmenleri görüşleri Tablo 10’da gösterilmiştir.

Tablo 10.
BT Öğretmenlerinin Sistem ve Gerçek Dünyanın Eşleşmesi Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	108	129	120	132	122,25
Hayır	24	23	38	21	26,5
Uygulaması Yok	15	15	10	8	12

Tablo 10’a göre Kodu Game yazılımı 38 Hayır cevabıyla ortalama üstü bir frekans almıştır. Buna göre Kodu Game aracı, bu başlığın 2. ve 6. sorularına sırasıyla 9 ve 12 Hayır cevabı verilmiştir. İlgili sorular şu şekildedir:

Eğer görsel bir ipucu olarak kullanılan şekil varsa, bu kültürel geleneklerle eşleşiyor mu?

Sistem kullanıcı jargonunu kullanıp ve bilgisayar jargonundan sakınmakta mıdır?

Kullanıcı Kontrol ve Özgürlüğü

Sezgisel kontrol aracına göre, kullanıcılar sık sık sistem fonksiyonlarının seçiminde hata yapar ve bu istenmeyen durumda çok detaya girmeden çıkmak için açıkça belirtilmiş bir “acil çıkış” a ihtiyaç duyarlar. Geri alma (undo) ve yeniden yapma (redo) seçenekleri bu amaçla sunulmaktadır. Araçları bu açıdan incelediğimizde Tablo 11’de ki sonuçlar ortaya çıkmıştır.

Tablo 11.
BT Öğretmenlerinin Kullanıcı Kontrol ve Özgürlüğü Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	101	131	129	138	124,75
Hayır	29	24	28	18	24,75
Uygulaması Yok	17	13	11	5	11,5

Tablo 11’e göre incelenen yazılımlar içinde Scratch yazılımının istenen özellikleri karşılamada ortalamanın (124,75) altında kaldığı görülmektedir. Scratch, bu başlığın 2. ve 3. sorularına sırasıyla 10 Uygulaması Yok ve 11 Hayır yanıtı alarak kullanıcı kontrol ve özgürlüğünde istenilen özellikleri yeteri kadar verememiştir. İlgili sorular aşağıdaki gibidir.

Açılır pencere olan sistemlerde, kullanıcılar için pencereler arası geçiş kolay mıdır?

Herhangi bir işlem hareketi, veri girişi veya birçok diğer işlem hareketleri için "geri al" fonksiyonu bulunmakta mıdır?

Tutarlılık ve Standartlar

Sezgisel kontrol aracına göre, kullanıcılar, farklı kelime, durum ve hareketlerin aynı şeyi ifade edip etmediğini bilmek zorunda değildirlere. Ortak bir tutarlılık tüm sistemde geçerli olmalıdır. Bu maddeye ilişkin öğretmen görüşleri Tablo 12’de sunulmuştur.

Tablo 12.

BT Öğretmenlerinin Tutarlılık ve Standartları Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	161	187	180	183	177,75
Hayır	36	43	52	49	45
Uygulaması Yok	24	28	24	12	22

Tablo 12’ye göre tutarlılık bakımından Kodu Game yazılımı Hayır ortalamasının (45) üstüne çıkmıştır. Detaylara bakıldığında Kodu Game’de bu başlığın 5. ve 6. sorularına sırasıyla 11 ve 10 Hayır yanıtı verilmiştir. Ayrıca Small Basic yazılımı da 28 ortalamayla Uygulaması Yok ortalamasının üstündedir. Bunda da 6. soruya 10 Uygulaması Yok yanıtı verilmiştir. İlgili sorular şu şekildedir:

Menüler dikey olarak sunulmuş mudur?

Eğer "çıkış" bir menü seçeneği ise listenin her zaman en altında yer almakta mıdır?

Kullanıcıların Hataları Tanınmasına, Onları Belirlemesine ve Önlemesine Yardımcı Olma

Sezgisel kontrol aracına göre, hata geri dönütleri, sade bir dilde (kodsuz) olmalı, sorunu açıklamalı ve yapıcı çözüm önerisi sunmalıdır. Bu maddeye ilişkin görüşler Tablo 13’te verilmiştir.

Tablo 13.

BT Öğretmenlerinin Hata Tanıma ve Önleme Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	23	95	88	108	78,5
Hayır	44	56	20	43	40,75
Uygulaması Yok	80	17	60	10	41,75

Tablo 13’e göre Scratch yazılımı Evet seçeneğinde ortalama altı katı Uygulaması Yok seçeneğinde ortalama üstüne (2 katı) çıkmıştır. Bu başlığın 2,3,4,5,6. ve 7. sorularına sırasıyla 9,11,13,14,13,12 Uygulaması Yok yanıtı verilmiştir. Diğer taraftan Small Basic yazılımı ise Hayır seçeneğinde ortalama üstüne çıkmıştır. Burada da 1,4 ve 7. sorulara sırasıyla 18,12 ve 13 Hayır yanıtı verilmiştir. Kodu Game yazılımının ise Uygulaması Yok seçeneğinde 60 frekansla ortalama üstünde

olduğu görülmektedir. Bu yazılım için 4,6 ve 7. sorulara sırasıyla 12,11 ve 12 Uygulaması Yok cevabı verilmiştir. İlgili sorular şu şekildedir:

Bir hatayı bildirmek için ses kullanılıyor mu?

Hatırlatıcılar, işlemin kullanıcının kontrolünde olduğunu ima etmekte midir?

Hatırlatıcılar, kısa, net ve anlaşılır mı?

Hata mesajları kullanıcının değil, sistemin sorumlu olduğunu belirtecek şekilde ifade edilmekte midir?

Hata mesajları gramer açısından doğru mudur?

Hata mesajları sorunun nedeni hakkında bilgiler sunuyor mu?

Hata mesajları, kullanıcıya hatayı düzeltmek için hangi hareketi yapması gerektiğini belirtiyor mu?

Hataları Önleme

Sezgisel kontrol aracına göre, hata mesajları dikkatli ve birinci aşamada oluşan problemleri önleyen bir yapıda olmalıdır. Bu maddeye ilişkin öğretmen görüşleri Tablo 14’te verilmiştir.

Tablo 14.
BT Öğretmenlerinin Hataları Önleme Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	46	59	62	59	56,5
Hayır	39	42	32	50	40,75
Uygulaması Yok	22	19	26	6	18,25

Tablo 14’te Robomind yazılımı Hayır seçeneğinde ortalamanın (40,75) üstündedir. Bu başlığın 2,3. ve 5. sorularına sırasıyla 10,18 ve 11 Hayır yanıtı verilmiştir. Diğer taraftan Kodu Game yazılımı ise Uygulaması Yok seçeneğinde ortalama üzerindedir. Aynı sorulara bu seçenekte sırasıyla 10,13 ve 10 Uygulaması Yok cevabı verilmiştir. İlgili sorular aşağıdaki şekildedir.

Veri girişleri mümkün olduğu kadarıyla büyük ve küçük harfe duyarlı olarak girilebiliyor mu?

Ciddi sonuçları olabilecek gösterge/düğmeler ulaşılması zor bir yerde midir?

Sistem, kullanıcıları sonuçları ciddi ve yıkıcı olabilecek hatalar yapmak üzere iken onları uyarıyor mu?

Hatırlama Yerine Tanıma

Sezgisel kontrol aracına göre, nesne, hareket ve seçenekler görünür olmalıdır. Kullanıcı, sistem boyunca kullanması gereken bilgiyi hatırlamak zorunda kalmamalıdır. Hatırlatma/bilgilendirme/açıklama bölümleri kullanıcının kolaylıkla ulaşabileceği şekilde sistemde yer almalıdırlar. Bu maddeye ilişkin öğretmen görüşleri Tablo 15’te sunulmuştur.

Tablo 15.

BT Öğretmenlerinin Hatırlama Yerine Tanıma Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	134	151	164	149	149,5
Hayır	23	27	23	28	25,25
Uygulaması Yok	11	14	5	7	9,25

Tablo 15'e göre Small Basic yazılımı Uygulaması Yok seçeneğinde ortalama üzerindedir. Detaylara bakıldığında bu başlığın 6. ve 7. sorularına 4 Uygulaması Yok yanıtı verilmiştir. İlgili sorular aşağıdaki şekildedir:

Anlamli grupları belirlemek için çerçeveler kullanılmış mıdır?

İlişkili elemanları gruplamak için aynı renkler kullanılmış mıdır?

Esneklik ve Verimlilik

Sezgisel kontrol aracına göre, acemi kullanıcılar tarafından fark edilemeyen hızlandırıcılar, uzman kullanıcıların sistemle etkileşimini artırabilir. Bu nedenle sistem hem deneyimli hem deneyimsiz kullanıcılara hitap edecek şekilde tasarlanmalıdır. Standart kullanıcılardan ziyade farklı (fiziksel, bilişsel yetenek, kültür, dil, vs bakımlarından) kullanıcıların sisteme alternatif ulaşma ve işlem yapabilme olanağı sağlanabilmelidir. Bu maddeye ilişkin öğretmen görüşleri Tablo 16'da verilmiştir.

Tablo 16.

BT Öğretmenlerinin Esneklik ve Verimlilik Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	29	46	51	55	45,25
Hayır	28	34	31	34	31,75
Uygulaması Yok	31	16	14	3	16

Tablo 16'da incelenen yazılımlardan Scratch Uygulaması Yok seçeneğinde ortalama üstündedir. Buna göre bu başlığın 1. sorusuna 10 Uygulaması Yok yanıtı verilmiştir. İlgili soru aşağıdaki gibidir:

Eğer sistem acemi ve profesyonel kullanıcıların ikisini de destekliyor ise detaylı hata mesajları farklı düzeylerde sağlanmış mıdır?

Estetik ve Sade Tasarım

Sezgisel kontrol aracına göre, sistem alakasız ve az ihtiyaç duyulan bilgiyi içermemelidir. Bu maddeye ilişkin öğretmen görüşleri Tablo 17'de verilmiştir.

Tablo 17.
BT Öğretmenlerinin Estetik ve Sade Tasarım Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	100	116	119	116	112,75
Hayır	22	16	18	17	18,25
Uygulaması Yok	4	12	7	5	7

Tablo 17’de Scratch yazılımı Hayır seçeneğinde ortalama üzerindedir. Ayrıntılı bakıldığında bu başlığın 6. sorusuna 10 Hayır yanıtı verilmiştir. İlgili soru aşağıdaki gibidir:

Sistem, veri girmek için kutu dışında açılır kutu, radyo butonu gibi başka seçenekler sunuyor mu?

Yardım ve Dokümantasyon

Sezgisel kontrol aracına göre, sistemi kullanmak için belgeleme olmaması iyi olacağı hâlde, gerekli durumlarda yardım ve belgeleme sağlanması önerilmektedir. Bu tür yardım bilgileri taranması kolay, kullanıcı görevlerim açıklayıcı ve kısa olmalıdır. Bu maddeye ilişkin öğretmen görüşleri Tablo 18’de sunulmuştur.

Tablo 18.
BT Öğretmenlerinin Yardım ve Dokümantasyon Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	81	55	97	107	85
Hayır	26	42	20	18	26,5
Uygulaması Yok	7	34	14	2	14,25

Tablo 18’e göre Small Basic yazılımı Hayır ve Uygulaması Yok seçeneklerinde ortalama üzerinde çıkmıştır. Hayır seçeneğinde 3. ve 5. sorulara sırasıyla 10,17 Hayır, Uygulaması Yok seçeneğinde 1. ve 6. sorulara sırasıyla 9,12 Uygulaması Yok yanıtı verilmiştir. İlgili sorular aşağıdaki gibidir.

Çevrim içi bilgilendirmeler (online yardım gibi) görsel olarak ayırt edilebilir durumda mıdır?

Yardım işlevi görünür mü? Örneğin YARDIM adında bir düğme veya özel bir menü mevcut mudur?

Kullanıcı, var olan detay seviyesini değiştirebiliyor mu?

Yardım sistemine kolayca ulaşılabilir mi veya bu bölümden kolayca çıkılabilir mi?

Yetenekler

Sezgisel kontrol aracına göre, sistem kullanıcının yeteneklerinin ve uzmanlığının gelişmesine ve fazlalaşmasına olanak sağlamalıdır. Bu maddeye ilişkin öğretmen görüşleri Tablo 19’da sunulmuştur.

Tablo 19.
BT Öğretmenlerinin Yetenekler Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	107	109	114	125	113,75
Hayır	14	25	21	9	17,25
Uygulaması Yok	5	10	9	4	7

Tablo 19’a göre Small Basic yazılımı Hayır ve Uygulaması Yok seçeneklerinde ortalama üzerinde çıkmıştır. Hayır seçeneğinde 4. ve 5. sorulara sırasıyla 6,8 Hayır, Uygulaması Yok seçeneğinde 4. soruya 4 Uygulaması Yok yanıtı verilmiştir. İlgili sorular aşağıdaki gibidir.

Kullanıcılar, bir alan içerisinde ileri ve geri hareket edebiliyorlar mı?

Fonksiyonelliği desteklemeye yetecek kadar sayıda, ancak aramayı ve bulmayı zorlaştırmayacak kadar yeterli işlev komutu (ekranda) var mı?

Kullanıcı ile Seviyeli İletişim

Sezgisel kontrol aracına göre, kullanıcının sistemle etkileşimi onun iş yaşamındaki kaliteyi arttırmalıdır. Kullanıcıya saygılı davranılmak ve tasarım görünüş ve işleme bakımından memnun edici olmalıdır. Bu başlığa göre verilen cevaplar Tablo 20’de sunulmuştur.

Tablo 20.
BT Öğretmenlerinin Kullanıcı ile Seviyeli Bir İletişim Boyutuna İlişkin Görüşleri

Seçenekler	Scratch	Small Basic	Kodu Game	Robomind	Ortalama
Evet	105	128	121	121	118,75
Hayır	17	13	22	16	17
Uygulaması Yok	4	3	2	1	2,5

Tablo 20’ye göre Kodu Game Hayır seçeneğinde ortalama üzerindedir. Bu başlığın 1. sorusuna 5 Hayır yanıtı verilmiştir. İlgili soru aşağıdaki gibidir:

Her bir ikon (simge), bir ikon grubunun uyumlu bir ögesi midir?

BT Öğretmenlerinin Görselleştirme Araçlarının Sahip Olduğu Sorunlara İlişkin Görüşleri

Sezgisel kontrol aracı başlıklarına göre yapılan yukarıdaki ayrı ayrı analizler sonucu öne çıkan ve farklılık gösteren (problemlerin/sorunların oluştuğu) ilgili soruların özetlenmiş hali Tablo 21’de sunulmuştur.

Tablo 21.

Kontrol Aracı Başlıklarına Göre Öne Çıkan Problemlerin Frekansları

	Toplam	19	21	21	31	150	72	8	10	10	48	18	15	423
Robomind	Hayır						39						10	49
	U.Yok													
Kodu Game	Hayır		21		21		13						5	60
	U.Yok					35	20							55
Small Basic	Hayır					43					27	14		84
	U.Yok				10			8			21	4		43
Scratch	Hayır			11						10				21
	U.Yok	19		10		72			10					111
	Kontrol Aracı Başlıkları													
	1. Sistem Durumunun Görünürlüğü													
	2. Sistem ve Gerçek Dünyanın Eşleşmesi													
	3. Kullanıcı Kontrol ve Özgürlüğü													
	4. Tutarlılık ve Standartlar													
	5. Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma													
	6. Hataları Önleme													
	7. Hatırlama Yerine Tanıma													
	8. Esneklik ve Verimlilik													
	9. Estetik ve Sade Tasarım													
	10. Yardım ve Dokümantasyon													
	11. Yetenekler													
	12. Kullanıcı ile Seviyeli Bir İletişim													
	Yanıtların Toplamı													

Tablo 21'e göre bütün yazılımlar için kontrol aracı başlıklarından *Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma* başlığında en çok problemin (150) oluştuğunu görmekteyiz. Bunu sırasıyla *Hataları Önleme* (72), *Yardım ve Dokümantasyon* (48) ve *Tutarlılıklar ve Standart* (31) başlıklarında sorunlar bulunmuştur.

Yazılımlar bazında tabloya göre ilk olarak, en çok soruna (132) sahip Scratch yazılımı *Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma* başlığında 72, *Sistem Durumunun Görünürlüğü* başlığında 19 önemli probleme sahip olduğu görülmüştür.

Toplam 127 sorunla ikinci sırada yer alan Small Basic yazılımı, sırayla *Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma* başlıkta toplam 43, *Yardım ve Dokümantasyon* başlığında toplam 48, *Yetenekler* başlığında 18 önemli problemi barındırdığı ortaya çıkmıştır.

Üçüncü olarak 115 sorun bulunan Kodu Game yazılımı sırayla *Sistem ve Gerçek Dünyanın Eşleşmesi ve Tutarlılık ve Standartlar başlığında 21, Kullanıcıların hataları tanımalarına, onları belirlemesine ve önlemesine yardımcı olma başlığında 35, Hataları Önleme başlığında 33 problemi bulunduğu tespit edilmiştir.*

Son olarak 59 sorun bulunan Robomind yazılımı sırayla *Hataları Önleme başlığında 39, Kullanıcı ile Seviyeli Bir İletişim başlığında 10 problem bulunmuştur.*

Tablo 22’de görülen sorunların toplam kullanılabilirlik ve tasarım sorunları Tablo 22’de görülmektedir.

Tablo 22.
Problem Sayısına Göre Görselleştirme Araçlarının Frekans ve Yüzdeleri

Sıralama	Görselleştirme Aracı	Sorun Sayısı	%
1.	Robomind	49	%11,5
2.	Kodu Game	115	%27,1
3.	Small Basic	127	%30,0
4.	Scratch	132	%31,2

Tablo 22’de görüldüğü gibi BT öğretmenlerine göre Robomind yazılımı en az probleme sahipken Scratch ise en çok problemi taşıyan yazılım olarak ortaya çıkmaktadır.

BT Öğretmenlerinin Görselleştirme Araçlarının Kullanılabilirlik ve Tasarım Problemlerine İlişkin Görüşleri

Bu başlıkta kontrol aracında bulunan kriterlerin kullanılabilirlik ve tasarım kavramlarından hangilerine karşılık geldiğine bakarak görselleştirme araçlarının yukarıdaki sorun sayılarına göre yapılan analizin kavram boyutu incelenecektir. Diğer bir deyişle, sorun çıkan ve farklılık gösteren sorunların hangi kavramla ilgili olduğu bulunup buna göre genel anlamda yazılımların hangi yönünün zayıf veya kuvvetli olduğu değerlendirilecektir. Bunun için konu alanı uzmanının vermiş olduğu destekle oluşturulan veriler Tablo 23’te görülmektedir.

Tablo 23.

Görselleştirme Araçlarının Sorunlarının Kullanılabilirlik ve Tasarım Bakımından Frekansları

Problem Barındıran Sorular	Kavramların Bulunuşu		Toplam Problem Sayısı (Hayır + Uygulanması Yok)		Small Basic		Kodu Game		Robomind	
	Kullanılabilirlik	Tasarım	Her ikisi	Scratch	Small Basic	Kodu Game	Robomind	Scratch	Small Basic	Kodu Game
1-2	✓		✓	10						
1-3				9						
2-2	✓					9	12			
2-6	✓									
3-2	✓		✓	10						
3-3				11						
4-5		✓								
4-6		✓								
5-1		✓								
5-2	✓			9						
5-3	✓			11						
5-4		✓		13						
5-5		✓		14						
5-6		✓		13						
5-7		✓		12						
6-2			✓							
6-3			✓							
6-5	✓									
7-6		✓								
7-7		✓								
8-1		✓								
9-6		✓								
10-1		✓								
10-3		✓								
10-5		✓								
10-6		✓								
11-4		✓								
11-5		✓								
12-1		✓								
12-6		✓								

Tablo 23'e ilk bakışta en çok öne çıkan soruların beşinci başlık yani *Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma* başlığında biriktiği görülmektedir. Ayrıca görselleştirme yazılımlarında var olan sorunların tasarım kavramında (13) en çok olduğu görülmektedir. Onu her ikisi (11) takip ederken sonuncu olarak kullanılabilirlik (6) kavramı gelmektedir. Bunu sorun sayılarının toplamına göre sunulmuş hali Tablo 24'te görülmektedir.

Tablo 24.
Görselleştirme Araçlarının Problem Sayılarının Kullanılabilirlik ve Tasarım Kavramlarına Göre Frekansları

	Kullanılabilirlik	Tasarım	Her ikisi
Robomind	11	10	28
Kodu Game	22	53	40
Scratch	39	50	43
Small Basic	0	67	60
Toplam	72	180	171

Tablo 24'e göre kullanılabilirlik ve tasarım kavramlarından her ikisine göre en az sorunu olan yazılım Robomind (%16,3) olurken, onu Kodu Game (%23,3) ve Scratch (%25,1) takip etmiştir. En çok sorun bu analizde de yine Small Basic (%35,0) yazılımında bulunmuştur.

Tasarım bakımından en az sorunu olan yazılım Robomind (%5,5), ikinci Kodu Game (%29,4), üçüncü Scratch (%27,7), sonuncu olarak Small Basic (%37,2) çıkmıştır.

Kullanılabilirlik bakımından hiç sorunu bulunmayan yazılım Small Basic olurken ikinci Robomind (%15,2), üçüncü Kodu Game (%30,5) ve sonuncu olarak Scratch (%54,1) çıkmıştır.

Kontrol aracı başlıklarının kullanılabilirlik ve tasarım kavramlarına göre dağılımı ise Tablo 25'te gösterilmiştir.

Tablo 25.
Kontrol Aracı Başlıklarına Göre Kullanılabilirlik ve Tasarım Kavramları Frekansları

Kontrol Aracı Başlıkları	Kullanılabilirlik	Tasarım	Her ikisi
1. Sistem Durumunun Görünürlüğü	9	-	10
2. Sistem ve Gerçek Dünyanın Eşleşmesi	12	9	-
3. Kullanıcı Kontrol ve Özgürlüğü	10	-	11
4. Tutarlılık ve Standartlar	-	31	-
5. Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma	20	93	37
6. Hataları Önleme	21	-	51
7. Hatırlama Yerine Tanıma	-	8	-
8. Esneklik ve Verimlilik	-	-	10
9. Estetik ve Sade Tasarım	-	10	-
10. Yardım ve Dokümantasyon	-	19	29
11. Yetenekler	-	-	18
12. Kullanıcı ile Seviyeli Bir İletişim	-	10	5
Toplam	72	180	171

Tablo 25'e göre görselleştirme araçlarında kullanılabilirlik boyutunda en çok sorun *Hataları Önleme* başlığında (%29,1), tasarım boyutunda *Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma* başlığında (%51,6), her iki kavram içinde *Hataları Önleme* başlığında (%29,8) olarak ortaya çıkmıştır.

SONUÇ VE TARTIŞMA

Eğitsel alanda kodlama ile ilgili çalışmalar 1960'lı yıllarda başlamıştır (Calao, Moreno-Leon, Correa & Robles, 2015). Bu kavram son yıllarda birtakım görselleştirme araçları sayesinde daha popüler hâle gelmeye başlamıştır. Çünkü bu araçlar sayesinde öğrenciler küçük yaştan başlayarak kodlamanın doğasını öğrenebilmektedir (Resnick vd., 2009). Bu araçların etkin kullanımı, bu becerileri edinecek öğrenciler açısından önem arz etmektedir. Bu bağlamda programlama öğretiminde kullanılacak görselleştirme araçlarıyla ilgili BT öğretmenlerinin kullanılabilirlik ve tasarım açısından görüşlerinin incelenmesini amaçlayan bu çalışmada, elde edilen sonuçlar aşağıda ayrı başlıklar halinde (hâlinde) özetlenmiştir.

BT Öğretmenlerinin görselleştirme araçları ile ilgili genel görüşleri

Elde edilen veriler ışığında BT öğretmenlerinin programlama öğretiminde görsel bir yazılım kullanmaya olumlu (%84,7) baktığı görülmüştür. Bu sonuç konu ile ilgili alanyazında yer alan diğer araştırmaların (Gülmez, 2009; Baldwin & Kuljis, 2001; Crews & Ziegler, 1998) sonuçlarıyla da örtüşmektedir. Bu durumda, programlama öğretiminde görsellik kullanımının ilköğretim öğrencileri için pozitif etkiye sahip olacağı düşünülebilir. Ayrıca BT öğretmenleri kullanılan görsel yazılımların genel itibarıyla (%72,8) ortaokul kademelerine daha uygun olduğunu belirtmiştir. Bununla beraber, BT öğretmenleri Small Basic yazılımının daha çok ortaokula hitap edeceğini belirtmenin yanı sıra, lise kademesinde de kullanılabileceğini de belirtmiştir (%9,7). Bunun nedeni Small Basic'in kod tabanlı yazım düzenine ve arayüze sahip olması ve bu haliyle profesyonel yazılım dillerine yakın durduğunun düşünülmesi olabilir. BT öğretmenleri Kodu Game yazılımı hariç diğerlerinin genel itibarıyla kullanılabilirlik ve tasarım yönüyle iyi olduğunu belirtmiştir.

BT öğretmenlerinin programlama öğretiminde görsel yazılım araçlarının kullanımına genel bakışlarının elde edilmesinden sonra, bu yazılımlara ilişkin görüşler sezgisel kontrol aracına göre detaylı olarak değerlendirilmiştir. Takip eden kısımda bu detaylı değerlendirmeler başlıklar hâlinde sunulacaktır.

BT Öğretmenlerinin görüşleri ışığında sezgisel kontrol aracında bulunan başlıklara bakılarak yazılımlarda tespit edilen problemler:

Kullanıcıların hataları tanınmasına, onları belirlemesine ve önlemesine yardımcı olma başlığında BT öğretmenleri toplam 150 sorunu vurgulamıştır. Bunlar arasında hata bildiriminde ses kullanımı, hata mesajlarında kullanıcının değil sistemin sorumlu olması, hata mesajlarında gramer sorunları, hatanın neden kaynaklandığı ve oluşan hatayı düzeltmek için neler yapılması gerektiği konularında görüşler dile getirilmiştir. Kısacası yazılımlarda hata yönetimi bağlamında birtakım problemler bulunmaktadır. Ayrıca oluşan bu problemlerin tasarım kavramı alanında yoğunlaştığı da görülmüştür. Scratch yazılımı bu başlıkta en çok soruna sahip yazılım olarak göze çarpmaktadır. Scratch yazılımı incelendiğinde oluşan hata mesajlarında sorunun nedeni ve bundan sonra ne yapılması gerektiğini belirten bir ifade bulunmamaktadır. Ayrıca ses kullanarak hata bildirimini de sunulmamaktadır. Gramer açısından ise Scratch'ın Türkçe sürümünde ufak tefek problemler olduğu görülmektedir. “Programdan çıkmadan önce değişiklikleri kaydetmek ister misin?” gibi cana yakın sempatik bir soruya “Kaydedin”, “Kaydetme”, ”İptal” gibi hem kendi aralarında hem de soruya göre çokta uyumlu olmayan çeviriler görülmektedir. Small Basic yazılımı da ses ile hata bildirimini vermezken hata mesajlarında teknik bir dil kullanılmaktadır ve daha sonrasında yapılması gerekenler belirtmemektedir. Kodu Game yazılımı ise oluşan hatanın sistem yerine kullanıcı kaynaklı olduğuna dair hata mesajları vermektedir. Bu hataların nasıl düzeltileceğine yönelik herhangi bir yönergede sunulmamaktadır.

Hataları Önleme başlığında (72 sorun) görselleştirme yazılımlarında veri girişi esnasında büyük/küçük harf duyarlılığının olmayışı, çalıştırılınca sonuçları itibarıyla ciddi sorunlar çıkarabilecek düğmeler/göstergelerin kolay ulaşılabilir yerlerde olması ve oluşabilecek bu olaylar öncesi kullanıcıların uyarılmaması gibi problemler bulunmuştur. Kodu Game yazılımı tasarım itibarıyla tak-çık (hot-plug) mantığına göre yapıldığı için kullanıcıdan oyun yapımı sırasında herhangi bir veri girişine imkân vermemektedir. Ayrıca yapılan oyun kurgusunun sonuçlarının ne olabileceği hakkında kullanıcıları uyarılmamaktadır, sonuçları deneme yanılma yoluyla kullanıcının bulması gerekmektedir. Robomind yazılımında ise haritadaki nesneyi (tank) hareket ettiren komut ya da komut bloklarının küçük harfle yazılması zorunludur. Bu şekilde kullanıcı sınırlanmış olmaktadır. Diğer taraftan Robomind girilen komutlar sayesinde oluşabilecek sonsuz döngüler ya da bölünebilme hatalarında programı çalıştırmadan önce sonucun ne olabileceği hakkında kullanıcıya herhangi bir uyarıda bulunmamaktadır.

Yardım ve Dökümantasyon başlığında (48 sorun) gerekli durumlarda yardım ve belgeleme olması beklenen yazılımlardan Small Basic'in bu ihtiyacı karşılamaktan uzak olduğu görülmüştür. Small Basic yazılımı incelendiğinde sorularda da beklendiği gibi bir yardım menüsü ya da aynı işleve gören bir düğme bulunmamaktadır. Ancak ekranın sağ tarafında yer alan alanda kod alanında yazılan

komutlara göre o komutun anlamı ve nasıl kullanıldığına dair minik örneklerin sunulduğu bir yardım metodu bulunmaktadır. Sunulan bu yardım işlevinde detay seviyeleri de bulunmamaktadır. Diğer yazılımlarda ise kullanıcıya yön gösterebilecek bir yardım menüsü var olup farklı seviyelerde bilgiler mevcuttur.

Tutarlılık ve Standartlar başlığında (31 sorun) Kodu Game ve Small Basic yazılımlarının sistem boyunca ortak bir tutarlılığın geçerli olmadığı görülmüştür. Kodu Game yazılımında açılış ekranında yer alan menü dikey iken programlama işlemi esnasında yatay bir menü tasarımı kullanıcıya sunulmaktadır. Yine “çıkış” seçeneği ise yatay menüde en solda ya da “ESC” tuşu ile sağlanmaktadır. Robomind yazılımı ise klasik dikey menü yerine menüyü sekme tarzı simgeli gruplar hâlinde kullanıcıya vermektedir.

Kullanıcı Kontrol ve Özgürlüğü başlığında (21 sorun) yazılımlarda kullanıcıların hata yaptıklarında veya hatayı geri alma işlemlerinde kullanıcıya tam bir özgürlük sunulmadığı görülmüştür. Buna göre Scratch yazılımı incelendiğinde programlama yapılırken yapılan işlemleri geri alma ve yineleme işlemlerinin neredeyse olmadığı, belli başlı temel işlemler (kaydetme, silme gibi) için kullanılabilirdiği görülmüştür. Blok tabanlı bir görselleştirme aracında geri alma ve yinelemenin çok kısıtlı olması kullanıcı bu bakımdan yormaktadır. Ayrıca Scratch'ta kullanıcılar “Motor Blokları Göster/Gizle” gibi nadir kullanılan görevleri hatırlamakta güçlük çekmektedir.

Sistem ve Gerçek Dünyanın Eşleşmesi başlığında (21 sorun) Kodu Game yazılımında sistem içerisinde kullanılan kelimeler, kavramlar, cümleler kullanıcıya fazla tanıdık gelmediği sonucuna varılmıştır. Buna göre Kodu Game'in Türkçe arayüzünde doğrudan bir çeviri yapıp “Yeni Dünya”, “Dünya Yükle”, “Benim Dünyalarım” gibi ilk bakışta garip gelebilecek bir bilgisayar üslubu kullanıcıyı karşılamaktadır. Ancak yazılımı kullanarak neler yapıldığını/üretildiğini ve nasıl bir mekânda geçtiğini görünce “Dünya” çevirisinin anlamlı olduğu ortaya çıkmaktadır. Diğer taraftan haritadaki nesnelere programlarken verilen komutların yerleştirildiği paletlere “Kare” gibi bir çeviri verilmiştir. Yazılımda bu ve bunun gibi gerçek dünya ile tam olarak uyuşmayan örnekler tespit edilmiştir.

Sistem Durumunun Görünürlüğü başlığında (19 sorun) ise yazılımların kullanımı esnasında o anki mevcut durum hakkında bilgilendirme, hatırlatma ve hata mesajlarının aynı yerde fark edilebilir bir şekilde gözükmemesi, nesne seçimini veya hareket ettirildiğini gösteren görsel bir bildirim olmayışı, grafik arayüzlü menülerde hangi seçeneğin seçildiğinin belli olmaması gibi problemler tespit edilmiştir. Scratch yazılımı çok sık hata vermemekle birlikte oluşan durumun hata olup olmadığı da anlaşılabilir değildir. Hata mesajı alanı sabit bir yerde çıkmayıp ekranın sol üst köşesinden başlayarak merdiven şeklinde her hata için bir alta inmektedir. Small Basic yazılımında ise menü kavramı bulunmadığı için herhangi bir seçim ve bu seçimi belli eden bir geri bildirim bulunmamaktadır.

Yetenekler başlığında (18 sorun), programlama öğretiminde kullanıcıların Small Basic ve Scratch yazılımlarının yetenekleri artırma ve uzmanlık kazandırma noktasında tam olarak istenileni karşılayamadığını göstermektedir.

Kullanıcı ile Seviyeli Bir İletişim başlığında (15 sorun) Scratch ve Kodu Game yazılımlarının herhangi bir veri giriş alanında otomatik tamamlama (auto-complete) özelliğini sağlayamadığı görülmektedir.

Estetik ve Sade Tasarım başlığında (10 sorun) Small Basic yazılım incelendiğinde dışarıdan veri girişi yapabilmek için kutu dışında farklı veri girişi elemanları bulunmadığı görülmektedir. Aynı şekilde kod tabanlı bir arayüze sahip olan Robomind yazılımında ise istenen veri giriş nesnelere bulunmaktadır.

Esneklik ve Verimlilik başlığında (10 sorun) Scratch ve Small Basic yazılımları incelendiğinde deneyimli ve deneyimsiz kullanıcılara göre hata oluştuğunda onların seviyesine göre hata mesajları oluşturmadığı görülmektedir. Özellikle Scratch bu bağlamda kimi durumlarda herhangi bir hata mesajı dahi vermemektedir. Kullanıcı kendi çabaları ve yeteneğine göre sorunu çözme yoluna gitmektedir.

Hatırlama Yerine Tanıma başlığında (8 sorun) Small Basic yazılımı için hatırlama yerine tanıma veya anımsamayı sağlayacak anlamlı grupların kod tabanlı bir yazılım arayüzünde tam olarak sağlanamadığı görülmektedir. Zira bu özellik menü alanında kısmen sunulsa da asıl işin yapıldığı kod alanında bulunmamaktadır.

BT Öğretmenlerinin görüşleri ışığında görselleştirme araçları bazında öne çıkan sorunlar

BT öğretmenlerinin toplam 132 sorun hakkında yorum yaptığı *Scratch* yazılımı, mevcut durum hakkında bilgilendirme ve hata mesajlarının aynı yerde çıkmaması, veri girişlerinin otomatik olarak tamamlanmaması, birçok işlem için geri al (*undo*) işlevinin olmayışı, az kullanılan görevlerin hatırlanamaması, sesli hata bildirim eksikliği, hatanın nedeni ve ne yapılması gerektiği konusunda yönlendirilmemesi, gramer olarak hatalı mesajların çıkması, kullanıcıya kullanımdan kaynaklanan bir uzmanlık katmaması gibi konularda önemli eksikliklere sahiptir.

BT öğretmenlerinin 127 sorun bildirdiği *Small Basic* yazılımında ise hata yönetimi dediğimiz; hatanın sesle bildirim, hatanın sorumlusunun sistem olduğunun vurgulanması, hata mesajlarında gramer olarak doğruluk, oluşan hatanın nedeni ve bundan sonra ne yapılması gerektiği gibi hususlarda ciddi eksiklikler olduğu fark edilmiştir. Bunun yanı sıra acemi ve uzman kullanıcılara karşı hata mesajlarının seviyelerinin uygun olmayışı, yine bu kullanıcılar için uygun bir yardım dokümantasyonunun bulunmaması, uzun süreli kullanım sonrasında kullanıcıya yetenek ve uzmanlık kazandıracak kısayollar ve işlevlerden mahrum oluşu, veri girişi esnasında kolaylık sağlayacak radyo

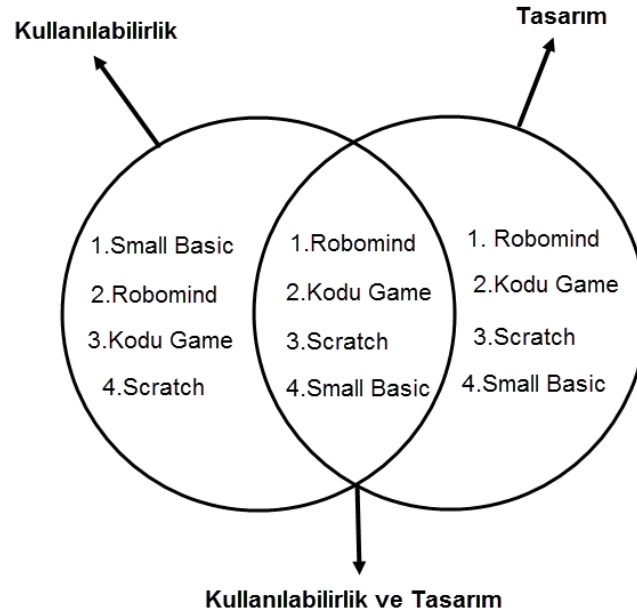
düğmeleri, açılır kutular gibi giriş nesnelерinin bulunmaması bir diğер önemli eksiklikler olarak karşımıza çıkmıştır.

Kodu Game yazılımında (115 sorun) ise, gerçek dünya ile uyuşmayan kültürel birtakım görsel unsurların oluşu, yine daha önce bahsedilen hata yönetimi ve bildirim problemleri, veri girişlerinin otomatik tamamlanmaması, dikey bir menünün olmayışı ve “çıkış” seçeneğinin bu menünün en altında bulunmaması gibi sorunlar bulunmuştur.

En az soruna (49) sahip *Robomind* yazılımında ise, nesne hareketi için girilen komutların sadece küçük harfle girilmesi sınırlaması, nadir de olsa meydana gelebilecek ve içinden çıkılmayacak hata durumları, menülerin çoğunlukla dikey değil de yatay sunulması, oluşan hatalarda her iki kullanıcı tipi için mesaj seviyelerinin bulunmaması gibi problemler göze çarpmıştır.

BT Öğretmenlerinin görüşleri ışığında görselleştirme araçlarının barındırdığı sorunların kullanılabilirlik ve tasarım boyutları açısından incelenmesi

Kullanılabilirlik ve tasarım kavramları bakımından BT öğretmenlerinden toplanan verilere göre yazılımlarda bir sıralama yapmak gerekirse, Şekil 1’deki gibi bir yapı elde edilebilir.



Şekil 1. Kullanılabilirlik ve Tasarım Boyutlarına Göre Görselleştirme Araçlarının Durumları

Şekil 1’e göre Small Basic yazılımı her iki kavramda en çok soruna sahip yazılım olarak karşımıza çıkmaktadır. En çok sorun dile getirilen Scratch yazılımı kullanılabilirlik ve tasarım kavramları bakımından üçüncü sırada yer almıştır. Kodu Game yazılımı ise bu alanda ikinci sırada yer almıştır. Ancak Kodu Game yazılımının kullanılabilirlik ve tasarımının diğerlerinden (3 ve 4 puan toplamına göre) geride çıkmıştır. Bunun nedeni oyun üretim odaklı bir yazılım olması nedeniyle, oyun

yaparken birçok kontrolün olması ve çok işlevsel bir arayüze sahip olması olabilir. Bu da kullanıcılar tarafında aracın kullanılabilirlik ve tasarım algısını etkilemiş olabilir. Hem kullanılabilirlik hem de tasarım açısından istenen özelliklere sahip yazılım ise bu başlıkta birinci olarak Robomind yazılımı olarak karşımıza çıkmaktadır.

Eğitmcilerin, programlama öğretiminde görselleştirme araçlarını kullanırken, konuya uygun araç seçiminin önemli olduğunu göz önünde bulundurmaları ve dersten önce uygun aracın belirlenmesi ve derste hangi örnekler üzerinde durulacağı ile ilgili ön çalışma yapmaları gerekmektedir. Bravo, Marcelino, Gomes, Esteves ve Mendes (2005), algoritmayı bir karakter yardımıyla oluşturmaya yardımcı olan araçların, öğrencilerin programları anlamaları ve analiz etmeleri konusunda yardımcı olabileceğini belirtmişlerdir. Bu durumda ilköğretim öğrencilerine algoritma geliştirme ile ilgili eğitim verilirken görselliğin daha fazla ön planda olduğu bir yardımcı araç kullanımının öğrencilerin konuyu öğrenmelerini kolaylaştırdığı söylenebilir.

Programlama öğretiminde yaygın olarak kullanılan Small Basic, Scratch, Kodu Game ve Robomind yazılımlarının, araştırma amaçlarına uygun olarak adapte edilen sezgisel kontrol aracılığıyla, BT öğretmenlerinin görüşleri ışığında kullanılabilirlik ve tasarım yönlerinin incelendiği bu araştırmada:

- Small Basic yazılımının hata yönetimi, yardım dokümantasyonu, hata mesajları, kullanıcıya göre esnek kullanım gibi konularda ciddi iyileştirmelere ihtiyaç duyduğu,
- Scratch yazılımında sık kullanılan işlevler için kısayollar, hatırlanabilme, hata yönetimi, kullanımda verimlilik gibi konularda düzeltmelere ihtiyaç olduğu,
- Microsoft Kodu Game yazılımında gerçek dünya ile simge uyumu, veri girişleri, menü tasarımı gibi görsel alana ilişkin problemlerinin çözülmesi gerektiği ve
- Robomind yazılımında ise, kod girişinde küçük/büyük harf duyarlılığı, menü tasarımı, hata mesajlarında seviye durumları gibi konularda düzeltmeler yapılması gerektiği sonuçlarına ulaşılmıştır.

KAYNAKLAR

- Akçay, T. (2009). *Perceptions of students and teachers about the use of A kid's programming language in computer courses*. Yayınlanmamış doktora tezi, Ortadoğu Teknik Üniversitesi, Ankara.
- Arabacıoğlu, T., Bülbül, H. İ. ve Filiz, A. (2007). *Bilgisayar programlama öğretiminde yeni bir yaklaşım*. IX. Akademik Bilişim Konferansı Bildirileri Kitabı. Kütahya.

- Baldwin, L. P., & Kuljis, J. (2001, January). *Learning programming using program visualization techniques*. In Proceedings of the 34th Annual Hawaii International Conference on System Sciences (pp. 8-pp). IEEE.
- Bishop-Clark, C., Courte, J., Evans, D., & Howard, E. V. (2007). A quantitative and qualitative investigation of using Alice programming to improve confidence, enjoyment and achievement among non-majors. *Journal of Educational Computing Research*, 37(2), 193-207.
- Bravo, C., Marcelino, M. J., Gomes, A. J., Esteves, M., & Mendes, A. J. (2005). Integrating Educational Tools for Collaborative Computer Programming Learning. *Journal of Universal Computer Science*, 11(9), 1505-1517.
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. In *Design for teaching and learning in a networked world* (pp. 17-27). Springer, Cham.
- Calder, N. (2010). Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- Crews, T., & Ziegler, U. (1998). *The flowchart interpreter for introductory programming courses*. In 28th Annual Frontiers in Education Conference. Moving from 'Teacher-Centered' to 'Learner-Centered' Education. Conference Proceedings (Cat. No. 98CH36214) (Vol. 1, pp. 307-312). IEEE.
- Drucker, P. F. (2000). *Yeni Gerçekler (Çev. Birtane Karanakçı)*. İstanbul: Türkiye İş Bankası Yayınları.
- Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE*. 41(3), 258-262.
- Gülmez, I. (2009). Programlama Öğretiminde Görselleştirme Araçlarının Kullanımının Öğrenci Başarı ve Motivasyonuna Etkisi. *Yayımlanmamış Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul*.
- Gültekin, K. (2006). *Çoklu Ortamın Bilgisayar Programlama Başarısı Üzerine Etkisi*. Yayımlanmamış Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Karasar, N. (2013). *Bilimsel araştırma yöntemi*. Ankara: Nobel Yayıncılık.
- Kaucic, B., & Asic, T. (2011). Improving Introductory Programming with Scratch. *Proceedings of the 34th International Convention* (s. 1095 - 1100). MIPRO.

- Kelleher, C., Pausch, R., Pausch, R., & 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.
- Lahtinen, E., Ahoniemi, T., & Salo, A. (2007, November). *Effectiveness of integrating program visualizations to a programming course*. In Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88 (pp. 195-198). Australian Computer Society, Inc.
- Lin, C., & Zhang, M. (2003). *The use of computer animation in teaching discrete structures course*. The 36th Annual Midwest Instruction and Computing Symposium. MICS.
- Nielsen, J. (2015). 10 Usability Heuristics for User Interface Design. Web: <http://www.nngroup.com/articles/ten-usability-heuristics/> den alınmıştır.
- Prawalpatagool, J. (2010). *An Effective Technique for Learning in The Computer Programming Subject*. Yayımlanmamış Yüksek Lisans Tezi, Prince of Songkla University., Management of Information Technology).
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67. [Çevrim-içi: <https://cacm.acm.org/magazines/2009/11/48421-scratch-programming-forall/fulltext>, ve <https://cacm.acm.org/magazines/2009/11/48421-scratch-programming-for-all/pdf>, Erişim tarihi: 09.01.2019.]
- Sleeman, D., Putnam, R. T., Baxter, J., & Kuspa, L. (1984). Pascal and High-School Students: A Study of Misconceptions. Technology Panel Study of Stanford and the Schools. Occasional Report #009
- Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). A model curriculum for K-12 computer science. *Final Report of the ACM K-12 Task Force Curriculum Committee, CSTA*. Retrieved from <http://dl.acm.org/citation.cfm?id=2593247>

EXTENDED ABSTRACT

The problem of how to give programming instruction to children has recently emerged with the introduction of programming teaching at primary school level. In schools, programming teaching is generally given using theoretical methods. It is known that programming instruction theoretically may not engage students; neither is it very effective. To overcome this problem, there are several

teaching software and visualization tools that have been used for many years. Some countries have developed some strategies for teaching programming by using this software within the curriculum. However, when using these software and visualization tools to certain ages and classes, it is possible to face some problems in terms of the usability and design of the software, given the current mental perceptions of the students. A programming tool that is poor in design, does not address a particular age or class, or has a low-level of usability, cannot be beneficial for the students, and there is a possibility that it may lead to complexity and incorrect learning. Therefore, it is necessary to take the opinions of expert teachers in selecting the software to be used in teaching.

In this context the purpose of this study is to examine the opinions of IT teachers about the usability and design aspects of some software that can be used in programming teaching. In this study, Scratch 1.4, Microsoft Small Basic 1.2, Microsoft Kodu Game Lab 1.4.64 and Robomind 6.01 software were examined. 92 IT teachers working in secondary schools in Bursa participated in the study. The interface design and usability analysis of the mentioned software were based on Nielsen's (2015) heuristic guide. The original version of this guideline was arranged in line with the research objectives and sent to teachers in an electronic form.

The research was carried out based on the survey model of quantitative research design. The frequency, average and percentages of the data were calculated and the opinions of the participants regarding the research subject were examined.

In light of the data obtained in the research, it was found that using visual tools as software in programming instruction could have a positive effect on students' learning. IT teachers stated that these tools are generally more suitable for secondary schools. Also when the problems detected in the light of the opinions of IT Teachers in the visualization software considering the topics in the control tool, the two most common problems are as follows: "Helping users to recognize, identifying and preventing errors" and "Preventing Errors". Regarding the first, ICT teachers emphasized 150 problems, while for the second, they highlighted 72 problems. In addition, Small Basic 1.2 software can also be used at high school level because it is closer to professional languages due to the code-based structure it contains. On the other hand, according to the ICT teachers, it was concluded that both the usability and design aspects should be considered when developing the software. These four visualization tools generally have good usability aspects and designs, among which the Robomind 6.01 software is the most readily available and has the least problems in terms of design.