

Programlama Dili Öğretiminde Alice Programının Kullanım Sürecinin İncelenmesi

Utilization of Alice Software in Teaching Programming Language

Ceren Baştemur Kaya*
Hasan Çakır**

To cite this article/Atf için:

Baştemur Kaya, C., ve Çakır, H. (2018). Programlama dili öğreniminde alice programının kullanım sürecinin incelenmesi. *Eğitimde Nitel Araştırmalar Dergisi- Journal of Qualitative Research in Education*, 6(2), 187-206. DOI:10.14689/issn.2148 - 2624.1.6c2s9m

Makale Hakkında

Gönderim Tarihi: 05. 01. 2018

Düzeltilme Tarihi: 19. 07. 2018

Kabul Tarihi: 23. 07. 2018

Öz. Programlama öğrenimi zor ve karmaşık bir süreçtir. Programlama öğreniminin karmaşıklığını azaltmak amacıyla görselleştirme araçlarının kullanımının arttığı görülmektedir. Bu görselleştirme araçlarından biri de Alice programıdır. Çalışma kapsamında Java programlama dili öğreniminde Alice programının kullanım süreci incelenmiştir. Çalışma grubu, meslek yüksekokulu bilgisayar teknolojileri bölümü 2. sınıfta öğrenim gören 20 öğrenciden oluşmaktadır. Araştırma haftada dört saat olmak üzere beş hafta sürmüştür ve toplamda 20 saat uygulanmıştır. Gagne'nin dokuz aşamalı öğretim modeli kullanılarak dersler yürütülmüştür. Öğrencilerden iki haftada bir değerlendirme notu toplanmıştır. Çalışmada tanımlayıcı durum çalışması kullanılmış ve veriler içerik analizi yöntemi ile analiz edilmiştir. Sonuçlar incelendiğinde öğrencilerin; Alice programının donanım gereksinimlerinin yüksek olması nedeniyle bilgisayarda kilitlenmelerin yaşandığını ancak Alice programının kod mantığını öğrenmeye yardımcı olduğunu, programlamaya olan ilgiyi artırdığını ve programlamayı eğlenceli hale getirdiğini belirttikleri görülmektedir. Programlama mantığının daha iyi anlaşılması için Alice programının kullanılmasının faydalı olacağı düşünülmektedir.

Anahtar Sözcükler: Alice programı, meslek yüksekokulu, programlama ortamları, programlama öğrenimi

Abstract. Learning a programming language is a difficult and complicated process. It has been seen that the use of visualization tools has increased to decrease the complexity of learning programming. One of the visualization tools used for this purpose is Alice software. This study investigates the use of Alice software in learning Java programming language. The study group consists of 20 students who study at the 2nd year of vocational schools, department of computer technologies. The study was carried out five weeks to be four hours in a week and it was applied for 20 hours in total. The courses were realized using Gagne's nine events of instruction model. Reflections were collected from the students in every other week. Descriptive case study method was utilized in the study and the data were analyzed using content analysis method. When the results were examined, it is seen that the students have stated that the computers freeze due to the high hardware requirements of the Alice program. Despite that, It is observed that Alice program helped learning the coding logic, increased interest in programming and made programming fun. It is thought that use of Alice software can be beneficial for better understanding of programming logic.

Keywords: Alice software, vocational schools, programming environments, programming learning

* *Sorumlu Yazar / Correspondence:* Ceren BAŞTEMUR KAYA, Nevşehir Hacı Bektaş Veli Üniversitesi, Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, e-mail: ceren@nevsehir.edu.tr ORCID:0000-0002-0091-3606

**Gazi Üniversitesi, Gazi Eğitim Fakültesi, e-mail: hasanc@gazi.edu.tr ORCID:0000-0002-4499-9712

Giriş

Teknolojinin gelişmesiyle birlikte bireylerden beklenen özellikler değişmekte, yaşam tarzı ve toplum yapısında dönüşümler meydana gelmektedir. Var olan bilgi ve becerilerin işlenip yeni durumlara göre şekillendirilebilmesi ve farklı ürünlerin ortaya çıkarılabilmesi, günümüz bilgi çağında büyük önem kazanmıştır (Partnership for 21st Century Skill, 2009). Bireylerin yeni ve özgün ürünler geliştirebilmesi, özellikle bilgi ve iletişim teknolojisi sektörünü ön plana çıkarmaktadır. Bilişim sektöründe farklı teknolojiler üreterek, yeni ürünler geliştirerek var olunabilmektedir. Bu durum programlamanın önemini ortaya çıkarmaktadır.

National Research Council tarafından yayınlanan "Bilgi Teknolojileri ile Anlaşılır Olma" isimli raporda programlama yeteneğine sahip olunması gerektiği vurgulanmıştır. Programlama yeteneğinin, bilişimsel düşünmeyi desteklediği belirtilmiştir. Özellikle dijital teknolojinin gelişiminin tüketici kavramını, yaratıcı kavramına dönüştürdüğü ifade edilmiştir (National Research Council, Committee on Information Technology Literacy, 1999; Resnick, Flanagan, vd., 2009).

Programlamanın; yeni ürünleri ve özgün teknolojileri geliştirmedeki etkisinin büyük olmasının yanında programlama bireylerin farklı yönlerden gelişimi için de önemli bir yere sahiptir. Örneğin programlama ile uğraşan bireylerin yaratıcılıkları, yansıtıcılıkları, üstbiliş yetenekleri (Clements & Gullo, 1984), kural öğrenmeleri (Gorman & Bourne, 1983), problem çözme ve analiz etme yetileri (Feurzeig, Papert & Lawler, 2011; Papert, 1993; Saeli, Perrenet, Jochems & Zwaneveld, 2011) gibi becerileri; programlama ile uğraşmayan bireylerden daha iyi yönde geliştiği yapılan çalışmalarda belirlenmiştir. Programlama öğrenimi, günümüz toplumundaki önemli sorunların çözümü ve yeni projelerin tasarlanması için önemli stratejiler geliştirmeyi de desteklemektedir (Resnick, 2013). Ancak programlama öğrenmek oldukça zordur (Powers, Ecott & Hirshfield, 2007; Resnick, Flanagan, vd., 2009; Robins, Rountree & Rountree, 2003).

Programlama karmaşık katmanlardan oluşmaktadır. Katmanlardan biri programlama dillerinin sözdizimi, diğer bir katman problemlerin çözümü için gerekli işlem adımlarını oluşturmayı sağlayan programlama mantığıdır (Biju, 2013). Sözdizimleri bir araya gelerek kod bloklarını ve programları oluşturur. Programlama sürecinde sözdiziminden kaynaklı hataların bulunması ayrı bir iş yüküdür. Çoğu zaman tespit edilemeyen veya geç tespit edilen hatalar programlama motivasyonunu düşürmekte (Biju, 2013) ve programlamaya karşı sıkıcı bir tutumun oluşmasını sağlamaktadır (Howard, Evans, Courte & Bishop-Clark, 2006). Özellikle programlamaya yeni başlayanlar programlama dilinin sözdizimini ve programlama mantığını kolay bir şekilde öğrenmek için yazılım araçlarına ihtiyaç duymaktadır (Biju, 2013; Bishop-Clark, Courte, Evans & Howard, 2007). Bunun yanında programlama dillerinin temel yapıları birbirine benzerdir. Temel kavramların bilinmesi farklı dillerin öğrenilmesi için kolaylık sağlamaktadır. Ayrıca programlamada geçmiş bilgi ve becerilerle yeni bilgilerin inşa edilmesi çok önemlidir (Gomes & Mendes, 2007). Ancak her programlama öğreniminin başında temel kavramların iyi bir şekilde anlaşılabilmesi nedeniyle tekrar temel kavramların öğrenilmesine yoğunlaşılması, yeni uygulamalar yapılmasını kısıtlamakta ve programlamada ilerlemeyi engellemektedir. Programlama dili öğreniminin zor olması (Resnick, Flanagan, vd., 2009), programlamadaki soyut kavramların ve programlama mantığının anlaşılmasının güç olması (Gomes & Mendes, 2007) programlama eğitiminde farklı öğretim ortamlarının kullanımını arttırmıştır.

Türkiye’de farklı bölümlerde programlama eğitimi verilmektedir. Özellikle programlama derslerinin yoğun bir şekilde verildiği bölümlerden biri, meslek yüksekokulu bilgisayar teknolojileri bölümüdür. Ancak bu derslerde düz anlatım, soru-cevap, gösterip yaptırma vb. geleneksel öğretim yöntemleri kullanılmaktadır (Binici & Necdet, 2004; Şahin & Fıfık, 2008; Ünal & Çakır, 2016). Bu durum

programlama öğrenmede yaşanan zorluklar düşünüldüğünde, öğrencilerin programlama mantığını anlamalarını zorlaştırmaktadır (Cooper, Dann & Pausch, 2000; Kölling, 2010; Kölling, Quig, Patterson & Rosenberg, 2003; Resnick, 1990; Resnick, Maloney, vd., 2009). Programlamanın zor olmasından dolayı, daha kolay bir şekilde öğrenilmesi amacıyla öğrenme ortamlarında görsel öğrenme araçlarına olan ihtiyaç artmaktadır (Howard vd., 2006; Powers vd., 2007).

Programlama eğitiminde yaşanan bu sıkıntılardan ötürü, programlama öğrenmenin zor süreçlerini kolaylaştırmak amacıyla görsel programlama araçları ve ortamları tasarlanmıştır (Cooper vd., 2000; Kölling, 2010; Kölling vd., 2003; Resnick, 1990; Resnick, Maloney, vd., 2009). Programlama öğretmek için tasarlanan bu ortamlar; ilgi çekici, öğrenmesi kolay ve öğrencilerin kısa sürede yeni uygulamaları ortaya çıkararak öğrenmelerine olanak sağlamaktadır (Bishop-Clark vd., 2007). Bu amaç için tasarlanan yazılım araçlarından biri Alice programıdır.

Görsel Bir Programlama Ortamı : Alice

Alice programı etkileşimli arayüze sahip üç boyutlu bir programlama ortamıdır. Kullanıcılar Alice programı ile bir program oluşturmak için ilgili kodları sürükleyip bırak mantığına göre çalışma alanına ekleyebilmektedir. Böylece sözdiziminden kaynaklanan hatalarla uğraşmak zorunda kalmazlar. Alice programındaki talimatlar nesne tabanlı programlama dilindeki standart kodlamalara karşılık gelmektedir (Hayat, Al-Shukaili & Sultan, 2017). Kullanım kolaylığı, etkileşimli ve görsel olması gibi avantajlarından dolayı literatürde farklı çalışmalarda kullanılmıştır.

Daly (2011) Alice programının Java programlama dili öğrenimindeki etkisini belirlemek amacıyla deneysel bir çalışma yapmıştır. Çalışmada deney grubu altı hafta boyunca ilk başta Alice programı ile çalışıp sonrasında Java programlama diline geçmiş, kontrol grubu süreç boyunca Java programlama dilinde geleneksel yöntemlerle çalışmıştır. Deneysel sürecin başında, ortasında ve sonunda katılımcıların programlama kavramlarını öğrendikçe güven seviyelerindeki artışı belirlemek amacıyla anketler uygulanmıştır. Çalışma sonucunda Alice programı ile çalışan öğrencilerin programlamaya güven seviyelerinin ve temel programlama kavramları öğrenimlerinin daha yüksek olduğu tespit edilmiştir.

Liu, Lin, Hasson ve Barnett (2011) çalışmalarında öğretmenlerin bilgisayar bilgilerini geliştirmek amacıyla Alice programı kullanarak bir eğitim tasarlamışlardır. Araştırmanın başında ve sonunda katılımcılara anket uygulanmış ve görüşme yapılmıştır. Çalışma sonucunda Alice programı kullanılarak bilgisayar bilimleri öğretiminde güven seviyesi ve bilgi birikim artışı olduğu tespit edilmiştir.

Aktunc (2013) çalışmasında Java programlama diline Alice programı kullanılarak geçişin sağlanması amacıyla bir yaklaşım sunmuştur. Sadece Alice programı kullanarak tüm öğrenme hedeflerine ulaşamayacağını ancak temel programlama dili kavramlarının daha iyi bir şekilde öğrenilmesi için Alice programının kullanılmasının yararlı olacağını vurgulamıştır. Alice programı ve Java programlama dilinin birlikte öğrenilmesinin sağlam bir programlama temeli vereceğini ve üst düzey programlamaya hazırlayacağını ifade etmiştir.

Price (2013) Alice programının akademik başarı ve kalıcılık üzerindeki etkisini belirlemek amacıyla deneysel bir çalışma yapmıştır. Deney grubunda Alice programı ile öğretim tasarlanmış, kontrol grubunda geleneksel öğretim yöntemleri kullanılmıştır. Veriler akademik başarı sınavı ve ara sınavlar aracılığıyla toplanmıştır. Çalışma sonucunda Alice programı ile çalışan grupta özellikle nesne ve sınıf kavramları öğreniminde anlamlı bir artışın olduğu belirlenmiştir.

Zhang, Liu, de Pablos ve She (2014) Alice programının akademik başarı, motivasyon ve ilgi düzeyine etkisi belirlemek amacıyla bir çalışma yapmışlardır. Deneysel olarak yapılan çalışmada deney grubu Alice programı ile kontrol grubu geleneksel yöntemler ile Java programlama dili öğrenimi görmüşlerdir. Araştırma sonucunda Alice programının akademik başarıya, motivasyona ve ilgi düzeyine anlamlı olarak olumlu yönde etki ettiği belirlenmiştir.

Kayabaşı (2016) Alice programının öğretmen adaylarının öz yeterlilik algılarına olan etkisini ve öğretmen adaylarının Alice programına ilişkin görüşlerinin belirlenmesini amaçlayan bir çalışma yapmıştır. Veriler görüşme formu ve anket yardımıyla toplanmıştır. Çalışma sonucunda katılımcılar Alice programı ile derse olan katılım ve ilgilinin artabileceği; Alice programının kalıcılığı artırıp; liderlik, yaratıcı düşünme ve problem çözme becerilerini geliştirebileceği yönünde görüş belirtmişlerdir. Ayrıca Alice programının öz yeterlilik algısına anlamlı olarak olumlu yönde etki ettiği ifade edilmiştir.

Hayat vd. (2017) Alice programını K-12 öğrencilerine öğretip etkisini incelemek amacıyla bir çalışma yapmışlardır. Alice programı öğretildikten sonra öğrencilere proje verilmiş ve anket yardımıyla görüşleri alınmıştır. Çalışma sonucunda öğrencilerin Alice programının kullanımının kolay olduğunu, Alice programı ile çalışmanın eğlenceli ve motivasyon artırıcı olarak gördüklerini belirttikleri ifade edilmiştir.

Alice programı ile ilgili yapılan çalışmalar incelendiğinde ortaokul öğrencileri (Hayat vd., 2017; Werner, Denner, Bliesner & Rex, 2009), lise öğrencileri (Wang, Mei, Lin, Chiu & Lin, 2009), üniversite öğrencileri (Cliburn, 2008; Cooper, Dann & Pausch, 2003), öğretmenler (Liu vd., 2011) ile programlamaya yeni başlayan ana bilim dalı bilgisayar olmayan kullanıcılar (Howard vd., 2006) gibi farklı yaş aralığı ve farklı statülerdeki çalışma gruplarında kullanımının olduğu belirlenmiştir. Alice programının etkileri olarak programlamaya yeni başlayanlarda özellikle görselleştirme araçlarının kullanımının; programlamanın temel kavramlarının anlaşılmasını kolaylaştırdığı (Cliburn, 2008; Cooper vd., 2003; Hayat vd., 2017; Howard vd., 2006; Wang vd., 2009; Werner vd., 2009), programlama yapmayı kolaylaştırdığı (Howard vd., 2006; Werner vd., 2009), programlama öğrenmede motivasyonu arttırdığı (Hayat vd., 2017; Werner vd., 2009), eğlenceli ve kullanımının kolay olduğu (Cliburn, 2008; Hayat vd., 2017; Howard vd., 2006; Werner vd., 2009) vurgulanmıştır.

Görsel öğrenme araçlarından biri olan Alice programının, özellikle programlama dili öğrenimini kolaylaştırmak ve temel programlama bilgisinin daha kolay bir şekilde öğrenilmesini sağlamak amacıyla kullanıldığı belirlenmiştir (Bishop-Clark vd., 2007; Cliburn, 2008; Hayat vd., 2017; Howard vd., 2006; Liu vd., 2011; Powers vd., 2007; Wang vd., 2009; Werner vd., 2009). Ancak Alice programının daha etkili, verimli ve ilgi çekici bir öğrenim için, öğrenme ortamlarında nasıl kullanılması gerektiği ile ilgili yapılan çalışmalar sınırlıdır. Özellikle öğrenim sürecinde yardımcı programların nasıl ve ne zaman kullanılması gerektiğini belirleyen en önemli veriler öğrenci görüşleridir.

Bu çalışmanın amacı meslek yüksekokulu bilgisayar teknolojileri bölümünde öğrenim gören 2. sınıf öğrencilerinin nesne tabanlı programlama dillerinden biri olan Java programlama dilinin öğretiminde görsel, etkileşimli ve üç boyutlu yazılım araçlarından biri olan Alice programının öğrenme ortamlarında daha etkili, verimli ve ilgi çekici bir şekilde kullanımını sağlamak amacıyla öğrenci görüşlerini incelemektir. Bu amaç çerçevesinde çalışmada " Öğrencilerin programlama öğreniminde Alice programının kullanımına ilişkin görüşleri nelerdir?" sorusuna cevap aranmaktadır.

Yöntem

Araştırma Deseni

Araştırmada nitel araştırma yöntemlerinden tanımlayıcı durum çalışması kullanılmıştır. Durum çalışmasında bir ya da birden fazla olgu, program, ortam, sosyal grup veya sistem gerçek yaşam bağlamı içerisinde çok yönlü bir şekilde derinlemesine incelenmektedir (Cohen & Manion, 2000; McMillan, 1996). İnceleme sonucunda durumlar ve duruma bağlı temalar ortaya çıkarılmaktadır (Creswell & Poth, 2017). Tanımlayıcı durum çalışmalarında amaç durumun ayrıntılı, derinlemesine ve gerçekçi bir şekilde ortaya konulmasıdır (Merriam, 2013). Bu çalışmadaki amaç programlama dili öğretiminde, Alice programının öğrenme ortamlarında etkili, verimli ve ilgili çekici kullanılmasını sağlamak amacıyla öğrenci görüşlerini derinlemesine ve gerçekçi bir şekilde incelemek olduğu için tanımlayıcı durum çalışması tercih edilmiştir.

Çalışmanın Bağlamı

Bilgisayar teknolojileri bölümünde, özel ve kamu sektörünün bilişim alanında ihtiyaçlarını karşılayacak ara elemanlar yetiştirilmektedir. Öğrenciler teknolojiyi farklı ortamlarda kullanabilecekleri, programlama altyapısına sahip olacakları, problem çözme yeteneklerini geliştirebilecekleri ayrıca gelişen teknolojiyi kolaylıkla izleyebilecekleri içeriğe sahip dersler almaktadır (Baştemur Kaya & Çakır, 2015). Bilgisayar teknolojileri bölümü; bilgisayar sistemlerinin yazılımsal ve donanımsal yapısının öğrenilmesi ve geliştirilmesine yönelik eğitim vermektedir. İki yıllık eğitim süresi sonucunda, başarılı bir şekilde mezun olan öğrenciler tekniker unvanını almaktadır.

2011 Öğrenci Seçme ve Yerleştirme Sistemi (ÖSYS) Yükseköğretim Programları ve Kontenjanları kılavuzuna göre bilgisayar teknolojileri bölümü Türkiye'nin yazılım sektörünün %78 oranında yetmişmiş insan gücünü karşılamaktadır. Uygulama ve teoriye yönelik eğitim verilen bu kurumlarda (Kalkınma Bakanlığı, 2014); öğretim elemanlarının öğrencileri derste aktif hale getirecek stratejileri benimsememeleri, yenilikçi ve dinamik ortamların kullanılmaması nedeniyle uygulamaya dönük eğitimler tam anlamıyla yapılamamakta ve öğrencilerden beklenen beceriler kazandırılmamaktadır (Binici & Necdet, 2004; Şahin & Fındık, 2008; Ünal & Çakır, 2016). Programlama mantığının en iyi şekilde anlaşılması ve sağlam bir temel oluşturulması hususunda özellikle bu kurumlarda görsel programlama ortamlarının kullanılmasının önemi ortaya çıkmaktadır (Cooper vd., 2003).

Çalışma Grubu

Araştırmanın çalışma grubunu 2015-2016 eğitim öğretim yılı bahar döneminde meslek yüksekokulu bilgisayar teknolojileri bölümü 2. sınıfta öğrenim gören ve gönüllük esasına göre seçilen 20 öğrenci oluşturmaktadır. Çalışma grubunun 2. sınıfta öğrenim gören öğrenciler arasından seçilmesinin nedeni; daha önce geleneksel öğretimle programla öğrenimi görmelerinden dolayı, Alice programının programlama öğrenimine sağladığı avantaj ve dezavantajlı durumları daha iyi ifade edebilir olmalarıdır. Ayrıca 2. sınıf öğrencilerine Alice programı ile programlama eğitimi verileceği açıklanmış ve sadece bu eğitimde yer almak isteyen öğrenciler çalışmaya dahil edilmiştir. Öğrencilerin Alice programı ile verilen programlama eğitimine karşı negatif bir tutum sergileyebilmelerinin önüne geçilmesi amacıyla çalışma grubu gönüllük esasına göre belirlenmiştir.

Öğrencilerin demografik bilgileri incelendiğinde çalışma grubunun %70'lik bir oranla erkek ağırlıklı olduğu belirlenmiştir. Öğrencilerden 15'inin yaş aralığının 18-20, beş öğrencinin ise 21-23 olduğu belirlenmiş ve çalışma grubunun %75 ile en yüksek yaş aralığının 18-20 olduğu tespit edilmiştir.

Öğrencilerden birinin genel liseden, 12'sinin meslek/teknik lisesinden, yedisinin anadolu meslek/teknik lisesinden mezun olduğu belirlenmiştir. Bu bilgiler ışığında çalışma grubunun %60 ile en çok meslek veya teknik liseden mezun oldukları tespit edilmiştir. Öğrencilerin beşinin lise mezuniyet derecesinin 2.00-3.00, sekizinin 3.01-4.00, yedisinin 4.01-5.00 arasında olduğu bilgisine ulaşılmıştır. %40 ile en yüksek lise mezuniyet derecesi aralığının 3.01-4.00 olduğu tespit edilmiştir. Öğrencilerin bölümlerini seçme nedenleri programlama başarısını ve ilgisini etkileyen bir unsurdur. Öğrencilerin bölüm seçme nedenlerini içeren bilgiler Tablo 1'de verilmektedir.

Tablo 1.*Öğrencilerin Bölümlerini Seçme Nedenleri*

Bölüm Seçme Nedeni	n	%
Bölümü sevdiğim için seçtim	8	40
Ailemin ve/veya çevrem bu bölümde okumamı istediği için seçtim	2	10
Daha fazla iş imkânı olduğu için seçtim	5	25
Toplumda bu mesleğe saygı duyulduğu için seçtim	0	0
Puanıma göre herhangi bir bölümü seçtim	5	25
Toplam	20	100

Bilgisayar programcılığı bölümünün temelinde programlama bilgisi bulunmaktadır. Öğrencilerin bölümlerini severek tercih etmelerinin dışında farklı etkenleri göz önüne bulundurarak gelmeleri, bölümlerine ve programlamaya karşı olumsuz bir tutumun oluşmasına neden olabilir. Programlamaya karşı oluşan olumsuz bir tutum araştırmanın sonuçlarını da olumsuz yönde etkileyebilir. Tablo 1 incelendiğinde öğrencilerin en fazla, %40 ile bölümlerini severek tercih ettikleri belirlenmiştir.

Öğrencilerin farklı programlama derslerindeki başarıları ile Ağırlıklı Geçme Notu Ortalama (AGNO) bilgileri incelenmiştir. Öğrencilerin farklı programlama derslerindeki ortalama puanları ve ortalama AGNO bilgileri Tablo 2'de verilmektedir.

Tablo 2.*Öğrencilerin Programlama Derslerindeki Ortalama Puanları ve AGNO Ortalamaları*

Programlama Dersleri ve AGNO Bilgisi	\bar{X}
Programlama Temelleri	67,45
Görsel Programlama 1	67,65
Görsel Programlama 2	74,8
Görsel Programlama 3	63,7
Nesne Tabanlı Programlama 1	60,65
Nesne Tabanlı Programlama 2	71,8
İnternet Programcılığı 1	59,4
İnternet Programcılığı 2	63,85
AGNO	2,75

Tablo 2 incelendiğinde öğrencilerin programlama derslerinden aldıkları puanların ve ortalama AGNO bilgilerinin ortalama düzeyde olduğu görülmektedir. Programlama eğitiminde temel kavramların öğretilmesi önemli yer tutmaktadır. Çünkü ancak eski bilgiler kullanılarak yeni uygulamalar

yapılabilir ve programlamada ilerlenebilir (Gomes & Mendes, 2007). Ayrıca programlama dillerinde temel kodların anlamları aynıdır. Sadece söz dizimlerinde farklılık görülebilmektedir. Öğrenciler dört dönem farklı programlama dersleri almalarına rağmen; bir başka ifadeyle temel kavramları sekiz farklı programlama dersinde görmelerine karşın, programlama derslerindeki puanları ortalama düzeydedir. İlgili tablo verileri programlama temelini iyi bir şekilde anlaşılmadığı, dolayısıyla ders başarısının ortalama düzeyde kaldığını gösterebilir. Bu durum programlama öğreniminde farklı öğretim yaklaşımlarının ve dinamik ortamların kullanılması gerektiğinin önemini ortaya çıkarmaktadır.

Uygulama Süreci

Uygulama süreci 2015-2016 eğitim-öğretim yılı bahar döneminde, beş hafta boyunca haftada dört ders saati olmak üzere toplam 20 saat sürmüştür. Alice programı ile nesne tabanlı programlama dillerinden Java programlama dilinin temel yapısının öğretilmesi hedeflenmiş, bu kapsamda Alice programı ile ders içi örnek ve uygulamalar gerçekleştirilmiştir. Uygulama sürecinde haftalara göre işlenen konu başlıkları Tablo 3'te verilmektedir.

Tablo 3.

Uygulama Sürecinde Haftalara Göre İşlenen Konu Başlıkları

Hafta	Konu Başlıkları
Hafta 1	Veri tipleri ve değişkenler, operatörler a. Veri Tipleri i. Tam sayı veri tipleri ii. Ondalık sayı veri tipleri iii. Karakterel veri tipleri iv. Mantıksal veri tipleri b. Değişken tanımlama i. Değişken tanımlama kuralları c. Operatörler i. Aritmetik operatörler
Hafta 2	Kontrol yapıları a. if yapısının kullanımı i. if / else ii. if / else if / else b. switch yapısının kullanımı i. switch / case c. Operatörler i. Mantıksal operatörler
Hafta 3	Döngüler a. For döngüsü b. While döngüsü c. Do / while
Hafta 4	Diziler a. Tek boyutlu dizi tanımlamaları b. Çok boyutlu dizi tanımlamaları
Hafta 5	Metotlar Kullanımı a. Metotlar i. Local ve Global değişkenler

Uygulamada Gagne'nin dokuz aşamalı öğretim modeli göz önünde bulundurularak öğretim tasarlanmıştır. Modelde basit seviyedeki yetenekler, daha üst seviyedeki yeteneklere alt yapı oluşturmakta ve öğrenme birbiri üstüne tanımlanan bir hiyerarşi olarak benimsenmektedir (Gagné,

1985). Programlama dili öğrenme süreci önce basit yetenekleri edinme sonrasında basit yetenekleri kullanarak karmaşık yetenekleri ortaya çıkarma ile oluşan bir öğrenme süreci olduğu için Gagne'nin dokuz aşamalı öğretim modeli çalışmada kullanılmıştır. Örnek teşkil etmesi amacıyla hafta 2 konusunun Gagne'nin dokuz aşamalı öğretim modeli göz önünde bulundurularak öğretimi aşağıda verilmektedir.

1. Dikkat Çekme : Öğrencilerin dikkatini çekmek amacıyla Alice programında yapılmış farklı kontrol yapısı kullanımlarını içeren animasyonlar öğrencilere gösterilmiştir. Öğrencilerin dikkati animasyonlara yoğunlaştıktan sonra bu animasyonların nasıl yapılabileceği, algoritmalarının nasıl olabileceği, hangi kodların kullanılabileceği ile ilgili sorular öğrencilere yöneltilmiştir.

2. Hedeflerden Haberdar Etme : Öğrencilere hafta 2 kapsamında yapacakları Alice uygulamaların çıktılarını gösterilmiş ve hem bu ders öğrenecekleri hem de geçmiş derslerde öğrendikleri bilgiler ile bu uygulamaları yapabilecekleri belirtilmiştir. Bu şekilde öğrenciler dersin amaçlarından haberdar edilmiştir.

3. Önbilgileri Hatırlatma : Öğrenciler hafta 2 konusuyla ilgili Alice uygulamalarını yaparken, algoritma oluşturmada ya da program yazımında hangi adımları izlemeleri gerektiği hatırlatılmıştır. Bir önceki hafta Alice programı ile öğrenilen veri tipleri ve operatörler konusuna yönelik hangi durumlarda hangi değişkenlerin kullanıldığı ile ilgili anımsatıcı bilgiler verilmiştir. İhtiyaç halinde eski Alice uygulamaları gösterilmiş ya da ek örnekler yapılmıştır.

4. İçeriğin Sunumu : Dersin içeriği olan kontrol yapıları; önemli yerlere vurgu yapacak ses tonuyla anlatılmış, if, switch vb. yapılar tahtada ve Alice programında gösterimleri yapıp projeksiyon cihazından yansıtılarak anlatılmıştır. Her yapının benzer ve ayrı yönleri, farklı kullanım şekilleri Alice programında gösterilmiştir. Ders anlatımı ayrıntılandırılırken hem Alice programında aynı örnek üzerinde, farklı kontrol yapıları kullanılarak gösterim yapılmış hem de farklı özellikteki örnekler yapılarak konunun daha iyi anlaşılması sağlanmıştır. Sonrasında öğrencilere Alice programı ile yapabilecekleri uygulamalar verilerek, kendi bilgisayarlarında çözmeleri istenmiştir.

5. Öğrenmeye Rehberlik Etme : Öğrenciler Alice programı ile uygulamalarını yaparken laboratuarda gezilerek gözlenmiştir. Programdan kaynaklı zorlanmalar olduğunda yardım edilmiştir. Uygulamalarında zorlandıkları yerlerde doğru çözüme ulaştırabilecek ipuçları verilmiştir. Doğru sonucu kendilerinin bulmalarına imkan tanınmış ve keşfederek öğrenmeleri desteklenmiştir. İpuçlarının yetersiz kaldığı durumlarda ek bilgiler sunulmuştur.

6. Davranışı Ortaya Çıkarma : Derste içerik verildikten ve Alice programı ile örnekler yapıldıktan sonra öğrencilerden Alice programı ile uygulama yapmaları istenmiştir. Uygulamalar sadece haftanın içeriği ile çözülebilecek uygulamalar olmayıp geçmiş bilgilerini de kullanacakları ayrıntıları içermektedir. Uygulamaları yapmaları için öğrencilere süre verilmiş, programlama mantığında takıldıkları noktalarda rehberlik edilmiştir. Teknik bir problem yaşandığı takdirde yardım edilerek sorun giderilmiştir.

7. Geribildirim Sağlama : Öğrencilerin ders anlatılımlarında kullanılan örnekleri yaparken ve Alice programı ile kendileri uygulama yaparken gidiş yolları incelenmiş, eksik veya yanlış olan yerler konusunda bilgilendirme yapılmıştır. Farklı çözüm yollarının doğruluğu test edilmiş ve bu çözüm yollarını örnek olarak diğer öğrencilerin görmeleri sağlanmıştır.

8. Performans Değerlendirmesi : Öğrencilerin ders içerisinde Alice programı ile yaptıkları örnek ve uygulamalar sınıf içinde doğruluk durumları ve farklı gidiş yolları açısından değerlendirilmiş ve geri dönüş sağlanmıştır.

9. Kalıcılık ve Bilgi Transferinin Sağlanması : Derste verilen Alice örnek ve uygulamaları haricinde Alice programıyla yapabilecekleri farklı sorular ve alıştırmalar öğrencilere yöneltilmiş, öğrendiklerini farklı durumlarda kullanabileceklerinin farkına varmaları sağlanmıştır. Ayrıca ders esnasında geribildirimler ile daha iyi öğrenmeleri ve öğrendiklerini pekiştirmeleri sağlanmıştır.

Ders içeriklerinin tümü yukarıda örnek bir konu anlatımı ile açıklanan modele göre işlenmiştir. Ders sürecinde Alice programı ile Java programlama dilinde ilgili haftanın konusu anlatıp örnekler yapılmış, öğrencilerin Alice programı ile uygulamalar yapması istenmiştir. Buna ek olarak öğrencilerden iki haftada bir eposta yolu ile değerlendirme notları toplanmıştır.

Veri Toplama Araçları

Çalışmada öğrencilerin Alice programı ile ilgili görüşlerini belirlemek amacıyla, öğrencilerden değerlendirme notları toplanmıştır. Öğrencilerin demografik özellikleri ve bölümü seçme nedenleri ile ilgili bilgiler öğrenci bilgi formu yardımıyla toplanmıştır.

Öğrenci değerlendirme notları

Alice programının Java programlama dili konularını öğrenmedeki katkıları, kullanımı, programlamayı kolaylaştıran ya da zorlaştıran durumları, süreç içerisinde sevilen, sevilmeyen durumları ve özellikleri içeren öğrenci görüşlerini belirlemek amacıyla öğrencilerden değerlendirme notları toplanmıştır. Öğrenci değerlendirme notları eposta yolu ile iki haftada bir olmak üzere öğrencilerden alınmıştır.

Öğrenci bilgi formu

Öğrencilerin cinsiyet, yaş aralığı, lise mezuniyet dereceleri, mezun olunan okul türü bilgilerini kapsayan demografik özelliklerini ve bölümü seçme nedenlerini belirlemek amacıyla araştırmacı tarafından hazırlanan bilgi formu kullanılmıştır. Bilgi formu araştırmacı tarafından hazırlandıktan sonra beş uzman görüşüne sunulmuş, hem biçimsel özellikleri hem de içeriği ile ilgili düzenlemeler yapılarak son hali verilmiştir.

Verilerin Analizi

Bilgi formu verileri betimsel analiz ile ve, öğrenci değerlendirme notlarından elde edilen nitel veriler ise içerik analizi kullanılarak analiz edilmiştir. Betimsel analizde frekans, yüzde, ortalama ve standart sapma kullanılmıştır. İçerik analizi ile veriler analiz edilirken öncelikle kodlar çıkarılmış kodlara uygun temalar belirlenmiştir. Güvenirliği sağlamak amacıyla araştırmacı dışında başka bir uzman tarafından kodlama tekrar yapılmıştır. 17 maddeden oluşan kodlama işleminin ardından iki kodlama arasında 15 madde eşleşmiştir. Araştırmacı ve uzman kişi tarafından eşleşmeyen 2 madde üzerinde görüşme yapılarak ortak fikirlere varılmış ve kodlama işlemi tamamlanmıştır. Eşleşme oranı %88,25 bulunmuştur.

Bulgular

Öğrenci Görüşlerine İlişkin Bulgular

Uygulama sırasında 20 öğrenciden elde edilen değerlendirme notları içerik analiz yöntemiyle analiz edilmiştir. Değerlendirme sonuçlarına göre elde edilen görüşler üç tema altında toplanmıştır. Bunlar :

- Alice programının programlama dili öğreniminde sağladığı yararlar
- Öğrencilerin Alice programı ile ilgili olumlu görüşleri
- Öğrencilerin Alice programı ile ilgili olumsuz görüşleri

Alice programının programlama dili öğreniminde sağladığı yararlar; görsellik, programlamaya olan etki ve sürüklen bırak ana başlıkları altında değerlendirilmiştir. Görselliğin alt başlığını kalıcılığı artırması oluşturmaktadır. Programlamaya olan etkinin alt başlıklarını; kod mantığını öğrenmeye yardımcı olması, programlama mantığını anlamayı kolaylaştırması, programlamaya olan ilgiyi artırması ve programlamayı eğlenceli hale getirmesi oluşturmaktadır. Sürüklen bırak ise hatasız kodlama yapma olanağı sağlaması alt başlığından oluşmaktadır. Alice programı ile tasarlanan ortamın programlama dili öğreniminde sağladığı yararlar ile ilgili görüşler Tablo 4'te verilmektedir.

Tablo 4.

Alice Programı ile Tasarlanan Ortamın Programlama Dili Öğrenimde Sağladığı Yararlar ile İlgili Görüşler

Alice programının programlama dili öğreniminde sağladığı yararlar		n	f
Görsellik (Kod işlevlerinin görsel olarak görülebilmesi)	Kalıcılığı artırması	8	9
	Kod mantığını öğrenmeye yardımcı olması	11	12
Programlamaya Olan Etkisi	Programlamaya olan ilgiyi artırması	6	9
	Programlama mantığını anlamayı kolaylaştırması	4	4
	Programlamayı eğlenceli hale getirmesi	4	4
Sürüklen-Bırak	Hatasız kodlama yapma olanağı sağlaması	3	3

Alice programının programlama dili öğreniminde sağladığı yararlar açısından etkileri incelendiğinde en çok; kullanılan kodların görsel olarak ne işe yaradığının görülebilmesinin kod mantığını anlamaya yardımcı olduğu belirtilmiştir. Değerlendirme notlarında bir öğrenci kod mantığını öğrenme ve sürüklen bırak özelliğinin avantajı olarak "*Kod yazmadan kolaylıkla program yazılabiliyor ve döngü gibi temel kavramların mantığını kolay bir şekilde öğreniyoruz.*" yorumunu yapmıştır. Başka bir öğrenci "*Programlamaya Alice ile başlasaydık kodları ve programları daha kolay ve hatasız bir şekilde öğrenebilirdik.*" yorumu ile Alice programının kod mantığını anlamayı ve programlama öğrenmeyi kolaylaştırdığını belirtmiştir. Cevaplar ayrıntılı bir şekilde incelendiğinde sekiz öğrenci kalıcılığı arttırdığını ifade etmiştir. Görselliğin kalıcılığı artırması ile ilgili bir öğrenci "*Görerek yaptığım şeyler aklımda daha kalıcı oluyor.*" yorumunu yapmıştır. Altı öğrencinin Alice programının programlamaya olan ilgisini arttırdığını, dört öğrencinin ise programlama mantığını anlamayı kolaylaştırdığını ifade ettikleri görülmektedir. Dört öğrenci Alice programının görselliği sayesinde programlamayı daha eğlenceli bulduklarını belirtmişlerdir. Bir öğrenci önceden programlama dersinin diğer derslerden herhangi bir farkı olmadığını ancak Alice programı ile derse gelme hevesinin arttığını ifade etmiştir.

Üç öğrenci Alice programının sürükle bırak mantığı ile kodların çalışma ekranına eklenebilmesi sayesinde kod hatasından uzak bir şekilde kodlama yapılabildiğini belirtmiştir.

Görüşler incelendiğinde Alice programı ile ilgili olumlu görüşlerin olduğu görülmektedir. Bu görüşler Alice programının olumlu özellikleri altında kategorize edilmiştir. Java kodlarının çalışma ekranın yanında görülmesi, nesnelere istenilen hareketlerin verilebilmesi, kod çıktılarının animasyon şeklinde görsel olarak görülebilmesi, sürükle-bırak özelliği ile kodların hazır olarak eklenmesi, eğlenceli olması ve programın kullanımının kolay olması Alice programının genel özellikleri ile ilgili olumlu görüşler olarak değerlendirilmiştir. Alice programı ile ilgili olumlu görüşler Tablo 5'te verilmektedir.

Tablo 5.

Alice Programı ile İlgili Olumlu Görüşler

Alice Programının Olumlu Özellikleri	n	f
Kod çıktılarının animasyon şeklinde görsel olarak görülebilmesi	9	13
Sürükle bırak özelliği ile kodların hazır olarak eklenmesi	9	11
Java kodlarının çalışma ekranın yanında görülmesi	7	7
Eğlenceli olması	7	7
Kullanımının kolay olması	5	5
Nesnelere istenilen hareketlerin verilebilmesi	2	2

Alice programı ile ilgili öğrencilerin olumlu görüşleri incelendiğinde öğrenciler en çok sevdikleri özellikler olarak; programda kodların sürükle-bırak mantığı ile hazır olarak eklenmesini ve kod çıktılarının animasyon şeklinde görsel olarak görülebilmesini belirtmişlerdir. Sürükle bırak özelliği ile ilgili bir öğrenci "*Kodları yazmak yerine seçip bırakmak ayrı bir kolaylık, hata yapmanın önüne geçiyor*" yorumunu yapmıştır. Cevaplar ayrıntılı bir şekilde incelendiğinde yedi öğrenci istenildiği zaman Alice çalışmalarında Java kodlarının çalışma ekranının yan tarafında ayrı bir pencerede görülmesini beğendiklerini belirtmişlerdir. Bir öğrenci Java kodlarının çalışma ekranının yan tarafında görülmesi ile ilgili "*Görsel ve kod kısmının bir arada olması kodları daha iyi anlamamı sağladı.*" yorumunu yapmıştır. Yedi öğrenci programın eğlenceli olduğunu belirtmiş, beş öğrenci Alice programının kullanımını kolay bulmuştur. Bir öğrenci Alice programı ile çalışmanın eğlenceli olduğunu "*Program çok zevkli. Uğraşırken veya uygulamaya yaparken canım hiç sıkılmıyor.*" yorumu ile ifade etmiştir. İki öğrenci nesnelere istenilen hareketlerin verilebilmesini sevdiğini belirtmişlerdir. Bu konuda bir öğrenci "*Nesnelere hareket ettirmek çok güzel ve eğlenceli. Bu sayede kodların ne işe yaradığı daha iyi anlaşılıyor.*" yorumunu yapmıştır.

Değerlendirme notları incelendiğinde öğrenciler olumlu görüşlerin yanı sıra olumsuz görüşlerde belirtmişlerdir. Değişken tanımlamanın zor olması, nesnelere eklem hareketi vb. ayrıntılarını hareket ettirmenin zor olması, döngü oluşturmanın zor olması, yazılımın donanım gereksinimlerinin yüksek olması ve arayüzün karışık bulunması Alice programına ait olumsuz görüşlerdir. Alice programı ile ilgili olumsuz öğrenci görüşleri Tablo 6'da verilmektedir.

Tablo 6.

Alice Programı ile İlgili Olumsuz Görüşler

Alice Programının Olumsuz Özellikleri	n	f
Değişken tanımlamanın zor olması	4	4
Yazılımın donanım gereksinimlerinin yüksek olması	3	5
Döngü oluşturmanın zor olması	2	2
Nesnelere eklem hareketi vb. ayrıntılarını hareket ettirmenin zor olması	2	2
Arayüzün karışık bulunması	2	2

Alice programı ile ilgili olumsuz görüşler incelendiğinde en çok; değişken atamanın zor olduğu ve döngü oluşturmanın zor olduğu belirtilmiştir. Bir öğrenci döngü ve değişken tanımlamada yaşadığı zorluklarla ilgili *"Döngülerde biraz zorlandım. Değişken tanımlamalarda çok zorlandım. Java'da daha pratik şekilde tanımlama yapıyorduk."* yorumunu yapmıştır. Ayrıca üç öğrenci yazılımın donanım gereksinimlerinin yüksek olmasından dolayı bilgisayarda kilitlenmelerin ve donmaların çok olduğunu bu durumun programla çalışılmasını engellediğini belirtmiştir. Bir öğrenci programın donanımsal gereksinimiyle ilgili *"Bazı bilgisayarlarda sistem gereksinimleri düşük kaldığından program hatalı çalışabiliyor."* yorumunu yapmıştır. Cevaplar ayrıntılı bir şekilde incelendiğinde iki öğrenci yazılımın arayüzünü karışık bulduklarını ve özelliklerin nereden ekleneceğini bulmalarının zaman aldığını ifade etmişlerdir. İki öğrenci nesnelere ayrıntılı bir şekilde hareket ettirmenin karmaşık olduğunu belirtmiştir. Bir öğrencinin nesnelere hareket ettirmenin zorluğuyla ilgili *"Nesneleri hareket ettirmek güzel ancak nesnelere ayrıntılı hareket ettirmek çok zor. Mesela bir insanı yürütmek çok ayrıntılı."* yorumunu yaptığı görülmüştür.

Sonuçlar ve Tartışma

Bu çalışmada, programlama dili öğretim ortamlarında Alice programının etkili, verimli ve ilgi çekici bir şekilde kullanımını sağlamak amacıyla öğrenci görüşlerini içeren nitel veriler analiz edilerek değerlendirilmiştir. 20 öğrenciden elde edilen değerlendirme notları analiz edildiğinde Alice programının programlama dili öğreniminde sağladığı yararlar ile Alice programının olumlu ve olumsuz yönleri üzerinde görüş belirtildiği görülmektedir.

Araştırma sonuçları incelendiğinde Alice programının programlama dili öğreniminde sağladığı yararlar ile ilgili öğrenciler en çok; kod mantığını öğrenmelerine yardımcı olduğunu, kodların görsel bir şekilde işlevinin görülebilmesinin kalıcılığı arttırdığını ve Alice programının programlamaya olan ilgilerini arttırdığını ifade etmişlerdir.

Biju (2013), Hayat vd. (2017), Johnsgard ve McDonald (2008), Schultz (2011), Solmaz (2014), Wang vd (2009) çalışmalarında Alice programının temel kod kavramlarını öğrenmeye yardımcı olduğunu ifade etmişlerdir. Liu vd. (2011) Alice programı ile çalışılmasının programlamada bilgi birikimini arttırdığını belirtmişlerdir. Courte, Howard(2006), Hayat vd. (2017) ve Bishop-Clark (2017) yaptıkları çalışmada, öğrencilerin Alice programının kendilerini programlamaya teşvik ettiğini belirttiklerini vurgulamışlardır. Hayat vd. (2017) ve Werner vd. (2009) Alice programı ile programlama öğrenmenin motivasyonu arttırdığını belirtmişlerdir. Werner vd. (2009) çalışmalarında kodlar ile istenilen animasyonların yapılmasının programlamaya olan ilgiyi arttırdığını ifade etmişlerdir. Buna karşın Wang vd. (2009) çalışmalarında Alice programı ile çalışmanın kullanıcıların motivasyonlarında herhangi bir artışın oluşturmadığı ve öğrenme deneyimlerine anlamlı yönde etkilemediği sonucuna ulaşmışlardır.

Alice programında eklenen nesnelere istenilen hareketleri yapabilmeleri için, çalışma ekranına sürükle-bırak mantığı ile kodlar eklenmektedir. Kullanıcılar hangi kodun nasıl bir sonuç verdiğini görsel olarak animasyon şeklinde görebilmektedir. Üstelik eklenen her bir kodun etkisine teker teker animasyonda bakılabilmektedir. Bu yüzden kullanıcılar kodlamada nasıl bir yol izlemeleri gerektiğini algoritmik olarak daha iyi anlayabilir ve kodların çalışma mantığını daha iyi kavrayabilirler. Böylece programlamayı yapabilme duygusu edinen kullanıcıların dersine olan ilgisinde artış sağlanabilir. Özellikle görsel zekaya sahip kullanıcılar için öğrenilen bilgilerin kalıcılığı artabilir. Öğrencilerin genelde sıkıcı olarak nitelendirdikleri programla dersine karşı olan ilgilerinin artması, programlama ile özgün teknolojilerin üretilebilmesi için umut vaat edicidir.

Alice programının olumlu özellikleri ile ilgili öğrencilerin en çok; kodların ne işe yaradığının görsel olarak görülebilmesini, sürükle-bırak özelliği sayesinde kod yazılmamasını ve Java kodlarının çalışma ekranında görülmesini sevindiklerini belirttikleri görülmektedir. Benzer şekilde Cooper vd. (2003) çalışmalarında sürükle-bırak mantığının sevilen bir özellik olduğunu belirtmiş ayrıca programlamaya yeni başlayan kullanıcılar için yararlı bir özellik olduğunu vurgulamışlardır. Solmaz (2014) öğrencilerin kodları hazır bir şekilde eklenilmesini sevindiklerini aktarmıştır. Cliburn (2008) ve Solmaz (2014) kodların görselleştirilmesinin programın en beğenilen özelliklerinden biri olduğunu, Powers vd. (2007) görselliğin öğrencilerin kod mantığını anlamalarına yardımcı olduğunu belirtmişlerdir.

Kullanıcılar Alice programı ile çalışırken kod yazmak yerine, ilgili kodları ekrana sürükleyip bırakarak programlarını oluşturmaktadırlar. Bu nedenle sözdizimi hataları ile uğraşmak yerine sadece kodların ne işe yaradığı ile ilgilenirler. Araştırma kapsamında öğrencilerin farklı programlama derslerindeki ortalamaları incelendiğinde ortalama düzeyde başarılarının olduğu gözlemlenmiştir (Bkz. Tablo 2). Bu durumun kodların ne işe yaradığının tam olarak anlaşılmasından kaynaklandığı düşünülmektedir. Yeni bir programlama diline geçerken farklı uygulamalardan başlamak yerine her seferinde temel bilgilerin yeniden verilmesi zaman kaybı olarak nitelendirilebilir. Kodların ne işe yaradığının tam olarak anlaşılması, yeni ve farklı uygulamaların, programların yapılmasına olanak sağlayabilir. Böylece bilişim sektöründe daha emin adımlarla ilerlenebilir.

Alice programında kodların ne işe yaradığı görsel bir şekilde görüldüğü için hem kod işlevleri daha iyi anlaşılabilir hem de programda oluşan hatalara daha kolay bir şekilde müdahale edilebilir. Bu görüşü destekler nitelikte Al-Linjawı ve Al-Nuaim (2010), Cooper vd. (2003), Solmaz (2014) yaptıkları çalışmada programlamada sıklıkla yaşanan kodlama hatalarının Alice programında daha kolay bir şekilde bulunduğunu ve programlamada hatalı gidiş yollarının kolay bir şekilde belirlendiğini ifade etmişlerdir. Ayrıca Alice programı ile oluşturulan animasyonlarda kullanılan kodlar istenildiğini takdirde çalışma ekranının yan tarafında görülebilir. Böylece kodların programlama dilinde kullanım şekli daha iyi şekilde anlaşılabilir.

Alice programı ile ilgili olumsuz görüş olarak öğrenciler en çok; değişken tanımlanması Java programlama diline göre daha fazla iş yükü getirdiğini, döngü oluşturmanın daha zor olduğunu ve yazılımın donanım gereksinimlerinin yüksek olmasından dolayı takılmaların yaşandığını belirttikleri sonucuna ulaşmıştır. Cliburn (2008) çalışmasında Alice programının gerektirdiği sistem belleği miktarı ve program yüklenme zamanının uzun olması gibi sistemsel problemleri olumsuz özellik olarak belirtmiştir. Değişken tanımlanması ile ilgili Cooper vd. (2000a; 2000b) yaptıkları çalışmalarda Alice programında değişken tanımlanmasına gerek olmadığını, değişken tanımlama ile uğraşmak yerine programlama dili yapılarının nasıl değiştirildiğine odaklanılabildiğini vurgulamışlardır. Ancak bu durum programlamaya yeni başlayan kullanıcılar için değişken tanımlanmasını ve değişken türlerini öğrenmek açısından bir dezavantaj yaratabilir.

Alice programı programlama dili öğrenimine kolaylıklar getirmesinin yanı sıra bazı teknik özellikleri öğrenenler için programlama diline göre daha zor gelebilir. Ayrıca Alice programının görselliğinin fazla olması ve özelliklerinin ayrıntılı olmasından dolayı sistem gereksinimleri yüksek olabilir. Bu durumlar kullanıcıları zorladığı için olumsuz bir özellik olarak belirtilmiş olabilir.

Bu çalışmada öğretimi yapan kişi ile araştırmacının aynı kişi olması bir sınırlılıktır. Araştırmacı çalışma grubunun önceden Java programlama dili eğitimi gördüklerinin farkındadır. Araştırmacı çalışma esnasında öğrenciler Alice programı ile uygulama yaparken, özellikle ilk haftalarda karşılaşılan zorlukların Java programlama dili bilgi eksikliğinden değil, Alice programı ile ilk defa çalışılmasından kaynaklanabilecek problemler olduğunu düşünmüş olabilir. Haftalar ilerledikçe

çalışmada karşılaşılan problemler azalmıştır. Bu durum sadece öğrencilerin Alice programı ile çalışmaya alışmaları ve programı iyi bir şekilde kullanmaya başlamaları olarak düşünülse de; Alice programının programlama mantığını öğrenmeye yardımcı olmasından dolayı da uygulamalar daha kolay bir şekilde yapılmış olabilir.

Öneriler

Eğitmenlere Yönelik Öneriler

Alice programı ayrıntılı bir programdır. Çalışmaya başlanılmadan önce programın ana hatlarıyla öğrenenlere tanıtılması, arayüzden kaynaklanabilecek ya da süreç içerisinde yaşanabilecek zorlukları azaltabilir. Eğer öğretimde amaç programlama dilinin ana hatlarıyla öğrenilmesi ise tüm eklemelerin hareket koordinasyonu gibi detaylı uygulamaların yapılmasından kaçınılmalıdır. Kullanıcılar bir süre sonra programlama mantığını anlamaya odaklanmadan ziyade ayrıntılarla uğraşırken derse karşı motivasyonlarını kaybedebilirler.

Alice programın çalışma mantığı kod yazmaya değil kodların ne işe yaradığının öğrenilmesine yöneliktir. Ancak öğretim amaçlarından biri kodlama pratikliğinin oluşturulması ise Alice programı ile çalışılırken kodların görüldüğü ekranının çalışma ekranının yan tarafında görünmesi sağlanmalıdır. Bu şekilde kullanıcılarda kodların sıralanması ve yazımı ile ilgili aşinalık oluşturulabilir.

Alice programı programlama öğrenmeyi farklı yönlerden kolaylaştırmaktadır. Ancak program, programlamada bazı konuların anlaşılmasını zorlaştırabilmektedir. Örneğin programda değişken tanımlama, programlama dilinde değişken tanımlamadan daha fazla iş yükü gerektirmektedir. Ayrıca Alice programında programlama dilindeki döngü türlerinin hepsinin bulunmamasıyla birlikte; bulunan döngü türleri ile ilgili kod eklenirken, algoritmasının oluşturulmasında kullanıcılar zorlanmaktadır. Bu yüzden döngü ile çalışırken öncelikle basit uygulamaların yapılması daha sonra ayrıntılı uygulamalara geçilmesi konunun daha iyi anlaşılmasını sağlayabilir.

Araştırmacılara Yönelik Öneriler

Bu çalışmada çalışma grubu olarak 20 kişilik küçük bir grup ile çalışılmıştır. Çalışma grubu daha önce programlama dersi alan ikinci sınıf öğrencilerinden oluşmaktadır. Çalışma kapsamında sadece nitel veriler incelenmiştir. Farklı çalışmalarda çalışma grubu artırılıp daha çeşitli veriler elde edilebilir. Nicel veriler de kullanılarak çalışma sonuçları desteklenebilir. Ayrıca programlamaya yeni başlayan kullanıcılar ile çalışılarak Alice programının programlama öğrenimine olan etkisi nicel ve nitel veriler kullanılarak araştırılabilir.

Nitel sonuçlar genel olarak değerlendirildiğinde; kodların ne işe yaradığının görsel olarak görülebilmesi, kod yazılmadan kodların eklenmesi sayesinde söz dizimi hatalarıyla uğraşmak yerine sadece kodların ne işe yaradığına odaklanılması, kullanıcıların programlama yaparken aynı zamanda özgün uygulamalar geliştirerek eğlenirken öğrenme imkânı elde etmesi, programlama öğrenimine yeni başlayan kişilerin programlamayı sevmesi ve programlama mantığının daha iyi anlaşılması için Alice programının kullanılmasının faydalı olacağı düşünülmektedir.

Kaynaklar / References

- Aktunc, O. (2013). A teaching methodology for introductory programming courses using Alice. *International Journal of Modern Engineering Research (IJMER)*, 3(1), 350-353.
- Al-Linjawi, A. A., & Al-Nuaim, H. A. (2010). Using Alice to teach novice programmers OOP concepts. *Journal of King Abdulaziz University: Science*, 22(1), 59-68.
- Baştemur Kaya, C., & Çakır, H. (2015). *Meslek yüksekokulu bilgisayar teknolojileri bölümü öğrencilerinin mesleki bilgi ve kariyer yeterlilikleri: İhtiyaç analizi*. 9th International Computer & Instructional Technologies Symposium' da sunulmuş bildiri, Anadolu Üniversitesi, Afyonkarahisar.
- Biju, S. M. (2013). Taking advantage of Alice to teach programming concepts. *E-Learning and Digital Media*, 10(1), 22-29.
- Binici, H., & Necdet, A. (2004). Mesleki ve teknik eğitimde arayışlar. *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, 24(3), 383-396.
- 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.
- Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051-1058.
- Cliburn, D. C. (2008). *Student opinions of Alice in CS1*. Paper presented at the Frontiers in Education Conference, 2008. FIE 2008. 38th Annual.
- Cohen, L., & Manion, K. (2000). Action research. In L. Cohen, L. Manion & K. Morrison (Eds.), *Research methods in education* (5 ed., pp. 226-244). London: Routledge-Falmer.
- Cooper, S., Dann, W., & Pausch, R. (2000). *Alice: a 3-D tool for introductory programming concepts*. Paper presented at the Journal of Computing Sciences in Colleges.
- Cooper, S., Dann, W., & Pausch, R. (2003). Using animated 3d graphics to prepare novices for CS1. *Computer Science Education*, 13(1), 3-30.
- Courte, J., Howard, E. V., & Bishop-Clark, C. (2006). Using Alice in a computer science survey course. *Information Systems Education Journal*, 4(87). <http://isedj.org/4/87/>. ISSN: 1545-679X. (A preliminary version appears in The Proceedings of ISECON 2005: §2133. ISSN: 1542-7382.)
- Creswell, J. W., & Poth, C. N. (2017). *Qualitative inquiry and research design: Choosing among five approaches*. Los Angeles: Sage publications.
- Daly, T. (2011). Minimizing to maximize: an initial attempt at teaching introductory programming using Alice. *Journal of Computing Sciences in Colleges*, 26(5), 23-30.
- Daly, T. (2011). Minimizing to maximize: an initial attempt at teaching introductory programming using Alice. *Journal of Computing Sciences in Colleges*, 26(5), 23-30.
- Feurzeig, W., Papet, S. A., & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, 19(5), 487-501.
- Gagné, R. (1985). *The conditions of learning and theory of instruction*. Fort Worth, TX: Holt, Rinehart and Winston: Inc.
- Gorman, H., & Bourne, L. E. (1983). Learning to think by learning LOGO: Rule learning in third-grade computer programmers. *Bulletin of the Psychonomic Society*, 21(3), 165-167.
- Hayat, K., Al-Shukaili, N. A., & Sultan, K. (2017). Alice in Oman. *Education and Information Technologies*, 22(4), 1553-1569.

- Howard, E. V., Evans, D., Courte, J., & Bishop-Clark, C. (2006). *A qualitative look at Alice and pair-programming*. Paper presented at the Proceedings of ISECON 2006.
- Johnsgard, K., & McDonald, J. (2008). *Using alice in overview courses to improve success rates in programming i*. Paper presented at the Software Engineering Education and Training, 2008. CSEET'08. IEEE 21 st Conference on.
- Kalkınma Bakanlığı (2014). *Onuncu kalkınma planı 2014-2018 mesleki eğitimin yeniden yapılandırılması çalışma grubu raporu*. <https://abdigm.meb.gov.tr/projeler/ois/egitim/022.pdf> sayfasından erişilmiştir.
- Kayabaşı, E. (2016). *Öğretmen adaylarının Alice deneyimi: 3B ortamda programlama*. (Yüksek Lisans Tezi, Uludağ Üniversitesi, Eğitim Bilimleri Enstitüsü, Bursa.
- K ölling, M. (2010). The greenfoot programming environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), Article 14. <http://doi.acm.org/10.1145/1868358.1868361>.
- Kölling, M., Quig, B., Patterson, A., & Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Computer Science Education*, 13(4), 249-268.
- Liu, J., Lin, C.-H., Hasson, E. P., & Barnett, Z. D. (2011). *Introducing computer science to K-12 through a summer computing workshop for teachers*. Paper presented at the Proceedings of the 42nd ACM technical symposium on Computer science education.
- McMillan, J. H. (1996). *Educational research: Fundamentals for the consumer*. New York: HarperCollins College Publishers.
- Merriam, S. B. (2013). *Nitel araştırma: Desen ve uygulama için bir rehber* (S. Turan, Çev.). Ankara: Nobel Yayınevi.
- National Research Council. (1999). *Being fluent with information technology*. Washington:DC, USA: National Academies Press.
- Öğrenci Seçme ve Yerleştirme Sistemi (2011). *2011-ösys: yükseköğretim programları ve kontenjanları kılavuzu*. <http://www.osym.gov.tr/TR,1092/2011-osys-yuksekogretim-programlari-ve-kontenjanlari-kilavuzu.html> sayfasından erişilmiştir.
- Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York: BasicBooks.
- Powers, K., Ecott, S., & Hirshfield, L. M. (2007). Through the looking glass: teaching CS0 with Alice. *ACM Sigcse Bulletin*, 39(1), 213-217.
- Partnership for 21st Century Skills [P21] (2009). *P21 framework definitions*. http://www.p21.org/storage/documents/P21_Framework_Definitions.pdf sayfasından erişilmiştir.
- Price, K. W. (2013). *Using visual technologies in the introductory programming courses for computer science majors*. United States: Nova Southeastern University.
- Resnick, M. (1990). MultiLogo: A study of children and concurrent programming. *Interactive Learning Environments*, 1(3), 153-170.
- Resnick, M. (2013). Learn to code, code to learn. *EdSurge*, May. Retrieved from <https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn>
- Resnick, M., Flanagan, M., Kelleher, C., MacLaurin, M., Ohshima, Y., Perlin, K., & Torres, R. (2009). *Growing up programming: democratizing the creation of dynamic, interactive media*. Paper presented at the CHI'09 Extended Abstracts on Human Factors in Computing Systems, ACM.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silverman, B. & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.

- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137-172.
- Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education, 10*(1), 73-88.
- Schultz, L. A. (2011). Student perceptions of instructional tools in programming logic: A comparison of traditional versus Alice teaching environments. *Information Systems Education Journal, 9*(1), 60-66.
- Solmaz, E. (2014). *Programlama dili öğretiminde Alice yazılımının ders başarısı, eleştirel düşünme ve problem çözme becerileri ile üstbilişsel farkındalık düzeyine etkisi*. Doktora Tezi, Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.
- Şahin, İ., & Fındık, T. (2008). Türkiye’de mesleki ve teknik eğitim: Mevcut durum, sorunlar ve çözüm önerileri. *Türkiye Sosyal Araştırmalar Dergisi, 12*(3), 65-86.
- Ünal, E., & Çakır, H. (2016). İşbirlikli teknolojilerle desteklenen yapılandırmacı öğrenme ortamının akademik uğraşıya etkisi. *Journal of Instructional Technologies & Teacher Education, 5*(1), 13-18.
- Wang, T.-C., Mei, W.-H., Lin, S.-L., Chiu, S.-K., & Lin, J. M.-C. (2009). *Teaching programming concepts to high school students with alice*. Paper presented at the Frontiers in Education Conference, 2009. FIE'09. 39th IEEE.
- Werner, L., Denner, J., Bliesner, M., & Rex, P. (2009). *Can middle-schoolers use Storytelling Alice to make games?: results of a pilot study*. Paper presented at the Proceedings of the 4th International Conference on foundations of digital games.
- Zhang, X., Liu, L., de Pablos, P. O., & She, J. (2014). The auxiliary role of information technology in teaching: Enhancing programming course using Alice. *International Journal of Engineering Education, 30*(3), 560-565.

Yazarlar

Ceren BAŞTEMUR KAYA, Nevşehir Hacı Bektaş Veli Üniversitesi Bilgisayar Teknolojileri Bölümü’nde öğretim görevlisidir. Gazi Üniversitesi Bilgisayar ve Öğretim Teknolojileri Bölümü’nde doktora eğitimine devam etmektedir. Çalışma alanları öğretim teknolojileri, öğrenme ortamları üzerindedir.

Dr. Hasan ÇAKIR, Gazi Üniversitesi Bilgisayar ve Öğretim Teknolojileri Bölümü’nde doçent doktordur. Çalışma alanları arasında öğretim teknolojileri, harmanlanmış öğrenme, dinamik web teknolojileri, öğrenme ortamları ve performans iyileştirme yer almaktadır.

İletişim

Öğr. Gör. Ceren BAŞTEMUR KAYA, Nevşehir Hacı Bektaş Veli Üniversitesi, Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Nevşehir, Türkiye.

e-mail: ceren@nevsehir.edu.tr

Doç. Dr., Hasan ÇAKIR, Gazi Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Bölümü, Ankara, Türkiye.

e-mail: hasanc@gazi.edu.tr

Summary

Purpose and Significance. The purpose of this study is to investigate the students' views to ensure use of Alice software effectively, efficiently and interestingly in learning environments in learning Java programming language by students studying at computer technologies department. The question which is "What are the student views investigated with the purpose of determining the use of Alice software in learning programming?" is answered within the framework of this purpose.

Methodology. In the study, descriptive case study method was utilized. The study group of the research consists of 20 students who study at the 2nd year of vocational schools, department of computer technologies in the spring semester of 2015-2016 academic year.

The application was carried out five weeks to be four hours in a week and took 20 hours in total. With Alice software, it is aimed to teach the basic structure of Java programming language and in that scope, examples and applications were realized using Alice software within the course.

In the application, the education is designed in accordance with Gagne's nine events of instruction model. During the courses, first the subject of the related week was told to students in Alice software, examples were solved and then practices were given to students.

In the study, reflections were collected from students via e-mail in every two weeks with the purpose of determining the students' views. Information on demographic features of the students and their reasons to choose this department is collected by means of an information form. Information form data was analysed using descriptive analysis. In addition the data containing students' views was analysed using content analysis.

Results. The reflections obtained from the students were evaluated using content analysis. Views obtained according to evaluation results are gathered under three themes; 1) Benefits of Alice software's on programming language learning, 2) Students' positive views on Alice software, 3) Students' negative views on Alice software.

When the usefulness of the Alice software and its impact on programming language learning are examined, it was stated that being able to see the function of codes visually helps understanding the encoding logic the most. When the answers were examined in detail, eight students stated that it increases persistency. It is seen that six students stated that Alice software increased their interest in programming and four students stated that it facilitates understanding programming logic. Four students specified that they find programming more fun by means of the visuality of Alice software. Three students indicated that coding was possible without error, due to the codes can be added to the working screen using drag and drop logic.

When the students' positive views on Alice software are examined, students stated that adding codes readily with drag-drop logic and being able to see code outputs visually in the form of animation are their favourite features. When the answers were examined in detail, seven

students and the researcher stated that they liked that Java codes are displayed in a separate window. Seven students indicated that the program is fun. Also five students find that Alice software easy to use. Two students stated that they like the ability to give any motion to the objects.

When the negative views on Alice software are investigated, it was pointed out that it is difficult to assign a variable and it is difficult to create a loop In addition, three students stated that the computer frequently freezes due to the high hardware requirements of the software and this prevents working with the software. When the answers were analyzed in detail, two students defined that they find the software interface complex. In addition to two students expressed that it is complicated to move objects in detail.

Discussion and Conclusion. When the study results were examined, the students stated that it helped them learn the coding logic, being able see the function of codes visually increased persistency and Alice software increased their interest in programming in relation to the practicability of Alice software and its benefits on programming language. In Alice software, users are able to see the result of each code in the form of an animation. Moreover, it is possible to see the impact of each code one by one in the animation. For that reason, users are able to understand path to be followed in encoding algorithmically and they are able to comprehend the working logic of codes better. Thus, interest of the users can be increased in the course. Especially for the users who have visual intelligence, the persistency of learned information can be increased. Increasing the interest of students in programming course, which they describe as boring in general, is promising for producing original technologies. In relation to positive features of Alice software, it is seen that the students stated what they like the most is ability to see the function of codes visually, not writing codes thanks to drag-drop feature and seeing Java codes on the working screen.

While working with Alice software, users create their programmes by dragging and dropping related codes into the screen instead of writing codes. For that reason, they are interested in the function of codes rather than dealing with syntax errors. When the averages of students in different programming courses are examined within the scope of the study, it was observed that they have an average level success (see Table 2). It is thought that the reason of this is the fact that the function of the codes is not understood completely. Rather than starting from different practices while switching to a new programming language, giving the basic information again each time may be considered as loss of time. Understanding the function of codes clearly may provide possibility for new and different practices and programmes. Thus, programmers can proceed with more confident steps in the IT sector. As a negative view on Alice software, it was concluded that students state that defining variables brings more work load compared to Java programming language, it is more difficult to create loops and computer freezes due to the high hardware requirements of the program. While Alice software facilitates learning programming language, some of its technical features may be difficult for learners compared to programming language. Moreover, due to high visuality and detailed features of Alice software, system requirements might be high. Since these situations may force users, they might have been stated as negative features.

Suggestions. In this study, the research group consisted of a small group of 20 students. Within the scope of the study, only qualitative data was examined. Various data can be obtained in different studies by expanding the study group. The study results can also be supported using quantitative data. In addition, the impact of Alice software on learning programming can be investigated using qualitative and quantitative data by means of working with users who started programming recently. When the qualitative results are evaluated in general; it is thought that using Alice software will be useful for seeing the function of codes visually, focusing on the function of codes instead of dealing with syntax errors by means of adding codes without writing codes, users have the opportunity to learn by having fun, people who started programming recently like programming and better understand the programming logic. Alice software is a detailed software. Introducing it to the learners before starting with Alice software may reduce difficulties which might be caused by interface or within the process. If the purpose of education is to teach the programming language with its main lines, detailed applications such as movement coordination of all joints should be avoided. After a certain time, users may lose their motivation towards the course while dealing with details instead of focusing on understanding the programming logic. The operating logic of Alice software is towards learning the function of codes, not writing codes. However, if one of the purposes of education is to create encoding practicality, it should be ensured that the codes are displayed next to the working screen. Thus, a familiarity related to the sorting and writing of codes can be formed in users.

Alice software may make understanding some subjects of programming difficult. For example, defining variables require more work load than defining variables in programming language. Moreover, Alice software does not include all of the loop types in programming language. In addition, while the code related to the types of loops found is appended, users are forced to create their algorithm. For that reason, while working with a loop, simple applications can be made at first. Then more detailed applications can be proceeded with. This approach can provide a better understanding of the subject.