

## OKUL ÖNCESİ DÖNEMİ ÇOCUKLARI İÇİN BİLGİSAYAR PROGRAMLAMA

**Büşra SOMUNCU**

Çukurova Üniversitesi, [busrasomuncu@gmail.com](mailto:busrasomuncu@gmail.com)

**Durmuş ASLAN**

Doç. Dr., Çukurova Üniversitesi, [asland@cu.edu.tr](mailto:asland@cu.edu.tr)

### ÖZET

21. yüzyıl ile birlikte teknoloji hızla gelişmiş ve bu yüzyılı teknoloji çağı olarak adlandırmıştır. Teknoloji odaklı olan günümüz dünyasında üretebilen bir toplum olabilmek için yeni yetişen nesillere var olan programları tüketmek yerine yeni programlar üretmeleri için yol göstermek önem kazanmaktadır. Bu durum ise okullarda bilgisayar programlama eğitimini popüler hale getirmiştir. Bilgisayar programlama günümüzde yeni okuryazarlık olarak kabul edilmektedir. Tıpkı okuma yazma öğrenmek gibi bilgisayar programlama da çocukların düşünme becerilerini geliştirmekte ve kendilerini ifade etmelerine olanak sağlamaktadır. Bilgisayar bilimi uygulamalarına erken yıllarda başlamak, çocuklara bilgisayar dünyasını nasıl şekillendirdiğini fark etmelerine, bilgisayar ve programlama ile ilgili bazı temel kavramları adlandırmalarına ve bunları tanımalarına yardımcı olmaktadır. Günümüzde okul öncesi dönemi çocuklarının bilgisayar programlamalarına olanak sağlayan birçok ara yüz ve araç bulunmaktadır. Bu ara yüz ve araçlar grafik tabanlı programlama araçları ve somut programlama araçları olarak iki başlık altında toplanmıştır. Bu derleme çalışması bilgisayar programlama eğitiminin okul öncesi dönemi çocuklarıyla uygulanmasına dair bilgi vermeyi, verilen bilgilerle Türkiye’de bu alanda yapılacak çalışmalara ve uygulamalara yardımcı olmayı amaçlamaktadır. Bu çalışmanın okul öncesi dönemi çocuklarıyla bilgisayar programlama ve kodlama üzerine yapılacak yeni araştırmalara yol göstereceği ve katkı sağlayacağı umulmaktadır.

**Anahtar Sözcükler:** Bilgisayar programlama; okul öncesi; kodlama.

### COMPUTER PROGRAMMING FOR PRESCHOOL CHILDREN

#### ABSTRACT

Technology has developed rapidly with the 21st century and this century called as the age of technology. In order to become a society that can produce in today's technology-oriented world, it becomes important to guide the new generations to produce new programs instead of consuming existing ones. Therefore computer programming education has become popular in schools. Computer programming is now regarded as a new literacy. Just as learning to read and write, computer programming also allows children to develop their thinking skills and express themselves. To begin early with the computer science practices, helps children to realize how they shape the computer world and to name and identify some basic concepts related to computer and programming. Nowadays, there are many interfaces and tools that enable computer programming for preschool children. These interfaces and tools are grouped under two headings: graphical programming tools and tangible programming tools. The aim of this review paper is to give information about the practice of computer programming education with preschool children and to serve information for the works and practices which will be made in this field in Turkey. It is hoped that this study will

lead and contribute to new researches on computer programming and coding with preschool children.

**Keywords:** computer programming; preschool; coding.

## **GİRİŞ**

Günümüz modern yaşamında çevremiz dijital teknolojilerle çevrili durumdadır ve bilgisayar programlama, sadece bilgisayar bilimcilerinin değil herkesin bilmesi gereken bir alan haline dönüşmüştür. Bu yaygınlaşmada eğitimin büyük bir payı bulunmaktadır. Çocukların gelecek yaşamlarında dijital dünya içinde kendilerini daha iyi ifade edebilmeleri için küçük yaşlardan itibaren nitelikli bir bilgisayar programlama eğitimine ihtiyaç duyulmaktadır. Bu nedenle, bu çalışmada eğitimde teknoloji, bilgisayar programlamanın tarihçesi, bilişimsel düşünme ve okul öncesi eğitimde bilgisayar programlama konuları ele alınmıştır.

Günlük yaşantımız içerisindeki dijital teknolojiden küçük çocuklar da en az yetişkinler kadar haberdardır. Örneğin, çocuklar her gün ellerini salladıklarında kâğıt havlu veren otomatik makineler veya fotoğraf çekebilen, müzik çalabilen cep telefonları gibi dijital teknoloji ürünleri ile karşılaşır ve bunların nasıl çalıştığını bilir (Kazakoff, Sullivan ve Bers, 2013: 245). Ancak erken çocukluk döneminde çocuklar, sınıflarında dijital teknoloji ile ilgili çok az şey öğrenmektedirler. Çocukları, "küçük mühendisler" olarak şehir ve köprü inşa ederken karton kutuları veya artık materyalleri kullanırken görmek olağandır. Bunun yanı sıra çocukları dijital teknolojiyi kullanarak bir ürün oluşturduğunu görmek olağan bir durum değildir (Bers, 2008; Bers, 2010: 1). Eğitim alanına giren ve çıkan diğer geçici eğilimlerden farklı olarak teknoloji kalmaya devam etmekte ve okullarda teknoloji kullanımının zaman içinde önemli ölçüde artacağı tahmin edilmektedir (Hicks, 2011: 188).

Gelecekte başarı, ister bireyler ve topluluklar isterse şirketler için olsun, bir bütün olarak baktığımızda ne kadar bildiğimize değil, yaratıcı düşünebilme ve hareket etme gücümüze bağlı olacaktır (Resnick, 2003). Bu yüzden, gelecekteki işgücünü desteklemek, hatta daha da önemlisi gençleri kendi dijital teknolojilerini yönetme ve oluşturma, bugünün toplumunda başarılı olma yetkinlikleri ile güçlendirmek için bireylere küçük yaşlardan itibaren bilgisayar programlama eğitimi verilmelidir (European Schoolnet, 2015: 13-15). 21. yüzyıl ile birlikte büyük bir hızla gelişen teknoloji, içinde

bulduğumuz yüzyılı teknoloji çağı olarak adlandırmış ve okullarda bilgisayar programlama eğitimini popüler hale getirmiştir.

Bilgisayar programlama, tüm dijital teknolojilerin temelini oluşturan ve 21. yüzyıl okuryazarlığı için önemli olan bir beceridir. Bu beceri aynı zamanda, çocukların gelecekteki kariyerleri ve kişisel uğraşları için ihtiyaç duyacağı bir beceridir (Kazakoff, Sullivan ve Bers, 2013: 246-247). Öte yandan, okullarda öğrencilere teknoloji kullanımını ve bilgisayar bilimini öğretmede öğretmene büyük rol düşmektedir. Öğretmenin kendisi bu konuda ne kadar yetkin ve becerikli olursa öğrenciler de bilgisayar bilimleri konusunda o kadar başarılı olacaktır. Önümüzdeki yıllarda, eğitim sisteminde sınıfta kullanılan teknoloji miktarı açısından büyük bir artış görüleceği gibi, öğretmenlerin teknolojiye yönelik tutumlarında, teknolojiyi başarılı bir şekilde entegre etme ve çözme becerisinde bir değişiklik görüleceği beklenmektedir (Hicks, 2011: 189).

Ülkemize baktığımızda, bilgisayar okuryazarlığı kavramı öğretim programlarında vurgulanan kavramlardan biri olarak karşımıza çıkmaktadır. Ancak uygulamalarda bu kavramın sadece okuma yönüyle ele alındığı, bilgisayar yazma işinin ise ihmal edildiği görülmektedir. Oysaki bilgisayar okuryazarlığı uzun yıllardır birçok ülkenin öğretim programında yer almakta ve uygulanmaktadır (Akpınar ve Altun, 2014: 3). Örneğin, 2014 yılında İngiltere’de Kraliyet Raporu ile bilgisayar bilimi, ulusal müfredatta bir konu olarak tanıtılmıştır. İngiltere’de çocuklar beş yaşında okula başlamaktadır. Bilgisayar bilimi dersleri ise çocukların okula başladıkları ilk yıldan itibaren eğitim hayatlarına dâhil olmaktadır (Manches ve Plowman, 2017: 193). Temel eğitim düzeyinde verilecek bilgisayar okuryazarlığı dersinin programlama eğitimi için de iyi bir zemin oluşturacağı kabul edilmektedir.

Günümüzde, yeni teknolojilere erişim artarken, çocukların bu araçların nasıl ve neden çalıştıkları konusundaki anlayışı da yeni bir araştırma alanı olarak giderek yaygınlaşmaktadır. Bilgisayar programlama ve robotik alanı, çocukların bu anlayışını geliştirmede önemli bir yere sahiptir. Robotik ve bilgisayar programlama çocuklara motorun, sensörün ve elektronik cihazların nasıl çalıştığını uygulamalı projelerle etkileşimli bir şekilde öğrenmelerini sağlar. Robotik ve programlama çocukların hipotezlerini test etmelerine, problem çözme süreçlerine girmelerine ve kişisel olarak

anlamli keşifler yapmalarına olanak sunmaktadır (Elkin, Sullivan ve Bers, 2016: 170). Eğitimcilerin ve ebeveynlerin bilgisayar programlamanın çocukların gelişim alanları ve becerileri üstündeki etkisini fark etmeye başlaması ile beraber bilgisayar programlama önem kazanmıştır ve her geçen gün bilgisayar programlamanın eğitim alanındaki önemi artmaktadır.

Bilgisayar programlamanın tarihi 19. yüzyıla dayansa da eğitime yansması ile ilgili çalışmalar son yıllarda artış göstermiştir. Çocuklarla yapılan çalışmaların büyük çoğunluğu ilköğretim kademesindeki öğrencileri kapsamaktadır. Okul öncesi dönemini kapsayan çalışmalara rastlansa da bu çalışmalar yeterli sayıda değildir. Bu derleme çalışmasında, bilgisayar programlama eğitiminin okul öncesi dönem çocuklarıyla uygulanmasına dair bir takım bilgiler sunulması ve böylece Türkiye’de bu alanda yapılacak çalışmalara ve uygulamalara katkı sağlanması amaçlanmıştır. Bu çalışma kapsamında, bilgisayar programlama eğitiminin okul öncesi eğitimi ile nasıl bir araya gelebileceği belirlenmeye çalışılmıştır. Bu amaçla literatürde yer alan kodlama, bilgisayar programlama ve bu alanların eğitime ve erken çocuklukta kullanımına yansması ile ilgili çalışmalar incelenmiştir. Bu çalışmalardan elde edilen bulgular eğitimde teknoloji, bilgisayar programlama ve tarihçesi, bilişimsel düşünme, eğitimde bilgisayar programlama ve okul öncesinde bilgisayar programlama olmak üzere beş başlık altında toplanmıştır.

### **Eğitimde Teknoloji**

Günümüzde, hayatımızın her alanında bulunan teknolojinin, öğrenciye ve öğretmene en üst seviyede fayda sağlayacağı şekilde eğitim ortamlarıyla bütünleştirilmesi oldukça önemlidir. Teknolojinin tarihi, ilk insanların taşları bileyerek yaptıkları bıçaklardan bugün birçok şeyin yapılmasını mümkün kılan makinelere kadar uzanır. Teknoloji sürekli olarak yeni teknolojiler üretilmesi ve var olan teknolojilerin geliştirilmesi ile genişlemekte ve olağanüstü bir hızla gelişmektedir (International Technology Education Association, 2007: 2). Bilim ve teknolojideki bu hızlı ve köklü gelişmeler, içinde bulunduğumuz 21. yüzyılı bilgi çağı olarak adlandırmaktadır (Coşar, 2013: 1). Bilişim teknolojileri evlerimizden işyerlerimize, eğitimden sanata hemen hemen her alanda temel veya destek unsuru olarak hayatımızda yer etmiştir (Erdoğan, 2005: 1).

Okulların ve eğitimcilerin de bu bilişim teknolojilerini eğitim ortamlarına dâhil etmesi gerekmektedir. Günümüz eğitim kurumlarından eğitimin yapılarını ve programlarını teknolojiye bütün bu gelişmelere paralel olarak uyarlamaları beklenmektedir (Coşar, 2013: 1-2). Fiske ve Hammond (1997), eğitimde teknoloji kullanımının, öğrencilerin yeni bir çağa geçmesiyle birlikte deneyimlerinin genel kalitesi için hayati önem taşıdığı belirtmektedirler (Akt. Parker, 2014: 12). Özenli bir şekilde hazırlanan ve dikkatlice uygulanan teknolojinin, öğretim uygulamalarının etkisini hızlandırmada, arttırmada ve genişletmede önemli bir etkisi olduğu unutulmamalıdır (Office of Educational Technology, 2016: 3).

Geçmiş yıllarda okullarda teknoloji kullanımı günümüzdeki kadar erişilebilir değildi. Okulların, öğrenme deneyimini geliştirmek için internet, e-posta, akıllı tahta gibi dijital teknoloji araçları yoktu. Günümüzde ise öğrenciler son derece teknoloji meraklılarıdır. En küçük çocuklar bile teknolojiyi sınıf içinde ve dışında kolaylıkla kullanabilmektedirler. “Dijital nesil” olarak da adlandırılan günümüz çocukları teknolojiye bu kadar yetkinlik gösterirken, öğretmenlerin de teknolojik yeterlilik düzeylerinin ve teknolojiyi başarıyla uygulayabilme yeteneklerinin günümüz teknolojisine uygun olması gerekmektedir (Hicks, 2011: 188).

192

Eğitimde teknoloji kullanımı, araç olarak teknoloji kullanımı ve amaç olarak teknoloji kullanımı olmak üzere iki boyutta incelenebilir. Araç olarak teknoloji kullanımında, öğretmen teknolojik araç gereçleri ders anlatımında destekleyici materyal olarak kullanmakta ve teknolojik araç gereçler öğrenen konumunda yer almaktadır. Amaç olarak teknoloji kullanımı ise, öğretmenin öğrencilere teknolojiyi öğretmesini içermektedir. Bu boyutta teknolojik araç gereçler öğrenilen konumundadır.

### **Bilgisayar Programlama ve Tarihçesi**

Bilgisayar programlama, bir bilgisayarın davranışını belirleme tekniğidir (Morgado, Cruz ve Kahn, 2010: 309) ve bilgisayarın belirli bir görevi yerine getirmesini, sorunları çözmesini, insan etkileşimini sağlamasını mümkün kılmak için çeşitli komutlar geliştirme ve uygulama sürecini içermektedir (European Schoolnet, 2015: 21). En genel tanımı ile programlama, bilgisayara komutlar vererek yapılması istenen işleri yaptırmaktır (Saygıner ve Tüzün, 2017: 79).

Program, diğ er bir adıyla yazılım, herhangi bir iş in ya da iş lemin yapılması için bilgisayar komutları ile yazılan ve bilgisayarda çalıştırılması sonucu oluşan yapıya verilen isimdir. Bilgisayarda bir program oluşturmak için programlama dilleri denilen kodlama sistemine ihtiyaç duyulur. Programlama dillerinin kendilerine has yazım kuralları ve mantıkları vardır. Bu sayede bilgisayar komutlarını belirli bir sistematik içerisinde yazmayı ve çalıştırmayı sağlarlar (Coş ar, 2013: 24). Gerçek hayattaki bir durumu bilgisayarın anlayacağı şekilde verilere dönüştürmek için kod sistemi kullanılır. Kullanılan bu dil de bilgisayar programlama öğrenimindeki en temel zorluktur (Erdoğan, 2005: 10).

İlk bilgisayar programcısı olarak anılan Ada Lovelace, yaşadığı 19. yüzyılda kadınlara karşı var olan dışlama, ön yargı ve kalıplaşmışlığa rağmen bilgisayar bilimcisi olarak başarılı olmuştur (Coe ve Ferworn, 2016: 46-47). Bilgisayar bilimlerine yaptığı katkılardan ötürü ABD Savunma Bakanlığı 1980'de, bilgisayar programlama dillerinden birine Ada'nın ismini vermiştir (Meriwether, 2013). Araştırmacılar, 1960'lardan beri hem eğitim için bilgisayar programlama konusunda hem de daha fazla insana bilgisayar programlamayı öğretmek amacıyla çeşitli programlama dilleri geliştirme konusunda araştırmalar yapmaktadır (Kelleher ve Pausch, 2005: 83; Morgado, Cruz ve Kahn, 2010: 310).

Jean Piaget ile Cenevre'de çalışan matematikçi Seymour Papert 1960'ların ortalarında, Amerika Birleşik Devletleri'ne gelerek Marvin Minsky ile birlikte MIT Yapay Zeka Laboratuvarı'nı kurmuştur. 1970'lerin başında, Papert ve MIT Yapay Zeka Laboratuvarı'ndaki öğrencileri, çocuklara bilgisayar bilimi dünyasını tanıtmaya ve programlamayı öğretme yöntemlerini araştırmaya başlamıştır. Bu araştırmalar LOGO programlama dilinin ve *Logo turtle* isimli, basketbol topu büyüklüğündeki bir robotun oluşturulmasıyla sonuçlanmıştır. Bu robot ileri, geri, sağ ve sol komutlarına göre hareket edebiliyordu (McNerney, 2004: 326-327). *Logo turtle*, matematiksel ve mantıksal problem çözme becerilerini değil, bilgisayar programlamayı öğretmek için tasarlanmış olsa da Logo'nun sadeliği, çocuklara bilgi işlem gücüne dair bir bakış açısı kazandırmıştır. Böylece programlamanın, çocukları zorlayıcı problem çözme becerilerine dâhil etmek için uygun bir araç olabileceği fark edilmiştir (Manches ve Plowman, 2017: 195). Öte yandan *Logo turtle* bir klavye ile Logo kodu yazılarak kullanılabilirdiği için küçük çocuklar

için uygun bir programlama aracı değildi (McNerney, 2004: 327). 1970'li yılların ortalarında MIT Logo Laboratuvarı'nda yüksek lisans öğrencisi olan Radia Perlman, küçük çocukların metin tabanlı bilgisayar programlarına erişimindeki temel engelin sadece bilgisayar dillerinin sözdizimi kurallarının değil, aynı zamanda kullanıcı arabirimi olduğunu düşünüyordu (McNerney, 2004: 328). 1974-1976 yılları arasında Radia Perlman *Logo turtle*'dan esinlenerek TORTIS isimindeki somut bir programlama sistemini geliştirmiştir (Morgado, Cruz ve Kahn, 2006: 1903). Perlman'ın temel tasarım hedefi, metin yazarak programlamanın oluşturduğu engelin üstesinden gelmek ve üç ila dört yaş arasındaki çocukların bilgisayar programlama öğreniminden faydalanmalarını sağlayacak bir araç geliştirmektir (Fessakis, Gouli ve Mavroudi, 2013: 88). Perlman, küçük çocukların çeşitli Logo komutlarını temsil eden kartları üç farklı renkteki raflara yerleştirmelerine olanak tanıyan "Slot Machines" adlı bir ara yüz geliştirdi ve bu ara yüz de kendinden sonra gelen alt programları temsil etti (Horn, 2009: 26). Perlman'ın kapısını açtığı somut programlama yazılımları, özellikle çocukları hedef alan ve giderek büyüyen bir araştırma alanı haline geldi (Xie, Antle ve Motamedi, 2008: 191).

### **Bilişimsel Düşünme**

194

Bütün öğrenciler sadece programlamanın ne olduğunu değil, her türlü problemi daha etkin bir şekilde çözmek için daha sistemli düşünmenin ne anlama geldiğini bilmelidirler (Kafai ve Burke, 2013: 65). Bu yüzden de öğrencilerin bilişimsel düşünme yeteneklerini geliştirmeleri gerekmektedir. Bundy (2007: 67), 21. yüzyılı anlamak istiyorsanız önce bilişimi anlamalısınız diyerek günümüzde bilişimsel düşünmenin önemini ortaya koymuştur. Wing'e (2006: 33) göre bilişimsel düşünme, bilgisayar bilimlerinin temel kavramlarını kullanarak problemleri çözmeyi, sistemleri tasarlamayı ve insan davranışlarını anlamayı içermektedir.

Bilişimsel düşünme, hem fen hem de sosyal bilimlerdeki neredeyse tüm disiplinleri ilgilendirmektedir (Bundy, 2007: 67). Wing (2006: 33), bilişimsel düşünmenin sadece bilgisayar bilimcileri için değil, herkes için temel bir beceri olduğunu savunmuştur. Okuma, yazma ve aritmetik için, her çocuğun analitik yeteneğine bilişimsel düşünmeyi de eklemesi gerektiğini vurgulamıştır.

Wing (2008: 3717), bilişimsel düşünmeyi analitik düşünmenin bir türü olarak ele almıştır. Matematiksel düşünmeyle bir problemi çözme yoluna gidebileceğimizi, mühendislik düşünmesiyle gerçek dünyanın sınırları içinde faaliyet gösteren büyük ve karmaşık bir sistemi tasarlamaya ve değerlendirmeye yaklaşabileceğimizi, bilişimsel düşünmeyle hesaplanabilirliği, zekayı, zihni ve insan davranışını anlama yaklaşımına girebileceğimizi genel şekilleriyle paylaştığını ifade etmiştir. Bunun yanında Barr, Harrison ve Conery (2011: 22), bilişimsel düşünmenin düşünme becerilerinin eşsiz bir birleşimi olduğu için matematiksel ve eleştirel düşünmeden ayrıldığını belirtmektedirler.

Bu bilgiler ışığında öğretmenler bilişimsel düşünmenin önemini kabul etmekte ve öğrencilerin bu becerileri kazanmalarına yardımcı olmak istemektedir. Ancak çalışmalarına rehberlik edecek açık ve pratik bir tariften yoksunlardır (Barr, Harrison ve Conery, 2011: 23).

### **Eğitimde Bilgisayar Programlama**

Günümüz dünyası teknoloji odaklıdır ve bu dünyada üretebilen bir ülke olmak için bireylerin programlama becerilerine sahip olmaları gerekmektedir. Bu yüzden yeni yetişen nesillere mevcut programları tüketmek yerine yeni programları nasıl üretecekleri konusunda yol göstermek önem kazanmaktadır (Demirer ve Sak, 2016: 523-524). Tüketiciliği önlemek için, çocukların yaratıcılığı teşvik edilmelidir. Bu da çocukların tüketmeyi bırakıp üreten olmaya başlaması gerektiği anlamına gelir. Bunun için de çocukların bilgisayar programlama becerilerini geliştirmesi önemlidir (Anzoátegui, Pereira ve Jarrín, 2017: 1). Dünyada ve ülkemizde programlama eğitimi alanının üzerinde dikkatle durulmasının sebebi, bu alanın bilgisayar teknolojisinin gelişim sürecinde yapıtaş olmasıdır. Programlama eğitimi, yazılım çalışmalarının var olmasını sağlayan önemli bir eğitim alanıdır (Kert ve Uğraş, 2009: 2). Amerikalı ebeveynlerin çoğu bilgisayar programlamanın en az okuma, yazma ve matematik öğrenmek kadar önemli olduğunu düşünmektedirler. Bugünün öğrencileri gelecek yaşantılarında sadece fen, teknoloji, mühendislik ve matematik (STEM) alanlarında değil, aynı zamanda STEM dışı alanlarda da kariyerleri için bilgisayar programlama kullanacaklardır (K-12 Computer Science Framework, 2016: 1).



Bilgisayar programlama araçları gibi dijital teknolojiler, akademik içerikler ile eğlenceli ve anlamlı aktiviteler arasında köprü kurarlar (Flannery, Kazakoff, Bontá, Silverman, Bers ve Resnick, 2013: 1). Böylelikle öğrenme çocuklar için eğlenceli bir sürece döner. Kodlama ya da bilgisayar programlama sürecine bir çeşit problem çözme süreci de denilebilir. Öğrenciler, hayal güçleri ve yaratıcılıkları ile programlamayı kullanarak var olan bir probleme çözüm üretirler (Çankaya, Durak ve Yünkül, 2017: 430). Bilgisayar programlama, sadece matematik ve bilimde değil aynı zamanda dil becerileri, yaratıcılık ve sosyal-duygusal etkileşimde çocukların başarısı üzerinde olumlu ve ölçülebilir bir etkiye sahiptir (Bers, 2008: 122).

### **Okul Öncesinde Bilgisayar Programlama**

Erken çocukluk eğitiminde küçük çocukların öğrenmelerine, aktif sorgulama yoluyla çeşitli deneyimler yaratarak, yardımcı olacak yapılandırmacı yöntemleri kullanmanın yararları uzun zaman önce fark edilmiş ve bu yöntemler yaygın olarak kabul görmüştür (Bers, Ponte, Juelich ve Schenker, 2002: 123). Bu çerçevede okul öncesi dönemde oyun ve öğrenme birbiri ile bütünleştirilmiştir. Çocuklar resim yaparken renklerin nasıl karıştığını, bloklardan kule yaparken de bir yapıyı neyin ayakta tuttuğunu veya neyin devirdiğini deneyimler. Bu gibi oyun aktiviteleri sayesinde çocuklar fikirlerini test eder, hayal güçlerini kullanır ve yeni olasılıkları keşfederler. Bilgisayar biliminin, çocuklara oynayarak öğrenebildikleri ve öğrenirken de oynayabildikleri bir öğrenme ortamı sunduğu için erken çocukluk eğitimi için oldukça uygun olduğu kabul edilmektedir (Resnick, 2003). Temelleri okul öncesi dönemde atılan bilgisayar bilimi kavramları ve uygulamaları, ilerleyen yıllardaki bilgisayar bilimleri uygulamaları için bilgi ve anlayışı geliştirirler (K-12 Computer Science Framework, 2016: 184).

Programlama dillerini öğrenmek aslında yetişkinler için bile zor ve sıkıcı bir süreçtir. Bu yüzden de küçük yaştaki çocukların ilgisini çekmemesi oldukça normaldir (Demirer ve Sak, 2016: 524). Küçük yaştaki çocukların programlama eğitimi sürecinde karşılaştığı en büyük engel programlama dillerinin yazıya dayalı olmasıdır. Ulaşılabilen çoğu programlama ara yüzleri çocukların kavramasını zorlaştıran yazı ve sembollere dayalıdır (Wang, Zhang, Qi ve Sun, 2015: 219). Ancak yapılan araştırmalar, dört yaşındaki çocukların temel bilgisayar programlama kavramlarını anlayabildiğini ve basit

robotik projeleri inşa edip programlayabildiğini göstermiştir (Flannery, Kazakoff, Bontá, Silverman, Bers ve Resnick, 2013: 1).

Başlangıçta programlama araçları sadece yazı ve grafik tabanlı olmasına karşın günümüzde dokunulabilir (tangible) programlama araçları da programlama eğitiminde kullanılmaktadır. Grafik tabanlı programlama araçları, kullanıcıların klavye veya fare aracılığı ile komutları yazarak veya simge halindeki komut bloklarını sürükleyip birbirine ekleyerek programlama yapıları oluşturmaya olanak tanımaktadır (Sapounidis, Demetriadis ve Stamelos, 2015: 225). Öte yandan bazı grafik tabanlı programlama araçlarında sadece simgeler kullanıldığı için çocukların okuma yazma bilmesi gerekmemektedir. Bu nedenle de bunlar okul öncesi dönemi çocukları için uygun yazılımlardır. Bu grafiksel yazılımlar çocukların bilgisayar ekranındaki simgeleri sürükleyip birleştirerek programlama yapmalarına olanak sağlarlar (Sapounidis ve Demetriadis, 2013: 1776). Fessakis, Gouli ve Mavroudi (2013), beş ila altı yaşındaki 10 çocukla Logo tabanlı bir dijital programlama aracı kullanarak çalışma gerçekleştirmiştir. Çalışma sırasında çocuklar ortaya konulan problemi çözmeye coşku ve zevk sergileyerek sürece aktif ve yoğun bir şekilde katılmışlardır. Çalışma sonunda çocuklar temel bilgisayar programlama kavramlarını tanımışlardır. Çocuklar problem çözme stratejilerini geliştirmelerinin yanı sıra işbirliği, yardımlaşma ve fikir alışverişi gibi etkileşimlerle çocuklar iletişim becerilerini de geliştirmişlerdir.

197

Okul öncesine yönelik popüler grafik tabanlı programlama yazılımlarından birisi ScratchJr'dır. ScratchJr, Scratch'ı temel alan ve beş ila yedi yaş aralığındaki çocukların gelişim ve öğrenme ihtiyaçları için yeniden tasarlanan bir grafik programlama dilidir. Küçük çocuklar için tasarlanan mobil uygulamalar, yazılım ve bağımsız teknolojilerin çoğu, içerik oluşturma veya bilişimsel düşünme yerine harf ve sayıların tanınması gibi temel akademik becerilere odaklanma eğilimindedir. ScratchJr projesi, beş ila yedi yaş arasındaki çocukların, etkileşimli animasyonlar ve hikâyeler oluşturarak programlama ve problem çözme kavramlarını gerçekten öğrenip uygulayabilmesi üzerine dayanmaktadır. Çocuklar seçtikleri karakterleri hareket ettirmek, atlatmak, dans ettirmek ve şarkı söyletmek için grafiksel programlama bloklarını bir araya getirirler. Çocuklar ScratchJr ile kodlarken, sadece bilgisayarla etkileşim kurmayı değil, bilgisayarla kendilerini nasıl

ifade edebileceklerini de öğrenirler (Flannery, Kazakoff, Bontá, Silverman, Bers ve Resnick, 2013: 7-10; ScratchJr, 2018).

Okul öncesi çocukların kodlama yapmasına olanak sağlayan bir başka programlama ara yüzü ise Kodable'dır. Kodable, beş ila yedi yaş arasındaki çocuklar için tasarlanmıştır, ancak Kodable'ın yaratıcılarından Huebner, yazılımın üç yaşındaki çocuklar tarafından bile kullanıldığını belirtmektedir (Orsini, 2013). Kodable yaratıcıları “öğretmenler öğrenmenin kapı bekçileridir” düşüncesiyle hareket ederek öğretmeni merkeze almış ve yazılım geliştirilirken öğretmenlerin ihtiyaçları göz önünde tutulmuştur (Kodable, 2018). Kodable'da yazılı talimatlar bulunmamaktadır. Çocuklar tablette sürükleyip bırakma komutunu öğrendiklerinde yani yaklaşık dört veya beş yaşlarında, Kodable ile oynamaya hazırlardır. Kodable çeşitli seviyelerden oluşan ve çocukların kodlama yaparak oyun oynamasına olanak sağlayan bir programdır. Çocuklara çeşitli seviyelerdeki problemleri çözmek için bir programcı gibi düşünmeyi öğretir. Çocuklar farklı seviyelerde oynarken programlamanın temel kavramları olan fonksiyonları, koşullu ifadeleri, eğer/sonra (if/then) ifadelerini ve döngüleri (loops) kullanırlar (Orsini, 2013).

Lightbot, okul öncesi dönem çocuklarının programlama yapmasına olanak sağlayan bir başka ara yüzüdür. 2008 yılında Daniel Yaroslavski tarafından geliştirilmiştir ve kodlama ile ilgili temel kavramları çocuklara oyun yoluyla bulmaca çözer gibi öğretmeyi hedeflemektedir (Saygıner ve Tüzün, 2017: 83). Bu uygulama da tıpkı Kodable gibi çeşitli seviyelerden oluşmaktadır. Her seviyede verilen problemin çocuklar tarafından kodlama ile ilgili temel becerileri kullanarak çözülmesi beklenmektedir. Ara yüz sembollerden oluşmaktadır ve yazılı komutlar bulunmadığı için çocukların kullanımına uygundur (Biggs, 2013).

Dokunulabilir ya da somut programlama araçları grafik tabanlı programlama yazılımlardan farklı olarak, uygulama sırasında çeşitli dijital etkileri tetikleyebilen fiziksel nesnelere olarak düşünülebilir. Geleneksel masaüstü bilgisayar kullanımında, ekran sadece dijital bir dünyaya ulaştığımız bir penceredir, somut programlama araçlarında ise programlama aracıyla birlikte hareket eder ve programlama aracının kendisine dokunuruz (Hornecker ve Buur, 2006: 437). Somut programlama araçları çocuklar veya programlamaya yeni başlayanlar için yaratıcı oyunlar ve yeni öğrenme

yolları sunmaktadır (Sapounidis, Demetriadis ve Stamelos, 2015: 226). Somut programlama, insan etkileşimi ile fiziksel dünyayı bir araya getiren bir etkileşim yöntemi olarak, öğrenmeyi geliştirmekte ve çocukları programlamaya kolayca dâhil etmektedir (Wang, Zhang, Qi ve Sun, 2015: 219). Programlama sürecinde yaşanan öğrenme zorluklarını azaltması somut programlama araçlarının avantajlarından biri olarak kabul edilmektedir (Sapounidis, Demetriadis ve Stamelos, 2015: 226). Wang, Zhang, Qi ve Sun (2015: 224), TanPro-Kit 2.0 isimli somut programlama aracını kullanarak, yaşları altı ila sekiz arasında olan 15 çocukla laboratuvar temelli bir çalışma gerçekleştirmişlerdir. Süreç boyunca çocukların davranışlarını ve konuşmalarını video ile kaydetmişlerdir. Araştırma sonunda çocukların TanPro-Kit 2.0'ın tüm programlama özelliklerini kısa sürede keşfettiği, IF (eğer) komutu gibi yeni kavramların ilgisini çektiği görülmüştür. Bunun yanında bazı çocuklar, programlamanın ilginç olduğunu düşündüklerini ve okuldan sonra bazı programlama dersleri almak istediklerini ifade etmişlerdir. Sapounidis ve Demetriadis (2013: 1784), çocukların bilgisayar programlamada kullanabilecekleri kullanıcı arayüzleri ile ilgili görüş ve tercihlerini araştırmıştır. Çalışmada elde edilen sonuçlar, somut programlama araçlarının özellikle kız çocukları için daha ilgi çekici, daha eğlenceli olduğu ve küçük çocuklar tarafından daha kolay kullanıldığını göstermiştir. Somut programlama aracı kullanmanın, çocuklara grup içindeki işbirliğinde eşit katılım fırsatını daha fazla sunduğu kaydedilmiştir. Grafik tabanlı programlama aracı kullanılırken bir çocuk programın kontrolünü sağlarken diğerlerinin ilgisinin azaldığı görülmüştür.

Somut programlama ara yüzlerinin temeli olarak kabul edilen TORTIS Slot Machine kendinden sonra gelen somut programlama araçlarına öncülük etmiştir. Ancak somut programlama terimi aslında AlgoBlock sisteminin geliştiricileri olan Suzuki ve Kato tarafından yaygınlaşmıştır. Logo gibi somut bir programlama dili olan AlgoBlock, bir program oluşturmak için birbirine manuel olarak bağlanabilen bir dizi somut bloktan oluşmaktadır. Bu bloklar daha sonra programı yürütmek için bir bilgisayara bağlanmaktadır. Her blok bir programlama dilinin komutuna karşılık gelmektedir. Programlamanın çalışma yolu somut olsa da, programın çalışmasının etkisi sanaldır. Kullanıcılar bloklar aracılığıyla komutlar vererek bilgisayar ekranındaki bir denizaltını yönlendirmektedirler. AlgoBlock'un amacı, ilkokul ve ortaokuldaki öğrencilerin grup etkinlikleri yoluyla planlama ve inşa etme becerilerini geliştirmelerine olanak tanımadır

(Suzuki ve Kato, 1993: 298; McNerney, 2004: 331-332). AlgoBlock'un tersi bir uygulama 2013 yılında Wonder Workshop tarafından sunulan bir robot ile hayata geçmiştir. Dash and Dot isimli bu robotu çocuklar tablet ya da telefonlarına ücretsiz olarak indirdikleri uygulama aracılığı ile programlayabilme olanağına sahiptirler (Elkin, Sullivan ve Bers, 2016: 170). Bu robotun ana hedefi, etkileşimli oyunlar yoluyla beş yaşından büyük çocuklara temel programlama kavramlarını öğretmektir (Ramírez-Benavides, López ve Guerrero, 2016: 2).

Son yıllarda, küçük çocuklar için piyasaya birçok yeni robot kitleri sürülmeye başlanmıştır (Elkin, Sullivan ve Bers, 2016: 170). Çocukları ekrandan uzak tutarak onlara programlamayı öğretmeyi amaçlayan bu eğitsel robotik kitler çocukların bloklar, boncuklar ve toplar gibi geleneksel materyallerle öğrendikleri sayı, boyut ve şekil gibi matematiksel kavramları daha güçlü bir şekilde anlamalarına yardımcı olan yeni nesil öğrenme araçları haline gelmiştir (Sullivan, Elkin ve Bers, 2015: 418). Örneğin, programlanabilen oyuncak olarak piyasada bulunan Bee-Bot, okul öncesi dönem çocuklarının sıralama, tahmin etme ve problem çözme becerilerini desteklemek için kullanılmaktadır (Janka, 2008: 114-116). Bee-Bot, arı şeklindeki bir zemin robotudur. Robotun üzerinde çocukların programlama yapmasını sağlayan yön tuşları bulunmaktadır (Elkin, Sullivan ve Bers, 2016: 170). Cubetto isimli bir başka robot ise Bee-Bot'a benzer özelliktedir, ancak Cubetto'da programlama işlemi kontrol tahtası aracılığı ile gerçekleşmektedir. Kitin içinde bulunan yönlendirme blokları kontrol tahtasına yerleştirilerek robot programlanmakta ve çocuklara ekran kullanmadan programlama deneyimi sunmaktadır. Cubetto çocuklara sistemik ve yaratıcı düşünme, işbirlikçi çalışma ve sosyal iletişim gibi gelişmiş bilişsel süreçler geliştirmeleri için fırsatlar sağlamaktadır (Anzoátegui, Pereira ve Jarrín, 2017: 2). Kazakoff, Sullivan ve Bers (2013: 251), robotik atölyesindeki bir haftalık yoğun bir süreç sonunda programlama robotlarının okul öncesi dönem çocuklarının sıralama yeteneği üzerindeki etkisini incelemiştir. Araştırma sonunda okul öncesi dönem çocukları ile sadece bir hafta robotik ve programlama çalışmanın sıralama puanları üzerinde anlamlı ve pozitif bir etki gösterdiği görülmüştür.

2011 yılında Tuft Üniversitesinde, Marina Umaschi Bers'in liderliğindeki DevTech araştırma grubu çocuklara kodlama öğretmek amacıyla çalışmalara başlamıştır. Bu araştırma projesi STEM'in bileşenlerinden, teknolojinin "T"si ve erken çocukluk

eğitiminde en fazla ihmal edilen mühendisliğin (engineering) “E”sine odaklanmaktaydı. Proje önce KIWI isimli robotu tasarlamış, daha sonra bu ilk örnek geliştirilerek şu an piyasada KIBO ismiyle anılan robot kitini geliştirmiştir (DevTech Research Group, 2018). KIBO, robotu hareket ettirmek için kullanılan donanımı (yani robotun kendisini) ve yazılımı (yani somut programlama bloklarını) içeren bir robotik yapı takımıdır. Bu sette, inşa etmesi oldukça kolay olan tekerlekler, motorlar, ışık ve sensörler bulunmaktadır (Sullivan, Bers ve Mihm, 2017: 111). Piyasada var olan diğer birçok eğitsel robot kitleri çocukları bir robot inşa etmenin tasarım yönüne dâhil etmezken KIBO çocuklara robotu kendilerinin inşa etmesine olanak tanımaktadır (Elkin, Sullivan ve Bers, 2016: 171-173). Kit içerisinde bulunan sanat platformu yardımıyla çocuklar robotlarını projelerine uygun olarak kişiselleştirebilmektedirler. KIBO, birbirine bağlanabilen ahşap programlama blokları kullanılarak hareket etmeye programlanmaktadır. Bu ahşap bloklar elektronik veya dijital bileşenler içermemektedir. Bunun yerine KIBO, üzerine yerleştirilen gömülü bir tarayıcıya sahiptir. Bu tarayıcı ile KIBO ahşap blokların üzerinde bulunan barkotları okumakta ve oluşturulan kod dizinine göre hareket etmektedir. KIBO ile programlamayı öğrenmek için bilgisayar, tablet ya da başka bir ekran zamanı aracı gerekmemektedir (Sullivan, Bers ve Mihm, 2017: 110-111; Sullivan, Elkin ve Bers, 2015: 419). Elkin, Sullivan ve Bers (2016: 180), yaşları üç ile beş arasında değişen 64 okul öncesi dönemi çocuğu ile çalışma yapmışlardır. Bu çalışmada KIBO’yu kullanarak Solve-It isminde bir program uygulamışlardır. Çalışma sonunda çalışmaya katılan çocuklar süreci tamamladıktan sonra temel programlama kavramlarında çok başarılı olmuştur. Bu çalışmadan elde edilen sonuçlar doğrultusunda, KIBO robotunun ve KIBO programlama dilinin bazı yönlerinin dört yaş ve üstü çocuklar için tasarlanmış olmasına rağmen, üç yaşındaki çocuklar için de uygun olduğu görülmüştür.

Sonuç olarak, bilgisayar bilimleri ile ilgili uygulamaları okul öncesi dönemi uygulamalarıyla birleştirmek, geleneksel uygulamalardan ayrılmış değildir. Hatta erken bilgisayar bilimi uygulamaları oyun temelli pedagojiyi desteklemektedir. Erken yıllarda başlayan bilgisayar bilimi uygulamaları çocuklara bilgisayar dünyasını nasıl şekillendirdiğini fark etmelerine, bilgisayar ve programlama ile ilgili temel bazı kavramları adlandırmalarına ve bunları tanımalarına yardımcı olmaktadır (K-12 Computer Science Framework, 2016: 196). Çocuklar kodlama sonunda ortaya çıkardıkları ürünlerle, matematik ve fen bilimleri ile bağlantılı bazı basit soyut kurallar

ve kavramlardan nasıl etkilendiklerine dair kavrayışlar edinmektedirler (Morgado, Cruz ve Kahn, 2010: 311).

## **SONUÇ VE ÖNERİLER**

21. yüzyıl çocukları dijital teknolojinin içinde büyümektedirler. Hemen hemen her çocuk çevrelerinde bulunan dijital teknoloji ürününün ne olduğunu ve nasıl çalıştığını hakkında fikir sahibidir. Dijital nesil ya da dijital yerliler olarak adlandırılan yeni nesil, vaktinin çoğunu teknoloji ile geçirmektedir. Bir başka açıdan baktığımızda da dijital nesil aslında vaktinin çoğunu dijital teknolojiyi tüketerek geçirmektedir. Yeni neslin, yeni teknoloji ürünleri ile birçok deneyimi ve etkileşimi bulunmakta, ancak buna karşılık yeni teknolojiyi üretmede ve teknoloji ile kendilerini ifade edebilme konusunda çok az deneyimleri bulunmaktadır. Kısacası yeni nesil, teknolojiyi okuyabilmekte ancak yazamamaktadır.

Günümüz dünyasında bilgisayar programlama yeni okur yazarlık olarak kabul edilmektedir. Çocuklar okuma yazma öğrendiklerinde kendilerini ifade etmede ve dünyayı anlamada yeni bir bakış açısı geliştirmektedirler. Bilgisayar programlama da tıpkı okuma yazma öğrenmek gibi çocukların düşünme becerilerini geliştirmede oldukça önemlidir. Çocuklar kodlamayı öğrendiklerinde bilişimsel düşünebilmeyi de öğrenmektedirler. Çocuklar sadece problem çözme becerilerini geliştirmekle kalmamakta aynı zamanda mantıksal düşünme becerilerini de geliştirmektedirler. En önemlisi çocuklar kodlama yaparken kendilerini yaratıcı yollarla ifade edebilme şansına da sahip olmaktadır. Bilgisayar programlama çocukları ileriki yaşamlarında birer mühendis ya da bilgisayar bilimcisi yapmak için değil meslekleri ne olursa olsun dijital teknoloji ile kendilerini ifade etmeleri için önemlidir.

Bilgisayar programlama eğitimi, çocukların gelişim alanları ve becerileri üstünde olumlu etkilere sahiptir. Bunları desteklemede ebeveynlere de rol düşmektedir. Çocuklar vakitlerinin çoğunu evde geçirmektedir. Ebeveynler, çocukların ekran zamanlarında teknolojiyi tüketen olarak değil üreten olarak rol almaları için ScratchJr, Kodable, Lightbot gibi programlama yapmaya olanak sağlayan uygulamalara yönlendirebilirler.

Dijital nesil olarak da adlandırılan yeni nesil çocuklarının teknoloji konusundaki yetkinlikleri öğretmenler tarafından unutulmamalıdır. Öğretmenler de kendilerini dijital teknoloji konusunda geliştirmeli, yenilikleri takip etmeli ve öğrenme ortamlarına bu teknolojiyi öğrencilerin en üst düzeyde fayda sağlayacağı şekilde dahil etmelidirler.



Çalışma bulguları göz önüne alındığında öğretmenlere, öğrencilere bilgisayar biliminin sadece okuma yönünü değil yazma yönünü de ele almaları önerilmektedir. Bunun için öğrencilerin yaş, gelişim düzeyi ve ilgileri doğrultusunda uygun bilgisayar programlama aracı kullanarak, öğrencilerin teknolojiyi üretmelerine olanak sağlanabilir.

Üretkenlik ve yaratıcılık bilgi çağının gerekliliği haline gelmiştir ve yeni yetişen nesle kazandırılmalıdır. Bu nedenle erken yaşta verilecek olan programlama eğitimi önem kazanmaktadır. Bilgisayar programlama eğitimi genellikle ortaöğretim ya da lise kademesinde öğrencilere öğretilmektedir. Bu bağlamda MEB'in yapacağı değişikliklerle programlama eğitimi daha erken yaşlarda çocuklara kazandırılabilir. Okul öncesi dönemde çocuklar çevrelerine karşı daha meraklı ve yeni şeyler öğrenmeye daha açıkken onlara bilgisayar programlamayı öğretmek, ileride çocukların dijital teknolojiyi daha akıcı kullanmalarını sağlayacaktır. Bu yüzden, programlama eğitimine daha erken yaşlarda başlanması konusunda adımlar atılmalıdır. Bunun yanında MEB öğretmenlere hizmet içi eğitimler düzenleyebilir ve öğretmenlerin teknoloji ve programlama ile ilgili bilgilerini güncel tutabilir.

Üniversitelerde öğretmen adaylarına teknoloji ve bilgisayar bilimi öğretimi ile ilgili dersler arttırılmalıdır. Öğretmen adaylarının, dünyadaki programlama eğitimi süreçleri ve bilgisayar programlamadaki farklı uygulamalar hakkında bilgi sahibi olmaları için girişimlerde bulunulmalıdır. Yerli uygulamalar yapılması için öğretmen adayları teşvik edilmelidir.

Literatür bulguları göz önüne alındığında okul öncesi dönemi çocukları için bilgisayar programlama öğretilirken yazılımı ve donanımı somut olan programlama araçlarının kullanılması gerekmektedir. Somut programlama araçları çocukların ekrana ihtiyaç duymadan programlama yapabilmesine olanak sağlayacaktır. Çocuklara programa etkinlikleri verilirken ortaya bir ürün çıkarabileceği projeler tercih edilmelidir.

## KAYNAKÇA

Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, (13)1: 1-4.

Anzoátegui, L. G., Pereira, M. I. ve Jarrín, M. d. (2017). Cubetto for preschoolers: Computer programming code to code. *Computers in Education (SIIE), 2017 International Symposium* (s. 1-5). Lizbon, Portekiz: IEEE. doi: 10.1109/SIIE.2017.8259649.

Barr, D., Harrison, J. ve Conery, L. (2011). Computational thinking: a digital age skill for everyone. *Learning & Leading with Technology*, (38)6: 20-23.

Bers, M. U. (2008). Engineers and Storytellers Using Robotic Manipulatives to Develop Technological Fluency in Early Childhood, O. N. Saracho ve B. Spodek içinde, *Contemporary Perspectives on Science and Technology in Early Childhood Education* (s. 105-125). Charlotte, NC: Information Age Publishing.

Bers, M. U. (2010). The tangibleK robotics program: applied computational thinking for young children. *Early Childhood Research ve Practice*, (12)2: 1-20

Bers, M. U., Ponte, I., Catherine Juelich, A. V. ve Schenker, J. (2002). Teachers as designers: integrating robotics in early childhood education. *Information Technology in Childhood Education Annual*, (2002)1: 123-145.

Bers, M. U., Seddighin, S. ve Sullivan, A. (2013). Ready for robotics: bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education*, (21)3: 355-377.

Biggs, J. (2013, Haziran 26). *LightBot teaches computer science with a cute little robot and some symbol-based programming*. 2 Mayıs 2018 tarihinde Tech Crunch: <https://techcrunch.com/2013/06/26/light-bot-teaches-computer-science-with-a-cute-little-robot-and-some-symbol-based-programming/> adresinden alındı.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, (1)2: 67-69.

Coe, I. R. ve Ferworn, A. (2016). The life and contributions of countess Ada Lovelace: unintended consequences of exclusion, prejudice, and stereotyping. *IEEE Technology and Society Magazine*, (35)4: 46-49. doi: 10.1109/MTS.2016.2618679.

Coşar, M. (2013). *Problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkileri*. Yayınlanmamış doktora tezi, Gazi Üniversitesi, Ankara.

Çankaya, S., Durak, G. ve Yüncül, E. (2017). Robotlarla programlama eğitimi: öğrencilerin deneyimlerinin ve görüşlerinin incelenmesi. *Turkish Online Journal of Qualitative Inquiry*, (8)4: 428-445. doi: 10.17569/tojqi.343218.

Demirer, V. ve Sak, N. (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, (12)3: 521-546.

DevTech Research Group. (2018). *Research: Kibo Robotics*. 8 Mayıs 2018 tarihinde Tufts University: <http://ase.tufts.edu/DevTech/ReadyForRobotics/research.asp> adresinden alındı.

Elkin, M., Sullivan, A. ve Bers, M. U. (2016). Programming with the KIBO robotics kit in preschool classrooms. *Computers in the Schools*, (33)3: 169-186. doi: 10.1080/07380569.2016.1216251

Erdoğan, B., (2005), *Programlama Başarısı ile Akademik Başarı, Genel Yetenek, Bilgisayara Karşı Tutum, Cinsiyet ve Lise Türü Arasındaki İlişkilerinin İncelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.

European Schoolnet. (2015). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. <http://www.eun.org/documents/411753/817341/Computing+our+future+final+2015.pdf/d3780a64-1081-4488-8549-6033200e3c03> Brüksel, Belçika: Anja Balanskat, Katja Engelhardt.

Fessakis, G., Gouli, E. ve Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, (63): 87-97.

Flannery, L. P., Kazakoff, E. R., Bontá, P., Silverman, B., Bers, M. U. ve Resnick, M. (2013). Designing scratchJr: support for early childhood learning through computer

programming. *12th International Conference on Interaction Design and Children* (s. 1-10). New York, USA: ACM. doi: 10.1145/2485760.2485785.

Hicks, S. D. (2011). Technology in today's classroom: are you a tech-savvy teacher? *The Clearing House: A Journal of Educational Strategies*, (84)5: 188-191. doi: 10.1080/00098655.2011.557406.

Horn, M. S. (2009). *Tangible computer programming: exploring the use of emerging technology in classrooms and science museums*. Doctoral dissertation, Tufts University.

Hornecker, E. ve Buur, J. (2006). Getting a grip on tangible interaction: a framework on physical space and social interaction. *Proceedings of the SIGCHI conference on Human Factors in computing systems* (s. 437-446). Montréal, Québec, Canada: ACM. Doi:10.1145/1124772.1124838.

International Technology Education Association. (2007). *Standards for technological literacy: Content for the study of technology*. Reston, Virginia, ABD: International Technology Education Association.

Janka, P. (2008). Using a programmable toy at preschool age: why and how. *International Conference on Simulation, Modeling and Programming for Autonomous Robots* (s. 112-121). Venedik, İtalya: SIMPAR.

K-12 Computer Science Framework. (2016). *K-12 Computer Science Framework*. 16 Nisan 2018 tarihinde K-12 Computer Science Framework: <http://www.k12cs.org>. adresinden alındı.

Kafai, Y. B. ve Burke, Q. (2013). Computer programming goes backt to school. *Phi Delta Kappan*, (95)1: 61-65.

Kazakoff, E. R., Sullivan, A. ve Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, (41)4: 245-255. doi: 10.1007/s10643-012-0554-5.

Kelleher, C. ve Pausch, R. (2005). Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, (37)2: 83-137.

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

Kodable. (2018). *About Kodable*. 2 Mayıs 2018 tarihinde Kodable: <https://www.kodable.com/about> adresinden alındı.

Manches, A. ve Plowman, L. (2017). Computing education in children's early years: a call for debate. *British Journal of Educational Technology*, (48)1: 191-201. doi:10.1111/bjet.12355.

McNerney, T. S. (2004). From turtles to tangible programming bricks: explorations in physical language design. *Personal and Ubiquitous Computing*, (8)5: 326-337. doi: 10.1007/s00779-004-0295-6.

Meriwether, D. H. (2013). *Ada Lovelace*. Salem Press Biographical Encyclopedia.

Morgado, L., Cruz, M. ve Kahn, K. (2010). Preschool cookbook of computer programming topics. *Australasian Journal of Educational Technology*, (26)3: 309-326.

Morgado, L., Cruz, M. ve Kahn, K. (2006). Radia Perlman – a pioneer of young children computer programming. *Current Developments in Technology-Assisted Education*, 1903-1908.

Orsini, L. (2013, Nisan 23). *Kodable Teaches Kids To Code Before They Learn To Read*. 2 Mayıs 2018 tarihinde Readwrite: <https://readwrite.com/2013/04/23/kodable-teaches-kids-to-code-before-they-learn-to-read/> adresinden alındı.

Parker, S. K. (2014). *Hawaii public school principals' level of technology use and the meaningful integration of technology in their school*. (Doktora tezi). ProQuest Dissertations and Theses veri tabanından erişildi. (UMI No: 3628272).

Ramírez-Benavides, K., López, G. ve Guerrero, L. A. (2016). A mobile application that allows children in the early childhood to program robots. *Mobile Information Systems*. doi: 10.1155/2016/1714350.

Resnick, M. (2003). Playful learning and creative societies. *Education Update*, (8)6: 1-2.

Sapounidis, T. ve Demetriadis, S. (2013). Tangible versus graphical user interfaces for robot programming: exploring cross-age children's preferences. *Personal and ubiquitous computing*, (17)8: 1775-1786. doi: 10.1007/s00779-013-0641-7.

Sapounidis, T., Demetriadis, S., ve Stamelos, I. (2015). Evaluating children performance with graphical and tangible robot programming tools. *Personal and Ubiquitous Computing*, (19)1: 225-237. doi: 10.1007/s00779-014-0774-3.

Saygıner, Ş. ve Tüzün, H. (2017). Programalama eğitiminde yaşanan zorluklar ve çözüm önerileri. *11. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu* (s. 78-90). Malatya: İnönü Üniversitesi.

ScratchJr. (2017). *About ScratchJr*. 27 Nisan 2018 tarihinde ScratchJr: <https://www.scratchjr.org/about/info> adresinden alındı.

Sullivan, A. A., Bers, M. U. ve Mihm, C. (2017). Imagining, playing, and coding with KIBO: using robotics to foster computational thinking in young children. *Conference Proceedings of International Conference on Computational Thinking Education* (s. 110-115). Hong Kong: The Education University of Hong Kong.

Sullivan, A., Elkin, M. ve Bers, M. U. (2015). KIBO robot demo: engaging young children in programming and engineering. *14th international conference on interaction design and children* (s. 418-421). Medford, USA: ACM.

Suzuki, H. ve Kato, H. (1993). AlgoBlock: a tangible programming language, a tool for collaborative learning. *4th European Logo Conference*, (s. 297-303).

Office of Educational Technology. (2016). *Future ready learning: reimagining the role of technology in education. 2016 national education technology plan*. Washington, ABD: Susan Thomas.

Wang, D., Zhang, L., Qi, Y. ve Sun, F. (2015). A TUI-based Programming Tool for Children. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education* (s. 219-224). Vilnius, Lithuania: ACM.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, (49)3: 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, (366): 3717-3725. doi: 10.1098/rsta.2008.0118.

Xie, L., Antle, A. N. ve Motamedi, N. (2008). Are tangibles more fun? comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. *Second International Conference on Tangible and Embedded Interaction* (s. 191-198). Bonn, Germany: ACM.